УТВЕРЖДЕН

RU.НКБГ.70007-08 90-ЛУ

Программно-аппаратный комплекс

Dionis-NX

Программное обеспечение

Руководство по настройке ПО

RU.НКБГ.70007-08 90

Листов 690

Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
Ф0585				

2024

ОГЛАВЛЕНИЕ

Стр.

1 Общие сведения **9**
1.1 Функциональные возможности ... 9
1.2 Дополнительные технические данные ... 15

2 Установка с флеш-диска **17**

3 Основы работы с интерфейсом командной строки **19**
3.1 Режимы работы с системой ... 20
3.2 Виртуальные консоли ... 21
3.3 Пространства имен ... 21
3.4 Работа с конфигурациями ... 23
3.5 Просмотр и копирование конфигураций ... 23
3.6 Команды режима configure и их влияние на действующую конфигурацию ... 25
3.7 Команды работы с файлами ... 29

4 Выключение и перезагрузка **32**

5 Предварительная настройка **33**
5.1 Начало работы (Установка и смена паролей) ... 33
5.2 Серийный номер изделия ... 33
5.3 Общие настройки ... 33
5.4 Включение IPv6 ... 35
5.5 Интерфейсы ... 35
5.6 Статическая маршрутизация ... 37
5.7 Клиент DNS ... 39
5.8 Просмотр и настройка ARP-таблицы ... 41
5.9 Просмотр и настройка таблицы соседей IPv6 (neighbours) ... 42
5.10 Настройки стека TCP/IP ... 43
5.11 Основы работы со службами (сервисами) ... 46
5.12 Диагностика ... 47

6 Учетные записи **50**
6.1 Учетная запись консольного доступа (учетная запись оператора) ... 50
6.2 Учетные записи администратора ... 50
6.3 Управление учетными записями ... 51

7 Ролевая модель **58**
7.1 Права доступа учетной записи администратора ... 58
7.2 Полномочия системы ... 58
7.3 Команды управления полномочиями и ролями системы для учетных записей ... 66
7.4 Управление ролями ... 67
7.5 Отображение полномочий и зависимости полномочий ... 68

8 Фильтрация и модификация **70**
8.1 Дефрагментация ... 70
8.2 Создание ip access-list ... 71
8.3 Привязка ip access-list ... 72
8.4 Правила отбора ... 73
8.5 Другие правила списков контроля доступа ... 84
8.6 Модификация ... 85
8.7 IPv6 ... 85
8.8 Группирование IP адресов ... 86
8.9 Группирование портов ... 86
8.10 Группирование протоколов и портов ... 87
8.11 Группирование IP адресов, протоколов и портов ... 87

9 Многоадресная передача 88
9.1 Общие сведения о настройке многоадресной маршрутизации ... 90
9.2 Настройка протокола DVMRP ... 90
9.3 Настройка протокола PIM ... 92
9.4 Настройка протокола IGMP ... 100
9.5 Настройка статической многоадресной маршрутизации ... 102
9.6 Мониторинг работы многоадресной маршрутизации ... 102
9.7 Работа со службой ... 103
9.8 Пример настройки видео-трансляции при помощи VideoLAN Vlc ... 103
9.9 Сокращения и термины ... 104

10 NAT 105
10.1 Создание ip nat-list ... 106
10.2 Другие типы NAT ... 107
10.3 Привязка ip nat-list ... 107
10.4 Просмотр и удаление активных соединений ... 108

11 Helper 109
11.1 Создание helper-list ... 109
11.2 Применение helper-list ... 109

12 Журналирование и отладка 110
12.1 tcpdump ... 110
12.2 Трассировка ... 111
12.3 Протоколирование правил фильтрации ... 114
12.4 Системные журналы ... 114
12.5 Сигнал тревоги ... 115

13 AAA - аутентификация и авторизация с помощью протоколов семейства AAA. 119
13.1 Введение ... 119
13.2 Настройка для подключения к RADIUS серверу ... 119
13.3 Настройка для подключения к TACACS+ серверу ... 120

14 FTP-сервер 122
14.1 Введение ... 122
14.2 Настройка анонимного FTP-сервера ... 122
14.3 Настройка приватного FTP-сервера ... 122
14.4 Дополнительные команды конфигурации FTP-сервера ... 123

15 VLAN 125

16 IEEE 802.1ad (QinQ) 126

17 VXLAN 127

18 Bonding-интерфейсы 128
18.1 Режимы агрегации ... 128
18.2 Режимы мониторинга ... 134

19 Сетевые мосты 135

20 Интерфейсы E1 137
20.1 Установка и подключение платы контроллера E1 ... 137
20.2 Настройка контроллера ... 138
20.3 Настройка интерфейса ... 140

21 Фильтрация и NAT в интерфейсах bridge 143
21.1 Создание bridge access-list ... 143
21.2 Привязка bridge access-list ... 144
21.3 Создание bridge nat-list ... 144

21.4 Привязка bridge nat-list 146
21.5 Правила отбора 146

22 GRE-туннели **148**

23 GRETAP-туннели **149**

24 IP6GRE-туннели **150**

25 IP6GRETAP-туннели **151**

26 VPN-туннели **152**
26.1 Введение 152
26.2 VPN-интерфейс 152
26.3 SVPN-интерфейс 156

27 Экспорт статистики по Netflow **162**

28 Служба NTP **165**

29 Служба DNS **168**
29.1 Контроль доступа 169
29.2 Виды 170
29.3 Зоны 172
29.4 Другие настройки 180
29.5 DNSSEC 182
29.6 Динамическое обновление зон 190
29.7 Ограничения ресурсов службы 195
29.8 Журналы 196
29.9 Диагностика 197
29.10 Работа со службой 199

30 Служба DHCP **201**
30.1 Общие настройки службы 201
30.2 Настройка статического назначения 203
30.3 Настройка динамического назначения 204
30.4 Сетевые DHCP-опции 204
30.5 Пользовательские DHCP-опции 205
30.6 Связь с DNS (динамическое обновление) 206
30.7 Работа со службой 206
30.8 Примеры 207

31 Служба DHCP6 **208**
31.1 Общие настройки службы 208
31.2 Настройка статического назначения 208
31.3 Настройка динамического назначения 209
31.4 Делегирование префиксов 210
31.5 Сетевые DHCP-опции 210
31.6 Пользовательские DHCP-опции 211
31.7 Связь с DNS (динамическое обновление) 211
31.8 Работа со службой 211
31.9 Примеры 211

32 Служба DCHPRELAY **213**
32.1 Основные настройки 213
32.2 Дополнительные настройки 213
32.3 Пример 214

33 Служба DCHPRELAY6 **215**
33.1 Основные настройки 215

33.2 Дополнительные настройки 215
33.3 Пример 216

34 Служба PROXY **217**
34.1 Общие понятия 218
34.2 Общая настройка службы 221
34.3 Настройка работы с HTTPS-трафиком 223
34.4 Настройка параметров кэширования. 228
34.5 Настройка доступа к службе 229
34.6 Настройка фильтрации HTTP-заголовков 230
34.7 Настройка выборочного проксирования 230
34.8 Настройка выборочного кэширования 231
34.9 Настройка аутентификации и авторизации 231
34.10 Настройка контроля пропускной способности сети 245
34.11 Правила проверки объектов в кэше на свежесть 247
34.12 Настройка адаптации содержимого 249
34.13 Прочие настройки службы 251
34.14 Настройка журналов службы 251
34.15 Работа со службой 254
34.16 Мониторинг службы 256

35 Служба WCF **269**
35.1 Быстрая настройка. 269
35.2 Аутентификация клиентов службы 269
35.3 Группы пользователей. 271
35.4 Списки фильтрации 271
35.5 Списки фраз 275
35.6 Файл истории 276
35.7 Настройка работы по ICAP 277
35.8 Настройка работы с HTTPS в режиме MITM 278
35.9 Настройка работы с HTTP/HTTPS в прозрачном режиме 278
35.10 Связь со службой proxy 279
35.11 Информация о службе 280

36 Служба SNMP **281**
36.1 Общая настройка службы SNMP 281
36.2 Настройка базовой SNMP информации 281
36.3 Настройка SNMPv3 281
36.4 Настройка правил доступа 283
36.5 Настройка правил нотификаций 285
36.6 Работа со службой 287

37 SSH **288**
37.1 Сервер SSH 288
37.2 Клиент SSH 290
37.3 Соединение с использованием ключей 291
37.4 Передача файлов 292

38 Telnet **293**
38.1 Настройка 293

39 Служба DIWEB **294**

40 Служба TOX **296**

41 Сервис XMPP **298**

42 Служба netperf **301**
42.1 Настройка службы netperf режима configure 301
42.2 Команда netperf режима enable 301

43 Служба IPERF **303**
43.1 Настройки службы iperf режима configure ... 303
43.2 Команда iperf режима enable ... 303

44 Служба SLAGENT **305**
44.1 Настройка службы slagent из режима configure ... 305
44.2 Герерация UDP-запросов ... 305

45 Служба LLDP **307**
45.1 Базовые настройки службы и настройки обязательных TLV ... 307
45.2 Настройка опциональных TLV (DOT1) ... 309
45.3 Настройка опциональных TLV (DOT3) ... 309
45.4 Настройка расширения LLDP-MED ... 311
45.5 Работа со службой ... 313

46 Служба предотвращения вторжений IDSM. **315**
46.1 Режимы ... 315
46.2 Путь пакета через службу idsm ... 315
46.3 Настройка подсистема выборки пакетов ... 316
46.4 Настройки подсистемы декодирования ... 317
46.5 Настройки подсистемы препроцессинга ... 317
46.6 Настройка подсистемы обнаружения ... 320
46.7 Настройка подсистемы событий ... 344
46.8 Настройка подсистемы журналирования ... 346
46.9 Настройка подсистемы счетчиков ... 348
46.10 Начальная настройка ... 349
46.11 Форматы журналов ... 350
46.12 Просмотр информации службы ... 354
46.13 Обновление правил ... 355
46.14 Примеры конфигураций ... 359
46.15 Установка базы данных MySql ... 362
46.16 Установка базы данных Postgres ... 363
46.17 Установка клиента просмотра базы данных BASE ... 365
46.18 Условные обозначения ... 366

47 MAILER - служба пересылки почтовых сообщений **368**
47.1 Введение ... 368
47.2 Настройка службы пересылки почтовых сообщений ... 368
47.3 Отправка сообщения или файла с помощью службы MAILER ... 369
47.4 Настройка службы watcher для отправки сообщений с помощью службы mailer ... 370

48 Служба автоконфигурирования IPv6 **371**
48.1 Секция настройки сервиса RA ... 371
48.2 Общие настройки интерфейса ... 372
48.3 Префикс ... 373
48.4 Маршруты ... 374
48.5 Клиентские хосты ... 375
48.6 Рекурсивный DNS-сервер ... 375
48.7 Список поиска DNS ... 376

49 Служба WATCHER **377**
49.1 Введение ... 377
49.2 Модули группы watch ... 377
49.3 Модули группы react ... 381

50 Служба CONNSTAT **386**

51 L2TP-туннели **390**
51.1 Введение ... 390
51.2 Настройка службы l2tp ... 391

51.3 Настройка клиентского интерфейса l2tp. 404
51.4 Информация о работе 406
51.5 Пример настройки 407

52 PPTP-туннели **410**
52.1 Введение 410
52.2 Настройка службы pptp 410
52.3 Настройка интерфейса pptp 411
52.4 Пример настройки 412

53 PPPOE-туннели **414**
53.1 Введение 414
53.2 Настройка службы pppoe 414
53.3 Настройка интерфейса pppoe 416
53.4 Пример настройки 417

54 Механизмы качества обслуживания (QoS) **418**
54.1 Классификация 418
54.2 Политика обслуживания policy-map 419
54.3 Пропускная способность интерфейса 421
54.4 Политика обслуживания prio-map 421
54.5 Политика обслуживания red-map 422
54.6 Политика обслуживания codel-map 423
54.7 Политика обслуживания fq-codel-map 423
54.8 Политика обслуживания sfq-map 424
54.9 Политика обслуживания gred-map 424
54.10 Ingress 425
54.11 Привязка политики к интерфейсу 425
54.12 Вложенные политики 426

55 Расширенная статическая маршрутизация **427**

56 Динамическая маршрутизация **429**
56.1 Списки 429
56.2 RIP 429
56.3 RIPNG (RIP для IPv6 сети) 436
56.4 OSPF 437
56.5 OSPF6 (OSPFv3 для IPv6 сети) 452
56.6 BGP 454
56.7 IS-IS 479
56.8 BFD 482
56.9 Карты маршрутов 485

57 MPLS - многопротокольная коммутация по меткам **491**
57.1 Введение 491
57.2 Статическая MPLS - маршрутизация 491
57.3 Динамическая MPLS - маршрутизация 492

58 VRF - Virtual Routing and Forwarding **493**
58.1 Введение 493
58.2 Базовая настройка 493
58.3 VRF-Lite 493
58.4 MPLS-L3VPN 494

59 Криптография **498**
59.1 Ключ доступа 498
59.2 Работа с USB-токенами/смарт-картами 504
59.3 Туннели Disec 505
59.4 Туннельные интерфейсы Ditun (ditun, ditap, ip6ditun, ip6ditap) 518
59.5 PSK (предварительно распределённые ключи) 526

59.6 PKI (закрытые ключи, сертификаты, COC, PKCS#10) 529
59.7 Туннели IPsec 553

60 Служба TLSPROXY **622**
60.1 Базовая настройка службы 622
60.2 Управление службой и диагностика 626
60.3 Другие настройки службы 628
60.4 Информирование сервера о клиенте 631
60.5 Причины неустановления соединения 631

61 Служба обновления CRL **632**
61.1 Настройки службы REVOCATION 633
61.2 Просмотр состояния службы 635

62 VRRP-кластер **636**
62.1 Основные понятия 636
62.2 Настройка кластера 636

63 Отказоустойчивый аппаратный кластер **639**
63.1 Требования к оборудованию 639
63.2 Подготовка к организации кластера 640
63.3 Настройки кластера 641
63.4 Получение информации о кластере 643
63.5 Пауза в работе кластера 644
63.6 Синхронизация настроек между маршрутизаторами 644
63.7 Дополнительные команды 645

64 Балансировка прерываний **646**

65 Обновление системы **649**
65.1 DIP-пакеты 649
65.2 Инфраструктура DIP 649
65.3 Установка обновления 651
65.4 Параметры загрузки 652
65.5 Конфигурация системы и данные 654
65.6 Привязка данных 655
65.7 Миграция ПО 655
65.8 Экспериментальный файл конфигурации 656
65.9 Резервная копия ПО 657

66 Обслуживание **658**
66.1 Резервное копирование 658
66.2 Самоконтроль целостности 660
66.3 Проверка файловых систем 660
66.4 Безопасная очистка внешнего носителя 661
66.5 Форматирование внешнего носителя 661
66.6 Сброс паролей в начальное значение 661
66.7 Сообщения об ошибках 662

67 Управление через COM-порты **664**
67.1 Настройка контроллера 664

68 Скрипты конфигурирования **665**
68.1 Описание встроенных функций и переменных 667
68.2 Пример скрипта для группового управления интерфейсами 670

69 Приложение **672**
69.1 Примеры конфигураций 672

1. Общие сведения

1.1 Функциональные возможности

Программно-аппаратный комплекс Dionis-NX (ПАК Dionis-NX или Dionis-NX) является средством защиты информации, реализующим функции межсетевого экрана и системы обнаружения вторжений.

Программное обеспечение программно-аппаратного комплека Dionis-NX (далее по тексту ПО ПАК Dionis-NX или ПО Dionis-NX) построено на базе ядра Linux 4.14.xx и является специализированной самодостаточной программой (прошивкой) для управления маршрутизатором, в рамках которой невозможно выполнение пользовательских программ.

Dionis-NX обладает следующими функциональными возможностями.

1.1.1 Сетевые интерфейсы

Поддерживаются следующие типы интерфейсов:

- Ethernet (большой спектр поддерживаемых плат, с поддержкой VLAN и QinQ);
- HDCL (E1, платы parabel, в том числе в unframed режиме);
- bond (агрегирование каналов в режимах: rr, active-backup, 802.3ad, tlb, alb и другие, в разных режимах мониторинга состояний линий связи;
- bridge (работа в режиме сетевого коммутатора с поддержкой stp, igmp-snooping и igm-routing);
- ditun (поддержка туннельных интерфейсов L3 в режиме “IPsec-Фактор” (далее - в режиме шифрования DISEC), с поддержкой icmp проб, автоопределением параметров (работа через NAT));
- ditap (поддержка туннельных интерфейсов L2 в режиме шифрования DISEC, с поддержкой icmp проб, автоопределением параметров (работа через NAT));
- gre (поддержка туннельных интерфейсов L3 в режиме GRE с поддержкой пинг-проб);
- gretap (поддержка туннельных интерфейсов L2 в режиме GRE с поддержкой пинг-проб);
- ip6ditun/ip6ditap (поддержка туннельных интерфейсов L2/L3 с шифрование ГОСТ (IPv6));
- ip6gre/ip6gretap (поддержка туннельных интерфейсов L2/L3 GRE IPv6);
- vxlan/ip6vxlan (поддержка технологии VXLAN);
- l2tp (реализована клиентская и серверная поддержка протокола L2TP);
- loopback (интерфейс - петля);
- modem (поддержка 3G/4G модемов);
- pppoe (реализована клиентская и серверная поддержка протокола PPPoE);
- pptp (реализрвана клиентская и серверная поддержка протокола PPTP);
- ppp (реализована клиентская и серверная поддержка PPP);
- vpn (реализована клиентская и серверная поддержка openvpn);
- serial controller (поддержка USB и COM портов).

Все интерфейсы поддерживают:

- произвольный набор адресов IPv4 (в т.ч. по DHCP) и IPv6;
- режим link-detect;
- правила фильтрации;
- правила NAT;
- правила модификации пакетов;
- правила QoS;
- полисер;
- правила RPS;
- L2 адрес;
- режим зеркалирования;
- режим отправки статистики по netflow (1, 5 и 9 версии протокола);
- mtu/multicast/arp и другие свойства.

1.1.2 Статическая маршрутизация

- Поддерживается статическая маршрутизация TCP/IP (v4 и v6) с метриками и автоматическим подниманием/опусканием маршрута в зависимости от состояния интерфейса (link-detect);
- Поддерживается расширенная статическая маргрутизация IPv4/IPv6 (policy route) со следующими возможностями:
  - поддержка icmp проб;
  - критерий выборки: ttl;
  - критерий выборки: адрес источника и/или адрес назначения;
  - критерий выборки: порты назначения и/или источника;
  - критерий выборки: интерфейс на который был принят данный сетевой пакет;
  - критерий выборки: мандатные метки МСВС;
  - критерий выборки: частота соединений;
  - критерий выборки: состояние соединения;
  - критерий выборки: состояние флагов TCP;
  - критерий выборки: MAC адрес источника;
  - критерий выборки: время/дата;
  - критерий выборки: tos/dscp;
  - критерий выборки: класс трафика;
  - другие критерии выборки;
  - маршрутизация осуществляется для первого пакета потока, все остальные пакеты потока маршрутизируются по выбранному правилу, что позволяет достичь высокой скорости маршрутизации даже для сложных правил отбора;
- Поддержка явного задания входов ARP таблицы IPv4;
- Поддержка явного задания входов NEIGHBOUR таблиц IPv6.

1.1.3 Динамическая маршрутизация

- Поддержка IGP;
- Поддержка OSPFv2;
- Поддержка OSPFv3;
- Поддержка BGPv4, v4+, v4-;
- Поддержка RIP/RIP2;
- Поддержка RIPNG;
- Поддержка LDP.

1.1.4 Поддержка мультикаст-трафика

- Поддержка статической мультикаст-маршрутизации;
- Поддержка DVMRP;
- Поддержка IGMP;
- Поддержка PIM.

1.1.5 Фильтрация

- Правила фильтрации задаются в виде списков;
- Поддерживается передача управления от спискам к спискам в режиме безусловного перехода или с возвратом управления;
- Правила фильтрации могут применяться на интерфейсах (с явным заданием направления трафика), на прием, на отправку и на маршрутизацию;
- Поддержка протоколов IPv4/IPv6;
- Поддерживается широкий набор критериев отбора и их комбинаций, применительно к отдельному интерфейсу или системе в целом:
  - протокола;
  - адресов и портов источника/назначения;
  - mac-адреса источника;
  - текущего времени;
  - поля TOS/DSCP;
  - содержимого поля данных пакета;
  - состояния соединения;
  - состоянию флагов TCP;
  - содержимому;
  - ttl;
  - с учетом недавней сетевой активности (списки недавних адресов и событий);
  - и другие критерии (около 20).

1.1.6 NAT

- Поддержка различных вариантов трансляции IP/IPv6-адресов (SNAT/DNAT):
  - SNAT;
  - DNAT;
  - MASQUARADE (вариант SNAT);
  - SMAP (SNAT для сети);
  - DMAP (DNAT для сети);
  - исключение трафика из NAT.

1.1.7 Модификация трафика

- Правила модификации трафика задаются в виде списков;
- Спики могут быть установлены в те же места, что и списки фильтрации;
- Критерии отбора такие же, как и у списков фильтраций;
- Поддержка протоколов IPv4/IPv6;
- Поддерживаются следующие операции:
  - модификация TOS/DSCP;
  - модификация MSS;
  - модификация DF;
  - модификация ttl;
  - модификация меток BSO;
  - модификация и анализ CoS.

1.1.8 Поддержка и отслеживание соединений и преобразование сетевых адресов (NAT)

- Для передачи audio, video и т.д.;
  - sip;
  - ftp;
  - pptp;
  - snmp;
  - tftp;
  - q931;
  - netbios_ns;
  - irc;
  - sane;
  - h245;
  - amanda;
  - ras.

1.1.9 Классификация трафика

- Dionis-NX поддерживает классификатор трафика, который позволяет относить один и тот же поток к разными классам (до 64).
- Классификатор трафика может быть использован для шейпинга и модификации трафика, а также его маршрутизации;
- Число списков классификации не ограничено, но одновременно может быть использовано 64 классов;
- Поддерживаются широкие критерии отбора.

1.1.10 Криптографическая защита

- Поддержка криптографической защиты данных, передаваемых по каналам связи сетей общего пользования, использующих протоколы семейства TCP/IP (v4/v6) (компоненты СКЗИ):
  - создание и поддержка статических криптотуннелей между узлами Dionis-NX с шифрованием и имитозащитой передаваемых IP/IPv6-пакетов с инкапсуляцией их в протокол “IP в IP” или “UDP”;
  - обеспечение функционирования туннельных интерфейсов или туннелей без создания интерфейсов;
  - реализация протоколов IPSEC ГОСТ (IKEv1, ESP), позволяющая создавать статические и динамические туннели IPSEC между узлами Dionis-NX.

1.1.11 MPLS

- Протокол LDP (Label Distribution Protocol) – протокол распространения меток согласно RFC-3036 и RFC-5036;
- Технология MPLS (Multi-Protocol Label Switching): статическое назначение меток и динамическое, используя LDP;
- Поле MPLS TC согласно RFC-5462 – Multi-Protocol Label Switching Label Stack Entry: “EXP” Field renamed to TC (Traffic Class) Field;
- Установка LSP (Label Switching Path) посредством LDP согласно RFC-3212, а также статическое описание пути коммутации меток – LSP;
- Организация L2 туннелей поверх MPLS с сигнализацией LDP с помощью интерфейсов Ditap, объединённых в Bridge-интерфейс с физическими Ethernet интерфейсами (с шифрованием и без шифрования);
- Организация L3 туннелей поверх MPLS с сигнализацией LDP с помощью интерфейсов Ditun, используя правила маршрутизации в туннельный интерфейс Ditun (с шифрованием и без шифрования).

1.1.12 QoS

- Поддежка полисера (ограничение полосы на вход/выход);
- Поддержка иерархичного шейпера HTB (любой глубины вложенности);
- Поддержка codel/fqcodel;
- Поддержка GRED;
- Поддержка RED;
- Поддержка SFQ;
- Поддержка prio (в режиме до 16 очередей, с заданием правил по tos или с помощью классификатора);
- Поддержка комбинирования политик HTB и prio с другими политиками в 2-х уровневой иерархии.

1.1.13 Протоколирование

- Поддержка протоколирования событий фильтрации IP/IPv6-пакетов;
- Поддержка протоколирования цикла обработки IP/IPv6-пакетов при прохождении их через маршрутизатор (трассировка);
- Поддержка настраиваемого механизма ротации журналов проходимого трафика;
- Поддержка выборочного протоколирования и механизмов оповещения;
- Протоколирование выполняется в реальном времени с гарантией того, что информация не будет потеряна. Возможна запись полного содержимого трафика или только заголовков;
- Режим слежения за системными журналами и реагирование на заданные администратором события (служба watcher):
  - генерации сообщения в системный журнал;
  - отправки сообщения по электронной почте;
  - выполнения команды по ssh;
- Поддержка звукового оповещения (для сообщений высокой важности).

1.1.14 Служба обнаружения/предотвращения вторжений

- Работа в режиме IDS/IPS;
- Совместная работа с программным комплексом Diamant:
  - отправка сообщений об атаках (работа в режиме сенсора);
  - централизованное обновление правил;
- Возможность выполнения заданных действий при срабатывании правил (через службу watcher);
- Протоколировние заданного количества пакетов, предшествующих сработавшему правилу.

1.1.15 Другие службы

- Поддержка сервера доменных имен (DNS)- работа в режиме первичного или вторичного DNS-сервера, наличие DNS-кэша и т.д.;
- Поддержка сервера динамической конфигурации узла (DHCP)- начальное конфигурирование рабочих станций локальных сетей (включая возможность задания нескольких шлюзов и севис DHCPRELAY);
- Поддержка прокси-сервера HTTP/FTP с возможностями прозрачного перехвата и фильтрации трафика;
- Поддержка сервера и клиента удаленного доступа SSH - обеспечение доступа к комплексу для управления им с удаленной консоли;
- Поддержка сервера/клиента синхронизации часов по сети (NTP) - служба синхронизации времени;
- Поддержка сервера удаленного мониторинга (SNMP);
- Поддержка управления устройством через WEB по HTTP(S) протоколу;
- Службы для оценки пропускной способности канала и SLA;
- Служба LLDP;
- Служба XMPP;
- Служба RA IPv6 (Router Advertisement Daemon);
- Реализация VRRP (v2/v3);
- Служба Telnet.

1.1.16 Обслуживание

- Контроль целостности конфигурации и прошивки;
- Поддержка процедур резервного архивирования и восстановления;
- Поддержка функционирования узлов Dionis-NX в режиме отказоустойчивого аппаратного кластера с синхронизацией состояния соединений.

В качестве основной системы управления маршрутизатором используется интерфейс командной строки. Для настройки основных функций существует возможность использовать web-интерфейс.

1.2 Дополнительные технические данные

Маршрутизаторы, работающие под управлением ПО Dionis-NX, для повышения надежности могут дублироваться . Кроме стандартного способа дублирования, основанного на применении протокола VRRP (п. 62), Dionis-NX представляет возможность аппаратного дублирования маршрутизаторов (создание отказоустойчивого кластера, п. 63). Этот режим обеспечивает высокую скорость переключения с основного на резервный маршрутизатор в случае сбоя. Количество маршрутизаторов, используемых в кластере “горячего” резервирования - два. Оба маршрутизатора, работающих в составе кластера,

должны быть одинаковыми и иметь одинаковую конфигурацию, и должны быть соединены между собой. По этому соединению от работающего маршрутизатора передается информация, характеризующая состояние компонента TCP/IP (и не происходит, например, передачи логов).

Число поддерживаемых сетевых интерфейсов и число каналов обслуживания прикладных сервисов TCP/IP зависит от аппаратной части (объем ОЗУ и число разъемов на материнской плате).

Использование программно-аппаратного комплекса подлежит лицензированию. Лицензия может ограничивать использование программных и аппаратных возможностей оборудования.

Максимальное число одновременно установленных TCP/IP-соединений зависит от конфигурации аппаратного обеспечения.

2. Установка с флеш-диска

Установка ПО Dionis-NX обычно осуществляется производителем оборудования, однако может производиться и с помощью специального загрузочного флеш-диска. Кроме установки ПО, диск может использоваться и для других целей: сохранение конфигурации, восстановление конфигурации, запись новых версий ПО, создание резервных копий и т.д. В случае утери такого диска клиент имеет возможность получения образа диска. Установочный флеш-диск содержит один раздел с файловой системой FAT32.

Флеш-диск в общем случае может содержать любые файлы и доступен как в ОС Linux, так и в ОС Windows. Директория install-images имеет специальное значение. В этой директории могут храниться сжатые образы системы Dionis-NX. Файлы со сжатыми образами ПО обычно имеют вид dionisnx-<версия>.<архитектура>.dip.

Как указано выше, на флеш-диск может быть выполнено резервное копирование текущей конфигурации и данных системы Dionis-NX. В этом случае на флеш-диске появится специальная директория dionisnx-backup.

Несмотря на то, что, как правило, начальная установка ПО Dionis-NX выполняется производителем оборудования, возможны специальные ситуации, когда такая установка выполняется пользователем, например, при отсутствии связи с сетью Интернет. В этом случае пользователь должен иметь полностью сформированный установочный флеш-диск с требуемой версией ПО, полученный от производителя до начала установки. Каждый образ ПО Dionis-NX предназначен для конкретного оборудования. Невозможно использовать на данной машине образ ПО, предназначенный для другого экземпляра аналогичной машины. Для установки ПО Dionis-NX следует загрузиться с установочного флеш-диска. Если маршрутизатор оборудован специальной платой “Сторож”, необходимо полностью обесточить маршрутизатор и перевести эту плату в технологический режим (режим SE), иначе эта плата отключит клавиатуру и USB-шину на время загрузки, и, соответственно, не удастся загрузиться с внешнего носителя.

После загрузки с установочного флеш-диска на экран будет выведен список возможных действий (для управления используется псевдографический интерфейс):

- Установка системы
- Обслуживание системы ->
- Выбор установочного диска
- Выбор целевого диска
- Журналирование ->
- Идентификатор платформы
- Диагностическая информация ->
- Перезагрузка
- Выключение компьютера

После установки ПО Dionis-NX необходимо перевести плату “Сторож” обратно в рабочий режим (режим JL).

Пункт “Установка системы” позволяет установить ПО Dionis-NX на жесткий диск. На одном жестком диске может быть установлено сразу несколько экземпляров ПО Dionis-NX, однако только один из установленных экземпляров будет активным в данный момент времени. Вновь установленная версия

автоматически становится активной. Если старая версия ПО больше не требуется, ее можно впоследствии удалить с жесткого диска.

При выборе пункта “Установка системы”, программа предложит выбрать диск (как правило, маршрутизатор имеет единственный диск), на который будет произведена установка ПО Dionis-NX. Затем установщик предложит выбрать образ ПО Dionis-NX, который следует установить. Как было сказано выше, в общем случае установочный флеш-диск может содержать несколько образов с различными версиями ПО. Если на диске маршрутизатора уже присутствуют установленные системы, то в следующем диалоговом окне можно указать, наследовать ли данные (слот данных) более ранних версий ПО для текущей установки.

После выполнения всех описанных действий, будет произведена установка выбранного ПО на жесткий диск, а затем пользователю будет предложено извлечь установочный флеш-диск и перезагрузить систему.

3. Основы работы с интерфейсом командной строки

В качестве основного средства управления маршрутизатором используется интерфейс командной строки. После входа в систему пользователь может набирать на клавиатуре команды, которые выполняют различные действия, в т.ч. и меняют конфигурацию устройства.

Команды вводятся в ответ на приглашение системы, например, такое:

```
adm@DionisNX> _
```

или

```
adm@DionisNX# _
```

Здесь DionisNX - имя узла

Команды Dionis-NX в общем случае состоят из двух частей: из имени команды и параметров. Параметры отделяются от имени команды и друг от друга пробелами. В команде может не быть ни одного параметра.

Для удобства пользователей команды в Dionis-NX разделены на группы по функциональному назначению. Каждая группа может содержать подгруппы (группы следующих уровней). В соответствии с этим имя команды может быть составным - состоять из нескольких “слов”: первое слово - имя первой группы команд, второе - имя группы следующего уровня и т.д. Слова в команде разделяются пробелами.

При вводе и редактировании команд можно использовать следующие клавиши:

- <Tab> или <Ctrl^I> - для автоматического дополнения имени команды или параметра (при однозначном варианте сразу выполняется дополнение; если возникает возможность выбора, выводится список вариантов);
- <?> - для вывода на экран списка команд или параметров, доступных в настоящий момент; список выводится вместе с краткой справкой по этим командам/параметрам;
- <стрелка вверх> - для вывода на экран предыдущих команд (для просмотра или повторного ввода);
- <Shift+PgUp/PgDn> - для постраничного просмотра содержимого экрана;
- <Ctrl^Z> - выход на уровень выше в дереве вложенных настроек (режим конфигурации). Соответствует выполнению команды “exit”;
- <Ctrl^Space> - просмотр конфигурации текущего уровня настроек (режим конфигурации). Соответствует выполнению команды “show”;
- <Ctrl^C> - отмена ввода и переход на новую строку;
- <Home> или <Ctrl^A> - переход в начало строки;
- <End> или <Ctrl^E> - переход в конец строки;
- <Del> или <Ctrl^D> - удаление текущего символа;
- <Backspace> или <Ctrl^H> - удаление предыдущего символа;
- <Ctrl^L> - очистка экрана;
- <Ctrl^J> или <Ctrl^M> - ввод. Соответствует нажатию клавиши ;

- <Ctrl^W> - удаление слова;
- <Ctrl^K> - удаление всей строки справа от курсора и копирование удаленной части в буфер;
- <Ctrl^U> - удаление всей строки слева от курсора и копирование удаленной части в буфер;
- <Ctrl^Y> - вставка из буфера.

Если по какой-либо команде на экран выводится длинный текст (командная строка оказывается за пределами экрана), то при просмотре такого текста можно использовать следующие клавиши:

- <стрелка вверх>/<стрелка вниз> - для перехода на предыдущую/последующую строку;
- <PgUp>/<PgDown> - для постраничного просмотра;
- <!—-> - для выхода в режим командной строки.

Заканчивается ввод команды нажатием клавиши .

Если строка начинается с символа "!", то она содержит комментарий.

3.1 Режимы работы с системой

Все действия в ПО Dionis-NX всегда производятся от имени какой-либо учетной записи. Перед началом работы пользователь должен войти в систему - ввести свое имя (имя учетной записи) и затем ввести свой пароль.

По умолчанию в системе существуют две учетные записи - учетная запись для получения консольного доступа к системе (учетная запись оператора с именем "cli") и учетная запись для администрирования ("adm").

В заводских настройках ПО Dionis-NX (если не было специальных указаний заказчика) для этих учетных записей установлены пароли, совпадающие с их именами (cli и adm соответственно).

При первом входе администратора adm в систему ему будет предложено сменить оба заводских пароля (cli и adm) на другие, которые и будут использоваться в дальнейшей штатной работе.

Для учетных записей предусмотрена проверка выполняемых команд и параметров которые контролируются на допустимые значения с помощью CLI (command line interface) dish. Запуск от учётных записей произвольных команд уровня ОС не предусмотрен.

Администратор имеет возможность создать любое количество учетных записей администраторов (возможно, с разными правами доступа), поэтому в дальнейшем будет говориться об учетной записи администратора.

При входе в систему под учетной записью для получения консольного доступа к системе система предоставляет доступ к командам непривилегированного режима - это только часть информационных команд. С помощью команды "enable" можно перейти в привилегированный режим, но при этом потребуется указать имя учетной записи администратора и ввести пароль.

Администратор в системе имеет доступ к командам привилегированного режима enable. В этом режиме доступны команды управления, не меняющие конфигурацию системы. Для входа в режим конфигурации используется команда "configure terminal". Доступность команд конфигурирования для учетной записи администратора определяется совокупностью прав этой учетной записи.

В дальнейшем изложении будут использоваться следующие обозначения режимов командного интерфейса:

- user - режим непривилегированного пользователя (оператора);
- enable - режим администратора (разрешены команды, не меняющие конфигурацию системы);
- configure - основной режим конфигурирования (изменение текущей конфигурации);
- (остальные) - являются вложенными режимами конфигурирования.

Вложенные режимы конфигурирования возникают после задания команды конфигурирования какого-либо объекта, например, интерфейса. Конфигурирование этого объекта может использовать специфический набор команд, например, команды конфигурирования интерфейса отличаются для разных интерфейсов. Доступность этих команд для учетной записи администратора определяется в рамках ролевой модели (п. 7).

Из режима enable доступны все команды режима user.

Из режимов конфигурирования (основного и вложенных) можно выполнить все команды режима enable, снабдив их префиксом “do”. Например, команда просмотра текущей версии и контрольных сумм имеет следующий формат:

В режиме enable:

```
# show version
```

В режиме configure:

```
(config)# do show version
```

3.2 Виртуальные консоли

Для удобства работы с системой реализована возможность одновременной работы на нескольких виртуальных консолях. Переключение с одной виртуальной консоли на другую выполняется нажатием клавиш <Alt+Fn>. На десятую виртуальную консоль (<Alt+F10>) (после ее активации клавишей “Ввод”) выводится информация системы мониторинга (п. 12, Журналирование и отладка).

3.3 Пространства имен

ПО Dionis-NX может работать с файлами, расположенными:

- в файловом пространстве маршрутизатора;
- на внешних носителях;
- на FTP/TFTP-серверах и HTTP-серверах.

Файловое пространство системы Dionis-NX разделено на несколько непересекающихся пространств:

- Локальное дисковое пространство файлов работающей версии ПО (слот данных системы);
- Дисковое пространство файлов маршрутизатора, разделяемое между всеми версиями ПО, установленными в маршрутизаторе;
- Логи (журналы);
- running-config, startup-config, default-config - фиксированные файлы конфигураций (п. 3.4) (файловые объекты с фиксированными именами).

Более подробно описание слотов данных и разделяемого между всеми версиями ПО хранилища приведен в разделе п. 65.2.

Ниже приведен список названий возможных пространств файлов (они используются как префиксы в командах, манипулирующих файлами для идентификации файлового пространства, в котором файл находится):

cdrom<число>:	носитель CD-ROM/DVD-ROM
floppy<число>:	носитель на гибких дисках
flash<число>[.<раздел>]:	сменный носитель
file:	локальное файловое пространство работающей версии ПО (префикс, используемый по умолчанию, если он не указан)
share:	пространство файлов маршрутизатора, разделяемое между всеми версиями ОС, установленными в маршрутизаторе
config:	пространство конфигурации (обычно, не указывается явно)
log:	файловое пространство журналов
ftp:	файлы, доступные по протоколу FTP
ftpserver:	пространство для сервиса FTP
tftp:	файлы, доступные по протоколу TFTP
http:	файлы, доступные по протоколу HTTP
https:	файлы, доступные по протоколу HTTPS
cert:	пространство для пользовательских сертификатов (например, сгенерированных командой ssl cert …)
ro:	пространство с правами только на чтение, в нем находятся snpm_mibs, а так же примеры скриптов и конфигураций
tmp:	пространство файлов, разделяемое между всеми пользователями в маршрутизаторе

Если обращение в команде происходит к файлам конфигурации, то никакой префикс перед именем файла конфигурации не ставится.

3.4 Работа с конфигурациями

Конфигурация представляет собой последовательность команд и определяет настройку системы.

В Dionis-NX существует три вида конфигурации:

default-config	заводская конфигурация системы
running-config	действующая конфигурация
startup-config	стартовая конфигурация

Заводская конфигурация (default-config) определяет заводские настройки системы. Она доступна только на чтение. Заводская конфигурация может быть использована для сброса всех текущих настроек Dionis-NX (установленных в процессе работы) и возврата к первоначальным заводским настройкам.

Действующая конфигурация (running-config) определяет текущие настройки системы (настройки, которые действуют в данный момент). Если администратор вводит команду в режиме configure, то она в случае ее успешного выполнения немедленно влияет на действующую конфигурацию.

При выходе из системы/при перезагрузке действующая конфигурация будет потеряна. При необходимости её можно сохранить командой копирования (см. ниже).

Стартовая конфигурация предназначена для создания действующей конфигурации после включения/перезагрузки системы. Работа системы всегда начинается с выполнения команд стартовой конфигурации; успешно выполненные команды стартовой конфигурации автоматически заносятся в действующую конфигурацию. В результате конфигурация running-config через некоторое время после начала работы системы становится эквивалентной конфигурации startup-config, за исключением тех команд из startup-config, которые по каким-то причинам завершились с ошибкой и вследствие этого не были добавлены в running-config. Если в ходе дальнейшей работы администратор выполнит команды конфигурирования (например, вводя их с консоли), то стартовая и действующая конфигурация станут различаться.

3.5 Просмотр и копирование конфигураций

Просмотреть любую из конфигураций можно с помощью команды “show” с соответствующим параметром (команда режима enable):

```
# show running—config
# show startup—config
# show default—config
```

Команда “show” без параметров, выполненная в режиме enable, эквивалентна команде “show running-config” и показывает действующую конфигурацию.

В качестве аргумента команды “show running-config” вы можете указать строку поиска. В таком случае, будет выведена информация о командах конфигурации первого уровня, попадающих под шаблон поиска. Например, для вывода конфигурация службы DNS:

```
# show running—config ”service dns”
```

В качестве строки поиска используются регулярные выражения POSIX.

В системе реализован целый ряд команд, которые позволяют из режима enable просмотреть конкретные части действующей конфигурации, например:

“show interface <тип> <номер> config”	просмотр настроек интерфейса
“show ip access-list <имя> config”	просмотр списка доступа
“show ip route config”	просмотр статических маршрутов

Команда “do show” из основного режима конфигурации эквивалентна команде “do show running-config” и показывает всю действующую конфигурацию. Если команда “do show” вызывается в одном из вложенных режимов конфигурации, то она покажет только часть действующей конфигурации, относящейся к данному режиму/конфигурируемому объекту.

Конфигурациями можно оперировать с помощью команды “copy” в режиме enable (или “do copy” в режиме configure) (п. 3.7.2).

Как было сказано выше, действующая конфигурация не сохраняется при завершении работы системы. Кроме того, может возникнуть необходимость иметь несколько вариантов действующей конфигурации. Для сохранения действующей конфигурации ее можно скопировать в стартовую конфигурацию, а также - в файл.

Стартовую конфигурацию можно заменить конфигурацией из файла. Заводскую конфигурацию можно скопировать в стартовую. Любую конфигурацию можно скопировать в файл.

Примеры команд:

“copy running-config startup-config”	Сохранение действующей конфигурации в стартовую конфигурацию
“write”	Эквивалент команды “copy running-config startup-config”
“copy running-config <имя_файла>”	Сохранение текущей конфигурации в файле
“write <имя_файла>”	Эквивалент команды “copy running-config <имя_файла>”
“copy startup-config <имя_файла>”	Копирование стартовой конфигурации в файл
“copy <имя_файла> startup-config”	Замена стартовой конфигурации конфигурацией из файла

С помощью команды “copy” можно задать выполнение команд из стартовой конфигурации, из заводской конфигурации, а также из файла с последующим изменением действующей конфигурации.

Формат соответствующих команд:

“copy startup-config running-config”	Выполнение команд из startup-config (поверх старой running-config)
“copy <имя_файла> running-config”	Выполнение команд из файла (поверх старой running-config)

Копирование команд из файла в действующую конфигурацию является достаточно опасным. Команды из копируемого файла сразу же будут выполняться, и их действие может повлиять на результаты

уже выполненных команд действующей конфигурации.

3.6 Команды режима configure и их влияние на действующую конфигурацию

Команды в режиме configure по способу их воздействия на действующую конфигурацию делятся на три основных типа:

- уникальные команды;
- списковые команды;
- отменяющие команды.

Для всех типов команд действует правило: если команда не нарушает консистентность конфигурации – она меняет running-config. Реальное применение действующей конфигурации может произойти позже. Таким образом, выполнение команды режима configure это конфигурация системы, а применение конфигурации в рамках ПО происходит асинхронно в те моменты времени, когда это необходимо.

При выполнении команды возможно получение диагностических сообщений следующих типов:

- Info: команда выполнилась успешно, предоставляется дополнительная информация по выполнению команды;
- Warning: команда выполнилась успешно, но с некоторыми замечаниями, которые могут означать логическую ошибку администратора;
- Error: при применении команды возникла ошибка. Невозможно применить данную конфигурацию;
- Fatal: критическая ошибка.

Уникальные команды после выполнения добавляются в текущую конфигурацию. Если аналогичная команда уже существовала в конфигурации, то она будет заменена командой с новыми параметрами.

Списковые команды после выполнения добавляются в текущую конфигурацию. Старые аналогичные команды не удаляются. В результате в конфигурации получается список аналогичных команд с разными параметрами. В некоторых случаях порядок команд в списке может иметь значение.

Отменяющие команды после выполнения не добавляются в текущую конфигурацию, они служат только для удаления из конфигурации других команд.

Пример уникальной команды (“hostname <имя_хоста>” - заменить имя хоста):

```
(config)# hostname Router1

(config)# do show
!по команде ”просмотреть” выводится вся действующая конфигурация
...
hostname Router1
...
```

```
(config)# hostname DionisNX
(config)# do show
...
hostname DionisNX
...
!в конфигурации заменена команда ”hostname” с параметром ”<имя_хоста>”
```

Пример списковой команды (“ip secondary-address <IP-адрес>” - задать вторичный IP-адрес интерфейса):

```
(config)# interface ethernet 0
!выполнен переход в следующий (вложенный) режим конфигурации
(config—if—ethernet0)# do show
!по команде ”просмотреть” на экран выводится только часть конфигурации
ip address 192.168.56.3/24
enable
(config—if—ethernet0)# ip secondary—address 192.168.56.4/24
(config—if—ethernet0)# do show
ip address 192.168.56.3/24
ip secondary—address 192.168.56.4/24
enable
(config—if—ethernet0)# ip secondary—address 192.168.56.5/24
(config—if—ethernet0)# do show
ip address 192.168.56.3/24
ip secondary—address 192.168.56.4/24
ip secondary—address 192.168.56.5/24
enable

!в конфигурацию добавлены две аналогичные команды с именем ”ip secondary—address” и
        разными параметрами
```

Пример отменяющей команды (“no ip secondary-address <IP-адрес>” - удалить указанный вторичный адрес интерфейса):

```
(config—if—ethernet0)# no ip secondary—address 192.168.56.5/24
DionisNX(config—if—ethernet0)# do show
ip address 192.168.56.3/24
ip secondary—address 192.168.56.4/24
enable
(config—if—ethernet0)# no ip secondary—address 192.168.56.4/24
(config—if—ethernet0)# do show
ip address 192.168.56.3/24
enable
```

Режим конфигурирования организован по принципу “дерева”. Из основного режима можно перейти в первый вложенный режим (с помощью команд), затем во второй и т.д. На каждом уровне вложенности формируется своё приглашение на ввод команды - приглашение текущего (вложенного) режима конфигурации. Будем называть его приглашением “текущего контекста”.

Чтобы выйти из вложенного режима в вышестоящий, можно ввести команду “exit” или команду из любого режима меньшего уровня вложенности - в этом случае будет выполнен переход на этот уровень вложенности. Команды всех уровней вложенности одной ветви дерева не пересекаются по именам.

Внутри одного уровня вложенности команды заносятся в действующую конфигурацию в определенном порядке. Очередность определяется, в первую очередь, приоритетом команды (приоритет является атрибутом команды, присваивается командам разработчиками системы). Вне зависимости от типа команды, сначала располагается команда с более высоким приоритетом. Если приоритеты у двух команд одинаковые, то для их размещения в конфигурации (внутри одного уровня вложенности) действуют следующие правила:

- Если из двух команд одна или обе уникальные, то они располагаются в алфавитном порядке;
- Если обе команды являются списковыми и хотя бы у одной из них неважен порядок ввода, то они располагаются в алфавитном порядке;
- Если обе команды являются списковыми и у обеих важен порядок ввода, то они располагаются в порядке ввода.

Это означает, что при сохранении, например, действующей конфигурации в файл, команды могут в нем оказаться не в том порядке, как они выполнялись при добавлении их в действующую конфигурацию.

Пример вложенных режимов конфигурирования (настройка 4-х сетевых интерфейсов)

```
# configure terminal

! — конфигурируем первый интерфейс
(config)# interface ethernet 0
(config—if—ethernet0)# enable
(config—if—ethernet0)# ip address 1.1.1.1/24
(config—if—ethernet0)# exit

! — вышли в вышестоящий режим командой exit
(config)# _

! — конфигурируем следующий интерфейс
(config)# interface ethernet 1
(config—if—ethernet1)# enable
(config—if—ethernet1)# ip address 2.2.2.2/24

! — переходим на предыдущий уровень заданием команды вышестоящего режима — например,
        команды ”ip forwarding” (включить транзит).
(config—if—ethernet1)# ip forwarding
(config)# _

! — конфигурируем следующий интерфейс
(config)# interface ethernet 2
(config—if—ethernet2)# enable
(config—if—ethernet2)# ip address 3.3.3.3/24
```

```
! — сразу переходим к конфигурации следующего интерфейса заданием команды вышестоящего
        режима
(config—if—ethernet2)# interface ethernet 3

! — система выполнила неявный переход на верхний уровень и сразу переход во вложенный
        режим, но к другой ветви ”дерева” команд
(config—if—ethernet3)# enable
(config—if—ethernet3)# ip address 4.4.4.4/24

! — просматриваем часть действующей конфигурации в данном контексте
(config—if—ethernet3)# do show
ip address 4.4.4.4/24
enable
! — просматриваем всю конфигурацию
(config—if—ethernet3)# exit
(config)# do show
!
hostname DionisNX
!
interface ethernet 0
 ip address 1.1.1.1/24
 enable
!
interface ethernet 1
 ip address 2.2.2.2/24
 enable
!
interface ethernet 2
 ip address 3.3.3.3/24
 enable
!
interface ethernet 3
 ip address 4.4.4.4/24
 enable
!
ip forwarding
```

Из примера видно, как меняется приглашение системы на ввод команды в зависимости от режима конфигурирования.

Приглашение основного режима конфигурирования имеет вид:

```
(config)#
```

приглашения следующего (первого) уровня вложенности имеют вид:

```
(config—if—ethernet0)#
(config—if—ethernet1)#
```

```
и т.д.
```

Приглашение каждого “текущего контекста” указывает администратору, в каком режиме он находится в данный момент.

Из примера также видно, что при просмотре полной конфигурации вложенные команды выводятся на экран с отступами.

Обратите внимание, что команда “ip forwarding” оказалась в конце конфигурации - она имеет самый низкий приоритет.

3.7 Команды работы с файлами

Команды работы с файлами доступны только администратору из режима enable.

3.7.1 Просмотр файлов

Для просмотра файлов используется команда “ls”.

По команде с параметром “*”на экран выводится список доступных внешних устройств и доступных пространств файловой системы, например:

```
DionisNX# ls *
config:
 file:
flash0:
ftp:
http:
```

Команда “ls” с названием устройства в качестве параметра позволяет просмотреть содержимое указанного устройства.

Например, для просмотра файлов на сменном носителе служит команда:

```
DionisNX# ls flash0:
```

Для просмотра списка файлов в директории PUB на ftp-сервере (пусть адрес сервера 192.168.33.160) служит команда:

```
DionisNX# ls ftp://192.168.33.160/pub
```

Если параметр команды “ls” не содержит названия устройства, то подразумевается, что файл находится в локальном файловом пространстве. Т.е для просмотра списка файлов в локальном файловом пространстве можно использовать команду с параметром “file:”, “/” или команду без параметра:

```
DionisNX# ls file:
```

или

```
DionisNX# ls
```

или

```
DionisNX# ls /
```

3.7.2 Копирование

Для копирования файлов используется команда: “copy <откуда> <куда>”.

Использование этой команды для работы с конфигурациями описаны выше (п. 3.4. Кроме путей, включающих в себя имена устройств, команда copy может принимать следующие имена файлов:

default-config	заводская конфигурация системы по-умолчанию (только для чтения)
running-config	действующая конфигурация
startup-config	сохраненная конфигурация

Если аргумент команды copy не содержит префикса устройства, то подразумевается file: Таким образом, администратор с помощью команды copy может поддерживать набор конфигураций, копировать их на внешние носители или получать с внешних носителей. Приведем типовые примеры использования команды copy.

Сохранение конфигурации на флеш-носитель:

```
DionisNX(config)# do copy running—config flash0:/dionisnx—config
```

Получение конфигурации с ftp-сервера:

```
DionisNX(config)# do copy ftp://192.168.33.160/pub/dionisnx—config startup—config
```

Копирование конфигурации в локальное файловое пространство:

```
DionisNX(config)# do copy startup—config config
```

Копирование журналов:

```
DionisNX(config)# do copy log:/auth.log flash0:/log
```

Просмотр списка файлов в локальном файловом пространстве:

```
DionisNX(config)# do ls file:
```

При использовании команды copy для некоторых типов устройств допустимо указывать в качестве источника и приемника каталоги. При этом каталог-приемник интерпретируется следующим образом. Если каталог-приемник не существует – то исходный каталог будет скопирован под именем каталога-приемника. Если каталог-приемник существует – то содержимого исходного каталога будет скопировано в каталог-приемник.

При копировании по протоколу SSH (п. 37) используется не команда copy, а команды ssh get и ssh put (п. **??**sshgetput)}.

3.7.3 Контрольная сумма файла

Просмотр контрольной суммы (ГОСТ-2489) файла осуществляется следующей командой:

```
DionisNX(config)# do gostsum file:/example.tar.gz
```

3.7.4 Другие команды

К другим командам относятся:

rm <что>	удаление файла или каталога (рекурсивно)
mkdir <что>	создание каталога
cat <что>	вывод содержимого файла

Например:

```
DionisNX# mkdir saved
DionisNX# copy running—config saved/1.config
DionisNX# ls saved
DionisNX# cat saved/1.config
DionisNX# rm saved
```

4. Выключение и перезагрузка

Чтобы выключить Программно-аппаратный комплекс (ПАК), нужно выполнить команду привилегированного режима:

```
# poweroff
```

Нажатие кнопки выключения на корпусе эквивалентно данной команде.

Так как данная команда довольно опасна, администратор должен будет подтвердить свои намерения в ответ на заданный вопрос. Чтобы принудительно выключить ПАК (без необходимости ответа на вопрос), используйте:

```
# poweroff force
```

Для перезагрузки ПАК нужно выполнить команду:

```
# reboot
```

При выключении/перезагрузке вся несохранённая текущая конфигурация (running-config) будет потеряна. При повторной загрузке системы будет применена сохранённая конфигурация (startup-config). Во время команды reeboot, если имеется несохранённая конфигурация, администратору потребуется подтвердить свои намерения в интерактивном режиме. Чтобы принудительно перезагрузить ПАК используйте команду:

```
# reboot force
```

5. Предварительная настройка

5.1 Начало работы (Установка и смена паролей)

В самом начале работы с системой Dionis-NX необходимо сменить пароли администратора и оператора. Порядок смены паролей и настройки учетных записей описан в п. 6.

5.2 Серийный номер изделия

Для собственного удобства вы можете записать серийный номер изделия на жёсткий диск с помощью команды:

```
DionisNX# serial 123—456—789
```

И в дальнейшем посмотреть его вместе с идентификатором платформы с помощью команды:

```
DionisNX# show platform
```

Для очистки серийного номера воспользуйтесь командой:

```
DionisNX# clear serial
```

5.3 Общие настройки

Предварительная настройка узла Dionis-NX включает в себя:

- установку имени узла;
- установку времени и часового пояса;
- нумерацию сетевых интерфейсов.

Имя узла задаётся с помощью команды hostname из режима конфигурации:

```
DionisNX(config)# hostname router—1
router—1(config)#
```

Часовой пояс задаётся с помощью команды timezone из режима конфигурации. Поддерживается два формата задания часового пояса: в виде буквенной аббревиатуры и часового смещения, а также в виде имени часовой зоны.

В первом случае при задании часового пояса необходимо ввести буквенную аббревиатуру часового пояса (MSK, OMST, VLAT и т.д.) и часовое смещение (со знаком + или -), которое необходимо прибавить к локальному времени, чтобы получить время UTC. Например, OMST-7, PST+8. По умолчанию в Dionis-NX задан часовой пояс MSK-3. Например:

```
(config)# timezone MSK—3
```

При использовании имени часовой зоны необходимо ввести имя зоны в виде основного и уточняющего региона:

```
(config)# timezone Europe/Moscow
```

Чтобы задать время и дату, из привилегированного режима нужно выполнить команду clock. Например:

```
# clock 13:58 27 02 2018
```

Время и дата задаются в формате «часы:минуты[:секунды] число_ номер месяца_год».

Чтобы скорректировать время без изменения даты, следует выполнить команду:

```
# clock 13:58
```

Посмотреть текущую дату, время и часовой пояс можно с помощью команды непривилегированного режима:

```
> show clock
```

Нумерация сетевых интерфейсов (сопоставление имён «ethernet <n>» с MAC-адресами) обычно делается заводом-изготовителем. Однако, если по каким-то причинам потребуется заново перенумеровать интерфейсы, то это можно сделать с помощью интерактивной команды привилегированного режима:

```
# interface enumerate ethernet
```

После подачи команды будет предложено сопоставить MAC-адреса с номерами. Чтобы введенные изменения вступили в силу, необходимо будет произвести перезагрузку:

```
# reboot
```

Порядок нумерации интерфейсов может не совпадать с их физическим расположением на передней панели маршрутизатора. Поэтому, администратор может посчитать удобным перенумеровать интерфейсы в естественном порядке. В случае, если необходимо перенумеровать некоторые интерфейсы, следует воспользоваться командами:

```
show interface bindings
interface blink ethernet <n>
interface bind ethernet <n> <mac>
```

Первая из этих команд покажет список всех интерфейсов с указанием MAC-адресов. Чтобы посмотреть, где физически расположен определенный интерфейс (с номером <n>) на панели маршрутизатора, следует применить вторую из команд. Если у интерфейса есть на панели цветовой индикатор, то он будет “мигать”. После этого, с помощью третьей из команд можно задать новый номер этого интерфейса. Последовательность второй и третьей команд следует повторить столько раз, сколько интерфейсов следует перенумеровать.

Для того, чтобы отменить сопоставление интерфейсов с номерами, воспользуйтесь командой:

```
clear interface bindings
```

5.4 Включение IPv6

Если маршрутизатор должен работать в IPv6 сетях, необходимо активировать реализацию протокола IPv6. Для этого, в configure режиме выполните команду: ip6 enable

```
(config)# ip6 enable
```

После этого, необходимо выполнить перезагрузку командой reboot (предварительно записав конфигурацию).

Для того, что бы включить режим передачи IPv6 пакетов с интерфейса на интерфейс, выполните команду: ip6 forwarding из режима configure.

```
(config)# ip6 forwarding
```

Для выключения работы с IPv6 выполните команды: no ip6 forwarding и ip6 disable. Затем выполните перезагрузку системы.

```
(config)# no ip6 forwarding
(config)# ip6 disable
(config)# do write
(config)# do reboot
```

5.5 Интерфейсы

В данном разделе рассматривается базовая настройка сетевых интерфейсов Ethernet. Однако многие команды настройки интерфейса типа ethernet существуют и для других типов интерфейсов, например pptp, l2tp и др., и работают похожим образом. Это такие команды, как: disable, enable, ip address и многие другие.

Для настройки конкретного интерфейса необходимо ввести команду в режиме конфигурации:

```
(config)# interface ethernet номер_интерфейса
```

Данная команда осуществляет вход в режим конфигурации интерфейса.

Следующие команды выполняют минимально необходимую настройку интерфейса (активация и назначение IPv4-адреса/маски подсети):

```
(config—if—ethernet0)# ip address 192.168.1.1/24
(config—if—ethernet0)# enable
```

Если интерфейс будет настраиваться с использованием службы DHCP (п. 30), то необходима команда:

```
(config—if—ethernet0)# ip address dhcp
```

В этом случае будет невозможно использовать часть команд настройки системы разрешения имен:

- ip resolver nameserver

- ip resolver domain

Чтобы использовать любые команды системы разрешения имен и одновременно иметь возможность получить IP-адрес по DHCP, следует подать команду:

```
(config—if—ethernet0)# ip address dhcp iponly
```

Данная команда только присваивает интерфейсу IP-адрес, не трогая остальные сетевые настройки, такие как адрес сервера имен (DNS-сервер) и др.

Если необходимо, интерфейсу может быть назначено несколько IP-адресов из разных сетей. Пример:

```
(config—if—ethernet0)# ip address 10.1.1.1/24
(config—if—ethernet0)# ip address 10.2.2.2/24
```

Если необходимо задать несколько адресов для одной сети, используйте команду ip secondary-address (только для IPv4):

```
(config—if—ethernet0)# ip address 10.1.1.1/24
(config—if—ethernet0)# ip secondary—address 10.1.1.2/24
(config—if—ethernet0)# ip address 10.2.2.2/24
(config—if—ethernet0)# ip secondary—address 10.2.2.3/24
```

Команды с префиксом «no» удаляют соответствующие настройки или возвращают значения по умолчанию.

Для задания IPv6 адресов, используйте команды ip6 address и no ip6 address.

При необходимости можно настроить другие параметры интерфейса с помощью команд:

- multicast - режим групповой передачи;
- speed - скорость интерфейса;
- mac - изменить MAC-адрес по умолчанию;
- mtu - изменить MTU;
- arp - запрет/разрешение ARP-обмена.

Для просмотра текущей конфигурации интерфейса можно ввести команду (из текущего режима):

```
(config—if—ethernet0)# do show
```

или из привилегированного режима:

```
show interface ethernet номер
```

Для вывода текущего состояния интерфейса и статистики по интерфейсу можно использовать команды привилегированного режима:

```
show interface ethernet номер
show interface ethernet номер link
show interface ethernet номер stat
show interface ethernet номер device
show interface ethernet номер device stat
```

Вторая из команд выводит информацию по текущему состоянию интерфейса, а третья - по статистике.

Если необходимо деактивировать интерфейс (без потери настроек), то нужно выполнить команду конфигурации интерфейса:

```
(config—if—ethernet0)# disable
```

Следующая команда режима конфигурации деактивирует интерфейс и удаляет все настройки:

```
(config)# no interface ethernet номер
```

Существует возможность дублирования всего входящего в интерфейс трафика на другой ethernet-интерфейс. Данная возможность может быть использована для последующего анализа или мониторинга трафика сторонним ПО, которое получает доступ к входящему трафику, слушая его на выделенном интерфейсе. Для установки дублирования трафика следует использовать команду mirror, например:

```
(config—if—ethernet0)# mirror ethernet 1
```

Теперь, весь входящий трафик будет дублироваться (включая ethernet-заголовки) на интерфейс ethernet 1.

Для отключения режима дублирования трафика используется команда no mirror:

```
(config—if—ethernet0)# no mirror
```

5.6 Статическая маршрутизация

Для того, чтобы узел Dionis-NX мог выполнять функции IPv4 маршрутизатора, необходимо разрешить передачу транзитных пакетов от одного сетевого интерфейса к другому с помощью команды режима конфигурации:

```
(config)# ip forwarding
```

Данная команда уже присутствует в файле конфигурации по умолчанию (default-config). Процедура включения функции маршрутизации IPv6 описана в главе “Включение IPv6” (п. 8.7)

Если по каким-то причинам нужно запретить транзит пакетов между интерфейсами, то нужно выполнить команду:

```
(config)# no ip forwarding
```

В системе Dionis-NX могут существовать следующие типы IP-маршрутов:

- connected - маршруты, появляющиеся автоматически при назначении IP-адресов сетевым интерфейсам;
- static - принудительно назначенные статические маршруты;
- kernel - маршруты, загруженные в ядро системы, минуя систему конфигурации Dionis-NX;
- bgp, rip, ospf - маршруты, создаваемые соответствующими службами динамической маршрутизации.

Для добавления статических IPv4 маршрутов используется команда режима конфигурации «ip route …». Для удаления статического маршрута используется аналогичная команда «no ip route …».

Формат команды:

```
[no] ip route <ip_prefix>|default <gw_ip>|<iface>|blackhole|reject|null0 [<distance>]
```

где:

- ip_prefix - A.B.C.D/M - шаблон IP-адресов назначения. Пакеты, IP-адреса назначения которых удовлетворяют данному шаблону, будут направляться по данному маршруту;
- default - маршрут по умолчанию (эквивалентно записи 0.0.0.0/0).
- <gw_ip> - A.B.C.D - IP-адрес соседнего маршрутизатора. Пакет будет направлен на данный маршрутизатор;
- <iface> - тип и номер интерфейса (например, ethernet 0). Пакет будет выпущен через данный интерфейс;
- blackhole, null0 - пакет будет удалён;
- reject - пакет будет удалён. Отправителю будет отправлено сообщение ICMP «Unreachable»;
- <distance> - административное расстояние (administrative distance). Данная величина имеет значение при выборе наиболее оптимального маршрута. (Имеет смысл вместе с динамической маршрутизацией).

Если задано несколько маршрутов, пересекающихся по адресам назначения, то более приоритетным будет более точный маршрут (с большей маской), а менее приоритетным - более общий маршрут (с меньшей маской). Маршрут по умолчанию (0.0.0.0/0) имеет наименьший приоритет.

Примеры настройки статических маршрутов:

Маршрут по умолчанию:

```
(config)# ip route default 192.168.1.1
```

Данная команда предписывает маршрутизатору направлять все проходящие/исходящие пакеты, не адресованные данному узлу и не попадающие под другие правила маршрутизации, на маршрутизатор 192.168.1.1.

Удаление статического маршрута:

```
(config)# no ip route default 192.168.1.1
```

Маршрут через интерфейс:

```
(config)# ip route 10.0.1.0/24 ethernet 1
```

Данная команда указывает маршрутизатору, что сеть 10.0.1.0/24 подключена непосредственно к интерфейсу ethernet 1, и что все пакеты, адресованные в данную сеть, будут направлены в данный интерфейс. (Если неизвестен MAC-адрес для IP-адреса назначения, то будет выполнен ARP-запрос через указанный интерфейс).

Тупиковый маршрут:

```
(config)# ip route 10.2.0.0/16 blackhole
```

Все пакеты, адресованные в сеть 10.2.0.0/16, будут отброшены.

Чтобы посмотреть все добавленные статические маршруты, нужно выполнить команду привилегированного режима:

```
# show ip route
```

Для вывода информации о всех маршрутах нужно выполнить команду:

```
# show ip route
```

Также имеется возможность выводить часть таблицы маршрутизации. Например:

```
# show ip route summary
# show ip route connected
# show ip route static
# show ip route 10.0.1.0/24
```

Данные команды выводят соответственно: количество маршрутов разных типов, только маршруты типа «connected», только статические маршруты, маршруты с префиксом назначения 10.0.1.0/24.

Для работы с IPv6 статическими маршрутами используйте команды с префиксом ip6: ip6 route, no ip6 route, show ip6 route и так далее.

5.7 Клиент DNS

Система Dionis-NX может рассматриваться не только как сервер, обеспечивающий различные сервисы клиентам, но и как клиент других сервисов, выполняющихся как на самой системе, так и на других узлах.

В данном разделе рассматривается настройка DNS-клиента системы Dionis-NX. Эта настройка необходима, если будут использоваться команды системы (режима enable или режима configure, за исключением команд службы DNS), в качестве параметров которых вместо IP-адресов указываются доменные имена. Например, это могут быть такие команды:

```
DionisNX# ping factor—ts.ru
DionisNX# netperf np—server udp
```

Без правильно настроенной клиентской части DNS-системы, данные команды не смогут получить IP-адреса узлов, имена которых указаны в качестве их параметров.

5.7.1 Связь с DHCP и динамическими интерфейсами.

Выполнение некоторых команд настройки клиента DNS невозможно, если один из интерфейсов системы настроен на обслуживание по DHCP. Это следующие команды:

```
(config)# ip resolver domain
(config)# ip resolver nameserver
```

Если необходимо использовать эти команды и применить DHCP (п. 30) на интерфейсе, то следует выполнить на этом интерфейсе команду ip address dhcp iponly.

В результате конфигурация сети, предлагаемая сервером DHCP (например, сервера имен и доменное имя), не будет использоваться, за исключением IP-адреса, который будет присвоен интерфейсу таким же образом, как и в случае использования для него команды ip address dhcp.

Взаимное использование resolver nameserver, dhcp и ppp dns:

- чтобы использовать ip resolver nameserver/domainlist: не должно быть команд ip address dhcp
- чтобы использовать ip address dhcp: не должно быть команд ip resolver nameserver/domainlist
- чтобы использовать ppp getdns (в динамических PPP-интерфейсах): если есть ip resolver nameserver, то полученные от провайдера адреса не будут использоваться

Команды ppp getdns и ip address dhcp могут быть использованы одновременно, в данном случае будут использоваться адреса серверов имен полученные последними по времени.

Чтобы настроить клиент DNS войдите в режим configure.

5.7.2 Базовая настройка клиента

Основные параметры клиента DNS:

- сервер имен,используемый системой для разрешения DNS-запросов (т.е. чтобы узнать IP-адрес узла,заданного по имени);
- доменное имя по умолчанию.

Рассмотрим пример настройки:

```
(config)# ip resolver domainlist zeta.int
(config)# 1 ip resolver domainlist factor—ts.int
(config)#
(config)# ip resolver nameserver 10.0.0.1
(config)# ip resolver nameserver 10.0.0.2
(config)# 1 ip resolver nameserver fc00::1
(config)# ip resolver host  10.0.0.3 zeta.int zeta—alias.int
(config)# ip resolver host  fc00::2 ipv6—server.int
```

В результате будет создана следующая конфигурация:

- список доменных имен (в порядке приоритета): factor-ts.int zeta.int;
- список IP-адресов серверов имен (в порядке приоритета): fc00::1, 10.0.0.1 10.0.0.2;
- список статической привязки IP-адресов к именам: имена zeta.int zeta-alias.int имеют адрес 10.0.0.3, а ipv6-server.int имеет адрес fc00::2.

5.7.3 Дополнительная настройка клиента

В дополнительной настройке описаны различные опции клиента DNS, более подробно о которых можно узнать в подразделе **Команды настройки сервиса DNS.**

Кратко перечислим соответствующие команды:

```
(config)# ip resolver sortlist 10.0.1.0/24
(config)# ip resolver sortlist 10.0.2.0/24
(config)# ip resolver options attempts 3
(config)# ip resolver options ndots 2
(config)# ip resolver options timeout 3
(config)# ip resolver options edns0
(config)# ip resolver options rotate
```

С помощью этих команд задаются следующие параметры сервиса DNS:

- первые две команды задают список сортировки: если имя соответствует нескольким IP-адресам, они будут возвращены в порядке определённом списком сортировки;
- число попыток запроса на сервер имен;
- минимальное число точек в имени домена, чтобы оно считалось абсолютным именем домена;
- начальный интервал ожидания ответа на запрос (в секундах);
- включение расширения DNS, позволяющего принимать/посылать сообщения DNS, размером больше 512 байт, по UDP-протоколу;
- включение механизма распределения нагрузки, связанной с разрешением имен, между серверами имен,которые указаны командами ip resolver namserver.

5.8 Просмотр и настройка ARP-таблицы

Чтобы вывести текущую таблицу соответствия IPv4- и MAC-адресов соседних узлов, нужно выполнить непривилегированную команду:

```
> show ip arp
```

Следующая команда выводит это соответствие для конкретного IP-адреса:

```
> show ip arp 192.168.1.1
```

Для очистки всей текущей ARP-таблицы (кроме принудительных соответствий IP-MAC) используется команда привилегированного режима:

```
# clear ip arp
```

Также можно удалить соответствие IP-MAC для конкретных адресов. Например:

```
# clear ip arp 192.168.1.1
```

Если необходимо установить принудительное соответствие IP-MAC для некоторых узлов, следует выполнить в режиме конфигурации команду:

```
(config)# ip arp <ip_addr> <mac_addr>
```

Удалить принудительное соответствие IP-MAC можно с помощью команды:

```
(config)# no ip arp <ip_addr>
```

Принудительные соответствия IP-MAC также отображаются командой «show ip arp» вместе с временными соответствиями. Чтобы отобразить только принудительные соответствия, можно использовать команду привилегированного режима:

```
# show ip arp config
```

Во время своей работы Dionis-NX поддерживает таблицы с информацией о хостах находящихся в том же сегменте сети, что и маршрутизатор. По умолчанию, максимальное число записей в таблице равно 8192. В некоторых случаях этого может оказаться недостаточно, поэтому существует команда, позволяющая изменить настройки кеша. Синтаксис команды:

ip arp thresh <минимальная граница> <ватерлиния> <максимальная граница>

- минимальная граница – это то число записей в кеше, которое могут находиться постоянно;
- ватерлиния – при достижении размера кеша, равного этому параметру, будет запущен сборщик мусора;
- максимальная граница - максимальное количество записей в кеше.

Например:

```
# ip arp thresh 1024 8192 16384
```

По этой команде устанавливается максимальный размер кеша в 16384 записей. Нижняя граница – 1024. Ватерлиния – 8192.

Для сброса параметров в первоначальное состояние следует использовать команду no ip arp thresh. Для просмотра - команду show ip arp thresh (из режима enable).

По-умолчанию, Dionis-NX отвечает на ARP запросы, направленные на IP адреса своих интерфейсов с любого интерфейса. Если такое поведение нежелательно, то его можно запретить:

```
# ip arp—filter
```

В таком случае, интерфейс не будет отвечать на ARP запросы, направленные на IP адреса соседних интерфейсов.

5.9 Просмотр и настройка таблицы соседей IPv6 (neighbours)

Команды работы с таблицей соседей IPv6 очень похожи на команды работы с ARP кеш-таблицей.

Команда	Назначение
[no] ip6 neighbour <адрес> <mac адрес>	Задание/отмена ручной привязки IPv6 адреса соседа
ip6 neighbours thresh …	Задание параметров кеша
clear ip6 neighbours …	Очистка кеша
show ip6 neighbours …	Показать информацию о таблице

5.10 Настройки стека TCP/IP

Данные настройки могут повлиять на производительность и работу различных подсистем и служб системы. Поэтому их следует использовать с особой аккуратностью.

Настройки осуществляются из режима configure.

5.10.1 Настройки протокола IP

Маршрутизация транзитных IP-пакетов, т.е. пакетов, не предназначенных для данной системы, называется IP-форвардинг. Если опция IP-форвардинга не включена, то система не будет пересылать транзитные пакеты через свои интерфейсы и будет обрабатывает пакеты, адресованные только ей.

IP-форвардинг включается командой:

```
(config)# ip forwarding
```

Обычно на маршрутизаторах всегда следует включать IP-форвардинг.

Процедура включения функции маршрутизации IPv6 описана в главе “Включение IPv6” (п. 8.7)

Следующей командой можно установить Time-To-Live для IP-пакета (т.е. максимальное число узлов, через которое может пройти данный пакет, прежде чем будет отброшен):

```
(config)# ip ttl 50
```

Установка таймаута сессии для неизвестных или неподдерживаемых протоколов уровня layer-4 (все, что кроме TCP/UDP):

```
(config)# ip timeout 600
```

Следующая команда задает максимальное число обычных и транзитных соединений:

```
(config)# ip max—connections 10000
```

При оптимизации пропускной способности сетевой подсистемы могут оказаться полезными опции размеров буферов сокетов.

Следующая команда задает минимальное, заданное по умолчанию и максимальное значение буфера сокета для исходящих пакетов протоколов TCP и UDP (в байтах)

```
(config)# ip wmem 10000 40000 100000
```

Следующая команда задает минимальное, заданное по умолчанию и максимальное значение буфера сокета для входящих пакетов протоколов TCP и UDP (в байтах)

```
(config)# ip rmem 10000 50000 90000
```

Значения параметров этих двух команд требуют пояснения. При создании сокета ему выделяется буфер для отправки и приема. Эти команды задают размеры в байтах этих буферов (на примере ip rmem):

- минимальный (10000) : размер буфер не может быть снижен системой или пользователем ниже этого значения, т.е. это гарантированный размер буфера;
- заданный по умолчанию (50000) : размер буфера, выделяемый системой по умолчанию при создании сокета;
- максимальный (90000) : это максимальный размер буфера, который может быть выделен сокету.

5.10.2 Настройки протокола TCP

5.10.2.1 Базовая настройка

Период посылки сообщения keep-alive при соединении по протоколу TCP задается следующими командами:

```
(config)# ip tcp keepalive interval 50
(config)# ip tcp keepalive probes 5
(config)# ip tcp keepalive time 1000
```

В этом примере для проверки того, не сорвано ли соединение, каждые 1000 секунд будут посылаться до пяти Keep-Alive сообщений с интервалом в 50 секунд. Если даже на пятое сообщение ответа не пришло, соединение будет разрываться.

Чтобы включить режим SACK (режим выборочных подтверждений, Selective Acknowlegment), следует выполнить команду:

```
(config)# ip tcp selective—ack
```

Использование выборочных подтверждений означает, что только те данные, которые не были получены, требуют повторной передачи, что повышает эффективность использования пропускной способности сети.

Чтобы включить режим syncookies, следует выполнить команду:

```
(config)# ip tcp syncookies
```

Использование этого режима позволяет защититься от DoS-атак типа SYN-спуфинг (посылке большого числа SYN-пакетов на систему.

Чтобы включить расширения TCP (RFC 1323) для сетей с большой пропускной способностью, следует выполнить команды:

```
(config)# ip tcp timestamps
(config)# ip tcp window—scaling
```

Чтобы включить ECN механизм (расширение TCP, RFC 3168), следует выполнить команду:

```
(config)# ip tcp ecn server—mode
```

Чтобы включить режим ABC механизма (расширение TCP, RFC 3465), следует выполнить комнду:

```
(config)# ip tcp abc aggresive
```

Установка таймаута установленной TCP-сессии:

```
(config)# ip tcp timeout established 432000
```

5.10.2.2 Настройка памяти

При тонкой настройке сетевой подсистемы бывает важно установить, как протокол TCP будет регулировать потребление памяти для своих нужд.

Регулирование осуществляется установкой минимального, среднего и максимального объема потребляемой памяти. Рассмотрим пример:

```
(config)# ip tcp mem 1000 50000 90000
```

Значения параметров задаются в 4Kb-страницах. Опишем каждый из трех параметров команды:

- минимальный размер (1000) : если объем используемой TCP-протоколом памятиниже 1000, протокол никак не будет снижать свое потребление;
- средний размер (50000): если объем используемой TCP-протоколом памяти выше 50000, протокол будет снижать свое потребление, пока не достигнет 1000.
- максимальный размер (90000): максимальный объем памяти, доступный для всех TCP-сокетов системы.

Аналогичная команда существует и для UDP протокола: она называется ip udp mem.

В настройках протокола TCP существует команда ip tcp rmem и ip tcp wmem, которые аналогичны командам ip rmem и ip wmem, рассмотренным выше. Однако в данном случае они задают размеры для буферов сокетов протокола TCP.

5.10.3 Настройки протокола UDP

При тонкой настройке сетевой подсистемы бывает важно установить, как протокол UDP будет регулировать потребление памяти для своих нужд.

Регулирование осуществляется установкой минимального, среднего и максимального объема потребляемой памяти. Рассмотрим пример:

```
(config)# ip udp mem 1000 50000 90000
```

Значения параметров аналогичны команде ip tcp mem для протокола TCP.

Значения минимального, умалчиваемого и максимально размеров для буферов сокетов UDP неявно устанавливаются равными размерами, заданным командой ip rmem и ip wmem.

Установка таймаута UDP-сессии:

```
(config)# ip udp timeout 30
```

Установка таймаута UDP-сессии для UDP-потока:

```
(config)# ip udp timeout stream 180
```

5.10.4 Настройки протокола ICMP

Чтобы включить обработку ICMP-запросов типа ECHO, следует выполнить команду:

```
(config)# ip icmp echo
```

Чтобы включить обработку широковещательных ICMP-запросов типа ECHO, следует выполнить команду:

```
(config)# ip icmp broadcast—echo
```

5.10.5 Настройки протокола IPv6

Команда	Назначение
[no] ip6 autoconf disable|enable	Автоконфигурировать ли адреса
[no] ip6 tempaddr disable|enable|prefer	Настройка реализации RFC3041
[no] ip6 redirects accept|reject	Принимать ли редиректы
[no] ip6 mtu <mtu>	Задать mtu по умолчанию
[no] ip6 hop_limit <число>	Задать hop limit по умолчанию (64)

5.11 Основы работы со службами (сервисами)

Работа со всеми службами Dionis-NX имеет общие особенности.

5.11.1 Команды режима configure

Команда входа в конфигурацию службы:

```
(config)# service NAME
```

В данной команде и далее параметр NAME соответствует имени службы (например, dns или dhcp).

Команда запуска службы:

```
(config—NAME)# enable
```

Команда остановки службы:

```
(config—NAME)# disable
```

5.11.2 Команды режима enable

Просмотр информации о службе:

```
# show service NAME ...
```

Изменение каких-либо данных службы, за исключением конфигурации (команда может отсутствовать):

```
# service NAME ...
```

Команда быстрого перезапуска службы (команда может отсутствовать):

```
# service NAME reload
```

Осуществляет быстрый перезапуск службы.

Эта команда может быть полезна для быстрой повторной инициализации службы, если поменялись какие-либо данные/настройки службы, которые администратор изменил командами режима enable. Например, для службы DNS это может быть изменение в зонных файлах (произведенное из режима enable), для службы DHCP - изменение в базе данных выданных адресов (также произведенное из режима enable).

Эту команду можно также вызвать из режима configure в секции службы, она называется reload.

Команда обычного перезапуска службы (команда может отсутствовать):

```
# service NAME restart
```

Осуществляет перезапуск службы.

Эта команда аналогична вызову команд disable и enable в секции службы в режиме configure.

Эту команду можно также вызвать из режима configure в секции службы, она называется restart.

5.12 Диагностика

Для диагностики проблем настройки и функционирования TCP/IP-сетей администратор может пользоваться довольно большим набором описанных ниже средств, большинство из которых доступны как непривилегированному пользователю, так и привилегированному пользователю (режим enable).

5.12.1 Утилита ping

Позволяет формировать icmp-пробы. В качестве обязательного параметра задается IP-адрес или имя хоста.

Для IPv6 используйте утилиту ping6

5.12.2 Утилита traceroute

Позволяет отследить маршрут, по которому движется пакет пробы.

5.12.3 Монитор sysmon

Позволяет следить за состоянием системы в реальном времени (только для enable режима). При доступе к локальной консоли, монитор можно активировать комбинацией Alt-F10. Затем активировать консоль, нажав “Ввод”. Alt-F1 - осуществляет возврат на первую консоль. С помощью клавиши «пробел» – изменяется выводимая информация. Кроме этого, во время работы монитора, нажав клавишу h или ?, можно ознакомиться с описанием структуры выводимой информации и с краткой справкой по использованию sysmon.

5.12.4 Информация на LCD-мониторе

На LCD-мониторе показывается краткая информация о состоянии системы. С помощью клавиш на панели осуществляется навигация по информационным полям. Цвет индикатора в правой части панели индикатора описывает общее состояние системы:

- зеленый - нормальное функционирование;
- желтый - загрузка или выключение системы;
- красный - требуется внимание администратора.

5.12.5 Команды show

Существует множество команд show, которые могут использоваться администратором для выявления проблем. Ниже приводится список основных команд:

команда	краткое описание
show interface <тип интерфейса> <номер интерфейса>	информация о состоянии сетевого интерфейса
show interface <тип интерфейса> <номер интерфейса> link	информация о состоянии среды и низкоуровневых настройках интерфейса
show interface <тип интерфейса> <номер интерфейса> stat	статистика интерфейса
show interface	информация обо всех интерфейсах
show interface stat	статистика по всем интерфейсам
show ip sock	информация по сокетам
show ip connections	информация по открытым соединениям и кэшу NAT
show ip stat	информация по IP-статистике

Эти команды могут выполняться из любого режима, но в режиме configure требуется префикс do.

6. Учетные записи

При работе с системой Dionis-NX существует два вида учетных записей — учетная запись для получения консольного доступа к системе и учетные записи для администрирования. Учетная запись для получения консольного доступа в системе всегда одна. Учетных записей для администрирования (учетных записей администратора) может быть несколько. Среди учетных записей администраторов выделяется учетная запись “adm”, которая является учетной записью по умолчанию. Остальные учетные записи администратора создаются в ходе работы системы.

6.1 Учетная запись консольного доступа (учетная запись оператора)

Учетная запись консольного доступа (учетная запись оператора) - одна для всей системы и имеет имя “cli”. Эту учетную запись невозможно удалить, но допустимо менять для нее пароль и изменять другие настройки, присущие учетным записям.

По умолчанию пароль для учетной записи оператора - “cli”. При вводе системы в эксплуатацию необходимо сменить пароль для учетной записи оператора. Это может сделать администратор системы. Процедура смены пароля описана в п. 6.3.3 .

Учетная запись оператора позволяет просматривать некоторые параметры системы и часть информации о ее состоянии. Работа под учетной записью оператора не позволяет менять какие-либо настройки системы.

Администратор может войти в систему, используя учетную запись “cli”, а при необходимости выполнения административных действий сменить непривилегированную запись на учетную запись администратора с помощью команды “enable”, как показано на примере ниже:

```
DionisNX> enable ivanov
```

В данном примере “ivanov” - это имя учетной записи администратора. Если учетная запись администратора не указана явно, то будет использовано предопределенное имя “adm”. Подробнее об учетной записи “adm” будет сказано далее.

6.2 Учетные записи администратора

В системе Dionis-NX может существовать множество учетных записей администратора. Все действия администратора отражены в системном журнале и привязаны к имени учетной записи.

Администраторы могут иметь различные права на изменение настроек системы. Например, одному администратору доступна настройка сетевых интерфейсов, а другому нет. Подробнее права доступа администраторов описаны в разделе “Ролевая модель”, п. 7. Также администратор может иметь права супервизора. В этом случае ему доступны любые операции по настройке системы. По умолчанию, учетная запись “adm” имеет права супервизора. Как объявить администратора супервизором, описано в разделе по администрированию учетных записей. Необходимо ответственно относиться к назначению

администратора супервизором, так как в этом случае система становится ему полностью подконтрольна.

При начале работы с системой Dionis-NX существует единственная учетная запись администратора "adm". Администратор "adm" является супервизором. Так же как и учетную запись консольного доступа, учетную запись "adm" невозможно удалить, однако ее можно заблокировать, либо лишить прав супервизора. Это можно сделать после того, как будут настроены другие рабочие записи администраторов. По умолчанию пароль для учетной записи "adm" - "adm". При вводе системы в эксплуатацию необходимо сменить пароль этой учетной записи.

В случае утери пароля администратора и невозможности администрирования системы, существует способ сброса паролей учетных записей "cli" и "adm" в значения по умолчанию. Это можно сделать, загрузившись с инсталляционного флеш-диска. В случае сброса пароля одновременно сбрасываются и другие настройки этих учетных записей.

6.3 Управление учетными записями

Управление учетными записями выполняется в режиме enable.

6.3.1 Создание и удаление учетных записей

Создать или удалить можно только учетную запись администратора (кроме adm). Учетные записи консольного доступа создавать и удалять нельзя.

Создается учетная запись с помощью следующей команды:

```
DionisNX# account create ivanov
```

В этом примере будет создана учетная запись с именем "ivanov". Команда создания учетной записи может иметь параметры, как это показано на примере ниже:

```
DionisNX# account create ivanov realname ”Иван Иванов” desc ”Администратор” supervisor locking 3 10
        60
```

В данном примере использованы четыре необязательных параметра команды. Параметр "realname" задает реальное имя администратора. Параметр "desc" задает текстовое описание для учетной записи. Параметр "supervisor" наделяет вновь созданную учетную запись правами супервизора. Необходимо осторожно относиться к использованию параметра "supervisor". Параметр "locking" позволяет автоматически блокировать учетную запись если за определенное время было предпринято несколько неудачных попыток входа в систему.

Удалить существующую учетную запись можно с помощью команды "account remove":

```
DionisNX# account remove ivanov
```

Команда удаляет учетную запись "ivanov". При удалении учетной записи не удаляются пользовательские данные учетной записи. Чтобы полностью удалить и учетную запись и ее пользовательские данные, используется параметр "purge" этой команды:

```
DionisNX# account remove ivanov purge
```

6.3.2 Просмотр учетных записей

Список существующих учетных записей можно получить с помощью следующей команды:

```
DionisNX# show account *
adm
 cli
ivanov
```

Данный пример отображает существующие в системе три учетный записи: “adm”, “cli”, “ivanov”.

Для просмотра подробной информации о выбранной учетной записи используется команда:

```
DionisNX# show account ivanov
realname ”Иван Иванов”
description ”Администратор”
locking 3 10 60
supervisor
expire period 99999 last 2015 4 16 warning 7
delegate @default
```

Значение отображаемых полей будет описано ниже, в разделе, посвященном настройке учетных записей (п. 6.3.4) .

В случае, если необходимо получить подробную информацию по всем учетным записям сразу, используется опция “verbose”:

```
DionisNX# show account * verbose
account config adm
 realname ”Administrator”
 description ”Default administrator”
 supervisor
 expire period 99999 last 2015 4 16 warning 7
 delegate @default
account config cli
 realname ”Console”
 description ”Console access”
 expire period 99999 last 2011 11 14 warning 7
 delegate @default
account config ivanov
 realname ”Иван Иванов”
 description ”Администратор”
 locking 3 10 60
 supervisor
 expire period 99999 last 2015 4 16 warning 7
 delegate @default
```

6.3.3 Изменение пароля учетной записи

Пароль для учетной записи можно задать/изменить с помощью команды “passwd”.

```
DionisNX# passwd ivanov
```

Система в интерактивном режиме попросит ввести новый пароль для учетной записи. Пароль может содержать любые символы латинского алфавита, цифры, знаки препинания. Длина пароля должна быть не меньше 8 символов. Если введенный пароль короче, смены пароля не произойдёт.

Для генерации надежного пароля можно воспользоваться командой “passwd-gen”:

```
DionisNX# passwd—gen
!yBNa74y
```

Опционально можно указать длину пароля от 8 до 64 символов:

```
DionisNX# passwd—gen 20
urW^!wDEK%I11xV&dk&S
```

6.3.4 Настройки учетной записи

Настройки существующей учетной записи можно редактировать. Для входа в режим редактирования настроек учетной записи используется команда “account config”, например для редактирования настроек учетной записи администратора “ivanov”:

```
DionisNX# account config ivanov
```

В режиме редактирования настроек учетной записи существует набор специальных команд, позволяющих:

- Задать реальное имя владельца учетной записи;
- Задать описание учетной записи;
- Назначить или снять права супервизора;
- Заблокировать или снять блокировку с учетной записи;
- Задать срок действия пароля учетной записи;
- Задать дату последнего изменения пароля;
- Задать количество дней до истечения срока действия пароля, начиная с которого пользователь будет получать предупреждение о необходимости смены пароля;
- Изменять полномочия и роли, доступные учетной записи;
- Настроить параметры автоматической блокировки учетной записи.

6.3.5 Реальное имя

Для задания реального имени владельца учетной записи используется команда “realname”.

```
DionisNX(account—ivanov)# realname ”Иван Иванов”
```

Данная настройка не является обязательной. Установленное значение реального имени может быть сброшено следующей командой:

```
DionisNX(account—ivanov)# no realname
```

6.3.6 Описание

Для учетной записи может быть задано описание. Это произвольное текстовое поле.

```
DionisNX(account—ivanov)# description ”Администратор. Комната 256”
```

Данная настройка не является обязательной. Установленное значение описания может быть сброшено следующей командой:

```
DionisNX(account—ivanov)# no description
```

6.3.7 Супервизор

В обычном режиме учетной записи администратора могут быть доступны не все возможности системы. Какие именно возможности системы доступны администратору, определяется его полномочиями и ролями в рамках ролевой модели (п. 7). В случае, если администратор объявлен супервизором, ролевая модель игнорируется и ему доступны все без исключения полномочия системы. Объявить администратора супервизором можно следующей командой:

```
DionisNX(account—ivanov)# supervisor
```

Отменить права супервизора для учетной записи можно следующей командой:

```
DionisNX(account—ivanov)# no supervisor
```

Так как супервизору доступны любые возможности системы, то необходимо крайне ответственно подходить к назначению администраторам прав супервизора.

6.3.8 Автоматическая блокировка учетной записи

Для предостращения подбора пароля учетная запись может быть автоматически заблокирована на определенное время, если было предпринято несколько неудачных попыток входа в систему.

Параметр “locking” позволяет настроить параметры автоматической блокировки:

```
DionisNX(account—ivanov)# locking 3 10 60
```

В данном примере конфигурация такова, что если подряд было предпринято 3 неудачных попытки входа с интервалом не более 10 секунд между попытками, то пользователь будет заблокирован на 1 минуту.

6.3.9 Блокировка учетной записи

Учетная запись может быть временно заблокирована. Если учетная запись заблокирована, то ее владелец не сможет войти в систему. В отличие от удаления учетной записи, блокировка позволяет сохранить все настройки учетной записи, включая пароль, и только временно запретить вход в систему.

Для блокирования учетной записи используется следующая команда:

```
DionisNX(account—ivanov)# disable
```

Для разблокирования учетной записи используется следующая команда:

```
DionisNX(account—ivanov)# enable
```

6.3.10 Срок действия пароля

Для учетной записи может быть ограничен срок действия пароля. Это делается в целях безопасности. Срок действия пароля определяется количеством дней с момента последней смены пароля.

```
DionisNX(account—ivanov)# expire period 90
```

В приведенном примере устанавливается, что пароль действителен 90 дней. Когда срок действия пароля истечет, владелец учетной записи обязан сменить пароль. Если данный администратор не имеет полномочий на редактирование учетных записей, то смену пароля выполняет назначенный администратор, имеющий такие полномочия. Без смены пароля вход в систему будет блокирован.

В некоторых случаях бывает необходимо вручную задать дату последней (предыдущей) смены пароля.

```
DionisNX(account—ivanov)# expire last 2015 04 16
```

В примере “2015 04 16” - это год, месяц и день, соответственно. Существует синтаксис команды для задания текущей даты, как даты последней смены пароля:

```
DionisNX(account—ivanov)# expire last now
```

Система предоставляет возможность указать, за сколько дней до истечения срока действия пароля владелец учетной записи будет получать предупреждение о необходимости его смены. Следующая команда устанавливает количество дней до истечения срока действия пароля, определяющее момент времени, начиная с которого владелец учетной записи будет получать предупреждение:

```
DionisNX(account—ivanov)# expire warning 5
```

Администратор ivanov будет получать предупреждение о необходимости смены пароля, начиная с 5 дней до истечения срока действия пароля.

6.3.11 Хешированный пароль

Для облегчения переноса информации о пользователях с одной машины на другую, существуют команды, которые позволяют получить пароль указанного пользователя в хешированном виде, а затем, на другой машине установить этот пароль пользователю. Так как пароль доступен только в хешированном виде, то сам пароль остается скрыт от администратора.

Администратор может использовать следующие команды для получения хешированных паролей для отдельного пользователя и для всех пользователей сразу, соответственно:

```
DionisNX# show account adm passwd—hash
 realname ”Administrator”
 description ”Default administrator”
 supervisor
 passwd—hash
 ”$6$RKIt60GMf59a.w.7$Q9eD13u6qvqIKUzKfdhfllV3xPFbpyk8OmxmELittOpOnyGmjnLPpjs6vtvtoe6ywFrX5qwN/BoHBA/tXMoM3.”
 expire period 99999 last 2015 4 16 warning 7
 delegate @default
DionisNX# show account * verbose passwd—hash
account config adm
 realname ”Administrator”
 description ”Default administrator”
 supervisor
 passwd—hash
 ”$6$RKIt60GMf59a.w.7$Q9eD13u6qvqIKUzKfdhfllV3xPFbpyk8OmxmELittOpOnyGmjnLPpjs6vtvtoe6ywFrX5qwN/BoHBA/tXMoM3.”
 expire period 99999 last 2015 4 16 warning 7
 delegate @default
account config cli
 realname ”Console”
 description ”Console access”
 passwd—hash
 ”$6$RKIt60GMf59a.w.7$Q9eD13u6qvqIKUzKfdhfllV3xPFbpyk8OmxmELittOpOnyGmjnLPpjs6vtvtoe6ywFrX5qwN/BoHBA/tXMoM3.”
 expire period 99999 last 2011 11 14 warning 7
 delegate @default
account config ivanov
 realname ”Иван Иванов”
 description ”Администратор”
 locking 3 10 60
 supervisor
 passwd—hash
 ”$6$RKIt60GMf59a.w.7$Q9eD13u6qvqIKUzKfdhfllV3xPFbpyk8OmxmELittOpOnyGmjnLPpjs6vtvtoe6ywFrX5qwN/BoHBA/tXMoM3.”
 expire period 99999 last 2015 4 16 warning 7
 delegate @default
```

Если администратору уже известен хешированный пароль пользователя, то выполнив на новой машине следующую команду в режиме редактирования пользователя, пароль будет установлен для этого пользователя:

DionisNX(account—ivanov)# passwd—hash
”6RKIt60GMf59a.w.7$Q9eD13u6qvqIKUzKfdhfllV3xPFbpyk8OmxmELittOpOnyGmjnLPpjs6vtvtoe6ywFrX5qwN/BoHBA/tXMoM3.”

7. Ролевая модель

7.1 Права доступа учетной записи администратора

Если администратор не имеет статуса супервизора, то его права доступа при настройке различных параметров системы определяются назначенным ему списком полномочий и ролей. Если администратор имеет статус супервизора, то имеет доступ к любым возможностям системы вне ролевой модели.

Каждое из полномочий может определять доступ к:

- командам одной подсистемы, которые используются в каком-то конкретном режиме (например, к командам для одного из интерфейсов, используемых в режиме enable, или к командам в режиме конфигурирования и т.д.);
- к одной конкретной команде;
- к определенным операндам одной из команд.

В случае если какие-то полномочия определяются на группу команд, и есть другие полномочия на конкретную команду из этой группы, то для доступа к этой конкретной команде необходимо иметь все эти полномочия. Аналогично, если полномочия определены на отдельную команду, и есть другие полномочия на использование некоторых параметров этой команды, то для использования указанных параметров этой команды необходимо иметь все эти полномочия.

Отдельные полномочия могут быть связаны друг с другом. Подробнее о зависимости полномочий — в разделе 7.5.

Каждая роль представляет собой совокупность полномочий. Учетная запись администратора, имеющая какую-либо роль, получает доступ, определяемый всеми полномочиях этой роли. Все роли системы создаются в ходе ее настройки и функционирования. По умолчанию никаких ролей в системе нет.

Полные полномочия учетной записи — это все полномочия, назначенные этой учетной записи, плюс все полномочия всех ролей, которые имеет эта учетная запись. Подробнее о полномочиях и ролях можно узнать в разделах 7.2, 7.4, 7.5. По умолчанию учетная запись обладает полномочиями “@default”. В основном, по умолчанию учетная запись получает права только на просмотр информации о системе.

7.2 Полномочия системы

Полномочия в системе являются предопределенными и не могут быть созданы пользователем. Однако набор полномочий может расширяться при появлении новых версий системы. Поэтому администратор должен учитывать в своей работе возможность появления новых полномочий при установке новой версии системы. Все предопределенные полномочия для удобства восприятия сгруппированы по подсистемам, к которым они относятся. При использовании полномочий полезно учитывать следующие мнемонические правила, используемые в именах полномочий: - Все имена полномочий начинаются с символа @; - Имена полномочий состоят из нескольких слов, разделенных символом “.”. Первое из слов описывает подсистему, к которой относится данное из полномочий. Последующие слова, как правило, указывают на типы полномочий для указанной подсистемы.

Основные типы полномочий:

- conf — право на использование команд режима конфигурирования;
- oper — право на использование команд режима enable;
- show — право на использование команд просмотра информации;
- key — право на использование команд управления ключами;
- crypto — право на использование команд шифрования;
- server — право на работу с сервером в протоколах, предусматривающих наличие клиента и сервера;
- client — право на работу с клиентом в протоколах, предусматривающих наличие клиента и сервера.

Приведенные типы полномочий могут представлять собой иерархическую систему — например, право на работу с сервером может быть определено для конфигурирования сервера или только для просмотра информации сервера. В таких случаях имена полномочий могут представлять собой цепочку, состоящую более чем из двух слов. Например, с помощью имени @l2tp.server.conf задаются полномочия, которые позволяют конфигурировать сервер для протокола l2tp.

7.2.1 Крипто - средства

Имена полномочий	Описание
@key.conf	Базовые крипто-средства, управление ключами
@key.clear	Удаление ключей и объектов PKI
@key.show	Получение информации о ключах
@ike.conf	Конфигурация службы и туннелей IKE (п. 59.7)
@ike.oper	Управление соединениями и состояниями службы IKE (без изменения конфигурации)
@ike.show	Получение информация о IKE
@disec.conf	Открытые туннели DiSEC (+ сжатие)
@disec.show	Открытые туннели DiSEC (без изменения конфигурации)
@disec.key	DiSEC-ключи
@disec.crypto	Шифрованные туннели DiSEC
@ca.conf	Конфигурация сервера DiCert
@ca.oper	Обслуживание сервера DiCert (без изменения конфигурации и БД)
@ca.show	Просмотр состояния и БД сервера DiCert.
@dikey.conf	Настройка ключевой подсистемы DiKey.
@dikey.oper	Файловые операции DiKey (подпись/проверка подписи).
@dikey.show	Просмотр информации о подсистеме DiKey.
@tlsproxy.conf	Настройка службы TLSProxy.
@tlsproxy.show	Диагностика службы TLSProxy.

7.2.2 Сетевые настройки

Имена полномочий	Описание
@net.conf	TCP/IP-настройки, resolver, arp, conntrack, clear interface stat
@net.oper	TCP/IP-настройки, resolver, arp, conntrack, clear interface stat (без изменения конфигурации)
@net.show	TCP/IP-настройки, resolver, arp, conntrack, clear interface stat (только получение информации)
@net.tools	команды диагностики ping, traceroute (п. 5.12), whois, arping, nslookup (п. 29.9)

7.2.3 Контроллеры устройств

Имена полномочий	Описание
@serial.conf	Конфигурирование последовательного порта (RS232)
@serial.show	Получение информации о последовательном порте
@e1.conf	Конфигурирование контроллеров E1
@e1.show	Получение информации о контроллерах E1

7.2.4 Сетевые интерфейсы

Имена полномочий	Описание
@ethernet.conf	Конфигурирование Ethernet
@ethernet.bind	Привязка номера интерфейса Ethernet к контроллеру
@ethernet.show	Получение информации об Ethernet
@bond.conf	Конфигурирование bonding
@bridge.conf	Конфигурирование моста
@bridge.show	Получение информации о мосте
@dummy.conf	Конфигурирование псевдоинтерфейса
@gre.conf	Конфигурирование gre
@gretap.conf	Конфигурирование gretap
@hdlc.conf	Конфигурирование hdlc (для E1)
@l2tp.server.conf	Конфигурирование L2TP-сервера
@l2tp.server.show	Получение информации о L2TP-сервере
@l2tp.client.conf	Конфигурирование L2TP-клиента
@l2tp.client.show	Получение информации о L2TP-клиенте
@pptp.server.conf	Конфигурирование PPTP-сервера
@pptp.server.show	Получение информации о PPTP-сервере
@pptp.client.conf	Конфигурирование PPTP клиента

Имена полномочий	Описание
@pptp.client.show	Получение информации о PPTP-клиенте
@ovpn.server.conf	Конфигурирование OpenVPN-сервера
@ovpn.server.show	Получение информации об OpenVPN-сервере
@ovpn.client.conf	Конфигурирование OpenVPN-клиента
@ovpn.client.show	Получение информации об OpenVPN-клиенте
@ovpn.key.conf	Конфигурирование ключей OpenVPN
@ovpn.key.show	Получение информации о ключах OpenVPN
@pep.conf	Конфигурирование pep
@pep.show	Получение информации о pep

7.2.5 Управление трафиком

Имена полномочий	Описание
@acl.conf	Конфигурирование списков доступа
@acl.oper	Действия (кроме настройки) со списками доступа. Обеспечивает доступ к команде “clear ip recent-list” (очистить списки недавних пакетов, п. 8.4.6)
@acl.show	Получение информации об ACL
@nat.conf	Конфигурирование NAT
@nat.show	Получение информации о NAT
@qos.conf	Конфигурирование QoS
@qos.show	Получение информации о QoS

7.2.6 Сетевые сервисы

Имена полномочий	Описание
@dhcp.server	Настройка серверов DHCP и DHCP-relay
@dhcp.oper	Действия (кроме настройки) с серверами DHCP и DHCP-relay
@dhcp.show	Получение информации о сервере DHCP и DHCP-relay
@dns.server	Настройка сервера доменных имен DNS
@dns.oper	Действия (кроме настройки) с сервером доменных имен DNS
@dns.show	Получение информации о DNS-сервере
@netperf.server	Настройка сервера тестирования netperf и iperf
@netperf.client	Настройка клиента тестирования netperf и iperf
@netperf.show	Получение информации тестирования netperf и iperf
@lldp.server	Настройка сервера LLDP
@lldp.show	Получение информации о сервере LLDP
@ntp.server	Настройка сервера времени NTP
@ntp.show	Получение информации о сервере NTP
@proxy.server	Настройка прокси-сервера
@proxy.oper	Действия с кешем прокси-сервера

Имена полномочий	Описание
@proxy.show	Получение информации о прокси-сервере
@slagent.server	Настройка сервера slagent
@slagent.show	Получение информации о slagent
@snmp.sever	Настройка сервера SNMP
@snmp.show	Получение информации о сервере SNMP
@ssh.server	Настройка сервера SSH
@ssh.client	Настройка клиента SSH
@ssh.auth	Возможность работать авторизованными ключами
@telnet.server	Настройка сервера telnet
@telnet.client	Настройка клиента telnet
@vrrp.server	Настройка сервера VRRP
@vrrp.show	Получение информации о VRRP
@diweb.server	Настройка через HTTP-протокол
@netflow.conf	Конфигурирование NetFlow
@netflow.oper	Использование NetFlow
@radius.conf	Конфигурирование авторизации по radius протоколу
@radius.show	Получение информации о настройках подключения к radius серверу
@tacacs.conf	Конфигурирование авторизации по tacacs+ протоколу
@tacacs.show	Получение информации о настройках подключения к tacacs+ серверу

7.2.7 Системы обнаружения вторжений

Имена полномочий	Описание
@ids.sever	Настройка системы обнаружения и предотвращения вторжений IDS
@ids.oper	Действия (кроме настройки) системы обнаружения и предотвращения вторжений IDS
@ids.show	Получение информации о системе обнаружения и предотвращения вторжений IDS
@diamant.server	Настройка системы Diamant. Эта система зависит от ids
@idssur.server	Настройка системы на основе пакета Suricata
@idssur.oper	Действия (кроме настройки) системы idssur
@idssur.show	Получение информации о системе idssur
@idsm.sever	Настройка системы обнаружения и предотвращения вторжений IDSM
@idsm.oper	Действия (кроме настройки) системы обнаружения и предотвращения вторжений IDSM
@idsm.show	Получение информации о системе обнаружения и предотвращения вторжений IDSM

7.2.8 Маршрутизация

7.2.8.1 Статическая маршрутизация

Имена полномочий	Описание
@route.conf	Конфигурирование статической маршрутизации
@route.show	Получение информации о статической маршрутизации
@policy-route.conf	Конфигурирование policy route
@policy-route.show	Получение информации о policy route
@mpls.conf	Конфигурирование MPLS
@mpls.show	Получение информации о настройках MPLS

7.2.8.2 Динамическая маршрутизация

Имена полномочий	Описание
@droute-common.conf	Общие команды конфигурирования для различных видов динамической маршрутизации
@droute-common.show	Получение информации об общих структурах данных для динамической маршрутизации
@ospf.conf	Конфигурирование OSPF
@ospf.oper	Действия с OSPF
@ospf.show	Получение информации OSPF
@ospf6.conf	Конфигурирование OSPF6 (OSPF для IPv6 сети)
@ospf6.oper	Действия с OSPF6 (OSPF для IPv6 сети)
@ospf6.show	Получение информации OSPF6 (OSPF для IPv6 сети)
@bgp.conf	Конфигурирование BGP
@bgp.oper	Действия с BGP
@bgp.show	Получение информации BGP
@rip.conf	Конфигурирование RIP
@rip.oper	Действия с RIP
@rip.show	Получение информации RIP
@rip6.conf	Конфигурирование RIPNG (RIP для IPv6 сети)
@rip6.oper	Действия с RIPNG (RIP для IPv6 сети)
@rip6.show	Получение информации RIPNG (RIP для IPv6 сети)
@ldp.conf	Конфигурирование LDP (MPLS)
@ldp.oper	Действия с LDP
@ldp.show	Получение информации LDP (MPLS)
@bfd.conf	Конфигурирование BFD
@bfd.oper	Действия с BFD
@bfd.show	Получение информации BFD

7.2.8.3 Маршрутизация с использованием VRF

Имена полномочий	Описание
@vrf.conf	Команды конфигурирования для различных видов маршрутизации с использованием VRF
@vrf.show	Получение информации о VRF

7.2.8.4 Мультикаст(multicast)-маршрутизация

Имена полномочий	Описание
@mroute.conf	Конфигурирование статической мультикаст-маршрутизации
@mroute.show	Получение информации о мультикаст-маршрутизации
@dvmrp.conf	Конфигурирование мультикаст-маршрутизации DVMRP
@dvmrp.oper	Действия с DVMRP
@igmp.conf	Конфигурирование мультикаст-маршрутизации IGMP
@igmp.oper	Действия с IGMP
@pim.conf	Конфигурирование мультикаст-маршрутизации PIM
@pim.oper	Действия с PIM

7.2.9 Журналирование и трассировка

Имена полномочий	Описание
@log.conf	Конфигурирование системных журналов
@log.oper	Действия (кроме конфигурирования) с системными журналами
@log.show	Получение информации о настройке системных журналов
@log.watcher	Работа со службой watcher
@log.watcher.priv	Работа со службой watcher (привилегированные команды)
@trace.conf	Конфигурирование трассировки
@trace.oper	Действия (кроме конфигурирования) с трассировкой
@trace.show	Получение информации о настройке трассировки
@tcpdump.oper	Действия с анализатором трафика tcpdump
@mailer.conf	Конфигурирование почтового клиента
@mailer.oper	Действия (кроме конфигурирования) с почтовым клиентом
@mailer.show	Получение информации о настройке почтового клиента

7.2.10 Системные операции

Имена полномочий	Описание
@dip.oper	Операции с DIP-пакетами
@dip.show	Получение информации о DIP-пакетах

Имена полномочий	Описание
@data.oper	Действия со слотами данных, командами над пакетами ПО ПАК Dionis-NX и слотами данных (restore, os bind, schedule rebind, schedule restore и т.д.) (п. 65.2)
@data.show	Получение информации о слотах данных
@backup.oper	Команды создания резервной копии данных (os data backup), безопасной резервной копии (schedule backup) (п. 65.2)
@backup.show	Получение информации о резервных копиях
@boot.oper	Команды загрузки системы boot default, boot fallback, boot experemental, и команда миграции на другой пакет ПО ПАК Dionis-NX schedule migrate (п. 65.2)
@boot.show	Получение информации о параметрах загрузки
@cluster.conf	Конфигурирование кластера (п. 63)
@cluster.oper	Действия с кластером (кроме конфигурирования)
@cluster.show	Получение информации о кластере
@host.conf	Общие настройки (времени, часового пояса, имени узла) (п. 5.3), принудительной проверки файловой системы на ошибки командой schedule fsck (п. 66.3)
@host.show	Получение информации об общих настройках системы
@schedule.show	Получение информации о действиях, которые должны быть выполнены после перезагрузки
@hw.show	Получение информации об оборудовании
@account.conf	Настройка учетных записей
@account.passwd	Управление паролями учетных записей
@account.show	Получение информации об учетных записях
@account.passwd-hash	Получение информации о хешированном пароле
@role.conf	Настройка ролей
@role.show	Получение информации о ролях
@poweroff.oper	Право выключения системы
@reboot.oper	Право перегружать систему
@conf.write	Право перезаписывать startup-config
@conf.show	Получение информации о startup-config
@watchdog.oper	Право использовать механизм сторожевого таймера для обеспечения перезагрузки удаленной системы (со старым конфигурационным файлом) в случае потери связи из-за ошибочного изменения настроек на этой удаленной системе
@file.oper	Право выполнения файловых операций
@file.net	Право выполнения сетевых операций с файлами
@removable.oper	Право выполнения операций с внешними устройствами (дисками)

Имена полномочий	Описание
@birq.conf	Право на конфигурирование балансировки прерываний

7.2.10.1 Мандатные метки

Имена полномочий	Описание
@mcbc.conf	Использование мандатных меток МСВС в фильтрах (п. 8.4.8)

7.2.10.2 DiPool

Имена полномочий	Описание
@dipool.server	Пользовательский DiPool
@dipool.client.conf	Конфигурирование подключения к удаленному DiPool-сервера
@dipool.client.oper	Право получения образов с удаленного DiPool-сервера
@dipool.client.show	Право получения информации с удаленного DiPool-сервера

7.3 Команды управления полномочиями и ролями системы для учетных записей

Расширить права доступа учетной записи добавлением ей полномочий или ролей можно следующими командами:

```
DionisNX(account—ivanov)# delegate @ospf.conf
DionisNX(account—ivanov)# delegate myrole
```

Первая команда в примере добавляет учетной записи ivanov полномочия "@ospf.conf", вторая добавляет роль с именем "myrole". Если учетная запись имела полномочия какой-либо роли, и в ходе дальнейшей работы полномочия этой роли были изменены, то и учетная запись изменит свои полномочия. Однако это изменение произойдет только после завершения текущей сессии и открытия новой сессии для этой учетной записи.

Отменить определенные права доступа учетной записи можно командами:

```
DionisNX(account—ivanov)# no delegate @ospf.conf
DionisNX(account—ivanov)# no delegate myrole
```

Первая команда удаляет полномочия "@ospf.conf" из списка назначенных полномочий администратора "ivanov". Вторая команда удаляет роль "myrole" из списка назначенных данному администратору ролей администратора.

Существует команда для добавления администратору всех полномочий сразу.

```
DionisNX(account—ivanov)# delegate *
```

Добавление всех существующих полномочий может понадобиться в случае, когда необходимо создать администратора, обладающего большинством полномочий, за исключением небольшого списка

отдельных полномочий. Тогда администратору добавляются все существующие полномочия, а затем из списка доступных полномочий исключаются те полномочия, которые не будут доступны данному администратору. При добавлении всех полномочий следует учитывать, что общий набор полномочий может быть изменен в случае обновления системы до новых версий (эти новые версии могут включать в себя новые, не существующие в данной версии, полномочия). При необходимости добавления учетной записи вновь появившихся полномочий следует либо добавить их в явном виде, либо снова выполнить команду "delegate *".

7.4 Управление ролями

Для перехода к конфигурированию роли следует выполнить команду (в режиме enable):

```
DionisNX# role <имя роли>
```

Если параметр соответствует существующей в системе роли, то произойдет переход к конфигурированию этой роли. Если же параметр не соответствует существующей в системе роли, то в системе будет создана новая роль с указанным именем, которая не будет иметь никаких полномочий. После того, как будет произведен вход в режим конфигурирования роли, могут быть даны команды добавления/удаления полномочий для этой роли, например:

```
DionisNX(myrole)# delegate @ospf.conf
DionisNX(myrole)# no delegate @ospf.conf
```

Первая из этих команд добавляет полномочия "@ospf.conf" роли myrole. Вторая из этих команд исключает полномочия "@ospf.conf" из роли myrole. В качестве параметров в команде delegate при конфигурировании роли могут быть указаны не полномочия, а уже существующие роли. Например, при выполнении команды:

```
DionisNX(myrole)# delegate testrole
```

роль myrole получит все полномочия роли testrole, причем связь между ролями сохранится. Это означает, что если для роли testrole полномочия будут расширены, то и роль myrole получит новые полномочия роли testrole.

Кроме управления полномочиями роли, в системе существует несколько команд управления ролью. В режиме конфигурирования роли можно добавить любое количество полномочий.

Команда:

```
DionisNX# no role <имя роли>
```

удаляет роль с именем . Если в качестве параметра задано значение "*", то будут удалены все роли системы.

Команда:

```
DionisNX# show role <имя роли>
```

показывает указанную роль , если она есть. Если в качестве параметра задано значение "*", то будут показаны все роли системы.

Команда:

```
DionisNX# clone role <имя роли—источника> <имя роли—получателя>
```

копирует роль-источник в роль-получатель, т.е. создает копию роли-источника с указанием имени новой роли.

7.5 Отображение полномочий и зависимости полномочий

В системе часто имеются отдельные полномочия на конфигурирование какой-либо подсистемы, на использование (в режиме enable) этой подсистемы и получение информации о подсистеме. Целесообразно, чтобы учетная запись, имеющая полномочия на конфигурирование подсистемы, по умолчанию имела права и на использование и получение информации о подсистеме. Таким образом, в системе появляется зависимости между полномочиями — например, зависимости между полномочиями conf, oper и show для подсистемы. Набор таких зависимостей определен в системе по умолчанию. В системе существуют команды для отмены таких зависимостей или для создания новых зависимостей, а также команды возврата всех зависимостей в состояние, заданное в системе по умолчанию.

Как правило, зависимости полномочий, заданные по умолчанию, в наибольшей степени обеспечивают удобство работы администратора. Изменять зависимости для полномочий администратору следует с осторожностью и только в случае, если он точно понимает, с какой целью он меняет зависимости, заданные по умолчанию.

Команда:

```
DionisNX# show capability <имя полномочия>
```

Показывает указанные полномочия и те полномочия, которые включаются в данные полномочия (зависят от них). Например, по умолчанию полномочия @log.conf включают полномочия @log.oper и @log.show. После выполнения команды:

```
DionisNX# show capability @log.conf
```

Будут отображены эти полномочия, а вслед за ним, отдельной строкой, зависимые полномочия (т.е. @log.oper и @log.show). Если в качестве параметра команды “show capability” задано значение “*“, то будут показаны все полномочия системы и их зависимости. Команда”show capability”может использоваться с параметром “delegate”. В этом случае будут показаны только те полномочия, у которых есть зависимости. Например, команда:

```
DionisNX# show capability * delegate
```

отобразит список всех полномочий системы, имеющих зависимости, и сами эти зависимости.

Команда:

```
DionisNX# capability <имя полномочия>
```

позволяет перейти в режим конфигурирования зависимостей полномочий. Сами полномочия, зависимые от конфигурируемых полномочий, добавляются командами “delegate”. Зависимости удаляются командами “no delegate”. Например, последовательность команд:

```
DionisNX# capability @log.conf
DionisNX(log.conf)# no delegate @log.oper
```

удалит зависимость полномочий @log.oper от полномочий @log.conf.

Команда

```
DionisNX# clear capability <имя полномочий>
```

Возвращает состояние всех зависимостей для полномочий в состояние, принятое в системе по умолчанию. Если в качестве параметра задано значение "*", то зависимости всех полномочий возвращаются в состояние, принятое в системе по умолчанию.

Команда

```
DionisNX# no capability <имя полномочий>
```

удаляет все существующие зависимости для полномочий . Если в качестве параметра задано значение "*", то удаляются зависимости всех полномочий.

8. Фильтрация и модификация

Подсистема фильтрации является базовым средством обеспечения безопасности сети и позволяет управлять прохождением трафика через интерфейсы маршрутизатора, разрешая или запрещая передачу пакетов, удовлетворяющих указанным правилам отбора. Правила отбора объединяются в IP-списки контроля доступа (ip access-list). Списки контроля доступа могут быть применены к конкретным интерфейсам (с учетом направления трафика), а также к маршрутизатору в целом (с учетом логики маршрутизации).

Ниже приведена упрощенная схема движения пакета через фильтры Dionis-NX:

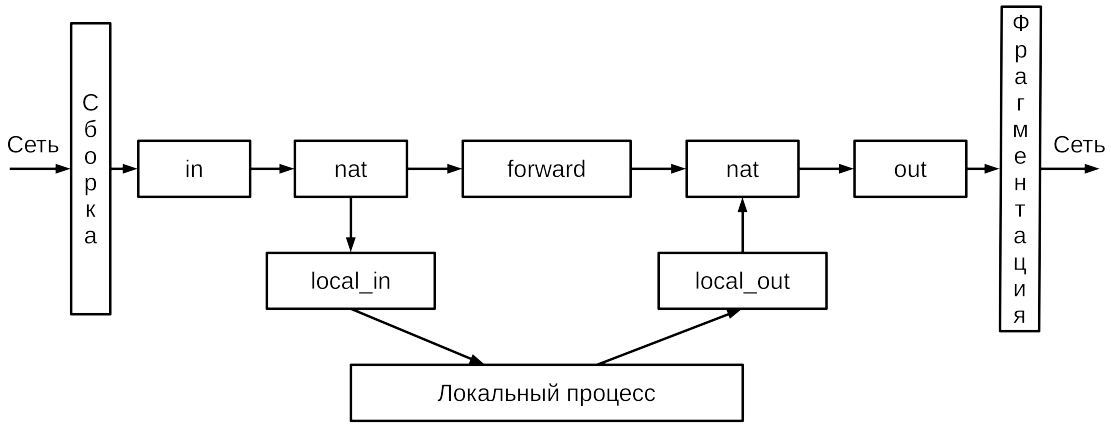

Рис. 8.1: Фильтрация

Название	Назначение
сборка	Сборка (дефрагментация) пакетов. Пакеты могут отбрасываться на этом этапе с помощью встроенной фильтрации (не требует настройки)
in	Входной фильтр интерфейса
nat	преобразование SNAT или DNAT
local_in	Внутренний фильтр для пакетов, направленных в систему
forward	Фильтр маршрутизации транзитных пакетов
local_out	Внутренний фильтр для сгенерированных в данной системе пакетов
out	Выходной фильтр интерфейса
фрагментация	Фрагментация пакетов перед отправкой их в сеть

8.1 Дефрагментация

Как видно на схеме, при прохождении через маршрутизатор пакеты всегда подвергаются дефрагментации. На этом этапе, ещё до попадания в фильтр in, пакеты могут быть отброшены. Это возможно,

например, в случае целенаправленной атаки сильно фрагментированным (или некорректным) трафиком.

Пакеты отбрасываются в случае превышения ограничения на суммарную длину фрагментов, ожидающих дефрагментацию. При превышении мягкого ограничения (по умолчанию 3Мб) фрагменты начинают случайно отбрасываться, а при превышении жёсткого ограничения (по умолчанию 4Мб) фрагменты начинают отбрасываться безусловно. Ограничения можно поменять командами:

```
DionisNX(config)# ip frag high <жёсткое ограничение>
DionisNX(config)# ip frag low <мягкое ограничение>
```

Кроме того, пакеты отбрасываются в случае, если фрагменты пакета ожидают дефрагментации слишком долго. Время ожидания следующего фрагмента по умолчанию равно 30 секунд и меняется командой:

```
DionisNX(config)# ip frag time <значение в секундах>
```

Для того, чтобы увидеть счётчики дефрагментации, воспользуйтесь командой:

```
DionisNX# show ip stat
```

8.2 Создание ip access-list

Для создания списка контроля доступа, в режиме configure необходимо выполнить команду ip access-list <acl_name>, где <acl_name> – это имя создаваемого списка, например:

```
DionisNX(config)# ip access—list mylist
```

После выполнения команды, задаются (или модифицируются) правила фильтрации этого списка. Каждое правило содержит критерии отбора трафика и может быть разрешающим (permit) или запрещающим (deny). Все правила в списке выполняются последовательно, до первого совпадения критериям отбора. Если ни одно из правил не удовлетворяет критериям, считается, что выполняется разрешающее правило. Например:

```
DionisNX(config—acl—mylist)# permit icmp
DionisNX(config—acl—mylist)# permit tcp
DionisNX(config—acl—mylist)# permit udp
DionisNX(config—acl—mylist)# deny
```

В данном примере, разрешается прохождение пакетов трех протоколов (icmp/udp/tcp), а все остальные протоколы запрещаются.

Для того, чтобы просмотреть правила отбора текущего редактируемого списка, следует выполнить команду:

```
DionisNX(config—acl—mylist)# do show
```

При этом будут выведены все правила текущего списка. Каждая строка снабжена числовым префиксом, указывающим позицию правила в списке.

Для того, чтобы удалить правило с конкретным номером, введите команду: no <номер правила> Например:

```
DionisNX(config—acl—mylist)# no 1
```

Для того, чтобы вставить новое правило в конкретную позицию, введите команду: <номер правила> <правило> Например:

```
DionisNX(config—acl—mylist)# 1 permit src 192.168.0.0/24
DionisNX(config—acl—mylist)# do show
1 permit src 192.168.0.0/24
2 permit tcp
3 permit udp
4 deny
```

Удаление всего содержимого текущего списка может быть осуществлено командой: no all. Для удаления списка, используется команда: no ip access-list <имя списка>.

Для просмотра информации о списках, существует две команды, доступные из enable режима.

show ip access-list <acl_name|*> config	Информация о действующей конфигурации
show ip access-list <acl_name|*>	Низкоуровневая информация из ядра ОС

Если в командах имя списка задано как *, будет показана информация о всех списках.

Например (из режима configure):

```
DionisNX(config)# do show ip access—list mylist config
```

8.3 Привязка ip access-list

Создание списка контроля доступа не означает, что список начинает действовать. Для того чтобы начала действовать фильтрация в соответствии с правилами списка, список должен быть привязан к интерфейсу и/или определенной цепочке в логике маршрутизации. Один и тот же список может быть привязан к нескольким интерфейсам/цепочкам.

8.3.1 Привязка к интерфейсу

Для того, чтобы привязать список к интерфейсу, нужно войти в режим конфигурации интерфейса и выполнить команду (или команды) ip access-group <имя списка> <направление>

Под параметром <направление> понимается направление трафика относительно интерфейса. Входящий трафик обозначается как in, а выходящий как out. Например:

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip access—group mylist in
DionisNX(config—if—ethernet0)# ip access—group mylist out
```

Для удаления связи с интерфейсом, необходимо выполнить команду: no ip access-group <имя> <направление>, например:

```
DionisNX(config—if—ethernet0)# no ip access—group mylist out
DionisNX(config—if—ethernet0)# do show
ip access—group mylist in
```

Иногда возникает необходимость фильтровать защищенный трафик (туннели DISEC/IPSEC). Фильтрация такого рода трафика означает, что правила access-list должны применяться на пакеты, которые уже были расшифрованы после приема их на интерфейсе, или еще не были зашифрованы, при их отправке через интерфейс. В этом случае для привязки фильтров применяется команда: ip access-group-xfrm, синтаксис которой аналогичен ip access-group. Для удаления связи с интерфейсом, следует использовать команду no ip access-group-xfrm.

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip access—group—xfrm mylist in
DionisNX(config—if—ethernet0)# ip access—group—xfrm mylist out
DionisNX(config—if—ethernet0)# no ip access—group—xfrm mylist out
DionisNX(config—if—ethernet0)# do show
ip access—group—xfrm mylist in
```

8.3.2 Привязка к цепочке обработки

Существует возможность привязать фильтр к внутренним цепочкам маршрутизатора. Для этого в режиме configure достаточно выполнить команды: ip access-group <имя> <цепочка>.

Где цепочка может принимать значения: local-in, local-out, forward, что соответствует цепочкам прохождения пакета в Dionis-NX.

Например:

```
DionisNX(config)# ip access—list mylist forward
```

8.4 Правила отбора

Выше были приведены примеры с очень простыми правилами отбора, фактически, единственным критерием задавался протокол датаграммы, или критерия не было вообще (deny). Списки контроля доступа могут содержать правила с комбинацией различных критериев отбора, которые объединяются в логическое «и», что делает фильтры простым и мощным инструментом по обеспечению безопасности в сети. Если какой-то из критериев не задан (например, не задан протокол датаграммы), то данному правилу отбора будут удовлетворять пакеты с любым значением критерия(любые протоколы).

Полный список критериев находится в Полном списке команд Dionis-NX , далее будут описаны основные критерии.

8.4.1 Адреса источника и назначения

Для задания адресов источника используется параметр src после которого указывается сетевой адрес. Например:

```
permit tcp src 192.168.16.0/24
```

Адрес, содержащий маску, определяет сеть. Адрес с маской 32 или без задания маски определяет адрес хоста. Если параметр src не задан, то правило отбора распространяется на датаграммы с любым адресом источника.

Аналогично параметру источника, существует параметр адреса приемника dst, с таким же синтаксисом:

```
permit tcp src 192.168.16.0/24 dst 10.0.0.1
deny
```

8.4.2 Порты источника и назначения

Если в критериях правила отбора указан протокол tcp или udp, то появляется возможность задать критерий отбора по портам источника и/или назначения. Для задания портов источника используется параметр: sport <port1> [port2]. Может быть задан как единственный порт <port1>, так и диапазон портов <port1> <port2>. например:

```
deny tcp sport 1 1000
```

Для задания портов приемника используется параметр dport, с аналогичным синтаксисом.

```
deny tcp dport 22
```

8.4.3 Направление передачи информации

Для задания направления используется параметр dir с аргументом <original|reply>. Например:

```
permit dir original
```

Согласно этому критерию отбираются пакеты, направление передачи информации в которых будет определено как “исходящее”.

8.4.4 Содержимое пакета

Существуют различные возможности задания критерия по содержимому пакета. Для этого используется либо параметр content, либо regex или pcre с аргументом <правило>, которое записывается на специальном языке и позволяет отслеживать содержимое байт/слов в пакете. Регулярные выражения для параметров regex и pcre должны соответствовать синтаксису PCRE2.

8.4.4.1 Параметр content

Для параметра content критерий работает следующим образом. Все позиции байтов считаются с 0. В простейшем случае, правило выбирает 4 байта по заданному смещению (start), применяет маску (mask) и сравнивает результат с попаданием в диапазон (range). В таком случае правило принимает вид: start&mask=range. Где range задается как диапазон (a:b) или значение (a).

Обычно нужно взять значение start меньшим на 3, чем позиция последнего байта, который нам нужен. Так, если нужны байты 4 и 5 из заголовка IP (поле ID), то start должен быть 5-3 = 2. mask убирает те байты, которые не нужны в критерии. Максимальная маска (которая означает анализ всех 4-х байт) может быть равной 0xffffffff. Так, чтобы взять байты 4 и 5, отбросим байты 2 и 3, что соответствует маске 0x0000ffff. Таким образом правило будет выглядеть так:

```
permit content 2&0xFFFF=0x2:0x0100
```

Пример критерия, который отбирает пакеты длиной, большей 256. Общая длина пакета лежит в байтах 2 и 3 IP датаграммы. Тогда стартовая позиция 3 - 3 = 0. Отбрасывая 2 байта получаем:

```
deny content 0&0xFFFF=0x100:0xFFFF
```

Пример критерия, основанного на выборке одного байта. Выберем байт TTL (смещение 8). 8 - 3 = 5, маска 0xff:

```
deny content 5&0xFF=0:3
```

Пример критерия, основанного на выборке четырех байт. Выберем IP-адрес назначения (байты 16-19). Маска не требуется:

```
permit content ”16=0xE0000001” remark ”224.0.0.1/32”
```

Если необходимо выбрать первые три байта (например, проверить сетевой адрес сети класса C), то необходима маска 0xffffff00.

```
deny content ”12&0xFFFFFF00=0xC0A80F00” remark ”src 192.168.15.0/24”
```

Чтобы посмотреть на поле TOS (1 байт заголовка IP), невозможно начать с байта 1-3 = -2. Нужно начать с байта 0, отбросить лишние байты и сдвинуть нужный байт в последнюю позицию. Для сдвига используются команды « и ».

```
permit content ”0&0x00FF0000>>16=0x08” remark ”tos 8”
```

Пример анализа флага MF.

```
deny content ”3&0x20>>5=1”
```

Анализ TCP-заголовка. Допустим, необходимо анализировать байты 4-7 заголовка TCP (sequence number). Для простоты, будем считать, что заголовок IP занимает 20 байт. Тогда критерий будет выглядеть, например, так:

```
permit content 24=0x29
```

Однако, в этом примере не анализируется протокол. Добавим проверку протокола (логическое «и»):

```
permit content ”6&0xFF=0x6 && 24=0x29”
```

Но размер IP-заголовка не всегда равен 20 байтам, поэтому необходимо еще доработать правило отбора. Специальный оператор @ позволяет использовать значение последнего выражение как стартовую позицию нового.

Для того, чтобы взять размер заголовка, необходим байт 0>>24, но нужны только 4 бита, умноженные на 4, то есть: 0>>22&0x3C. Теперь необходимо, чтобы это выражение стало новым смещением. Для этого и нужен оператор @:

```
permit content ”6&0xFF=0x6 && 0>>22&0x3C@4=0x29”
```

Еще один, более сложный, пример. Проверка на TCP, проверка на первый фрагмент или нефрагментированный пакет, проверка, что 4-7 байты TCP заголовка равны 41:

```
permit content ”6&0xFF=0x6 && 4&0x1FFF=0 && 0>>22&0x3C@4=0x29”
```

Пример проверки соответствия пакета сообщению ICMP Host Unreachables (ICMP, type 3, code 1).

```
permit content ”6&0xFF=1 && 4&0x1FFF=0 && 0>>22&0x3C@0>>16=0x0301”
```

8.4.4.2 Параметр regex

Параметр regex предписывает отбирать пакеты по содержимому в соответствии с регулярным выражением и определяет синтаксис аргумента <правило> как /<regex>/[N|G|f|p|i|m|s|x|1|2|3], где <regex> - posix регулярное выражение, а опции имеют следующее значение:

N	REG_NOSUB - не требовать поддержки адресации внутристрочных совпадений
G	REG_UNGREEDY - опция сообщает механизму регулярных выражений, чтобы они соответствовали как можно меньшему количеству данных
f	REG_UTF - шаблон и целевая строка обрабатываются как UTF-8 строки
p	REG_UCP - шаблон и целевая строка обрабатываются как Unicode строки
i	REG_ICASE - не различать регистр символов
m	REG_NEWLINE - символ новой строки будет обрабатываться как и любой другой символ строки
s	REG_DOTALL - метасимвол “точка” в шаблоне соответствует всем символам, включая перевод строк
x	REG_EXTENDED - использовать синтаксис расширенных регулярных рыражений POSIX во время интерпретации
1	REG_NOTBOL - не учитывается оператор начала новой строки
2	REG_NOTEOL - не учитывается оператор конца строки
3	REG_NOTEMPTY - пустая строка не является допустимым соответствием

8.4.4.3 Параметр pcre

Параметр pcre предписывает отбирать пакеты по содержимому в соответствии с регулярным выражением в стиле Perl и определяет синтаксис аргумента <правило> как /<pcre>/[A|E|G|i|m|s|x], где <pcre> - регулярное выражение в стиле Perl, а опции имеют следующее значение:

A	PCRE2_ANCHORED - соответствие только на первой позиции
G	PCRE2_$_ENDONLY - $ не соответствует символу новой строки в конце
f	PCRE2_UNGREEDY - опция сообщает механизму регулярных выражений, чтобы они соответствовали как можно меньшему количеству данных
i	PCRE2_CASELESS - не различать регистр символов
m	PCRE2_MULTILINE - ^ и $ соответствуют символам новой строки
s	PCRE2_DOTALL - метасимвол "точка" в шаблоне соответствует всем символам, включая перевод строк
x	PCRE2_EXTENDED - игнорировать пробел и # комментарии

8.4.5 Pcap

В качестве критериев отбора можно использовать pcap-выражения. Язык pcap-выражений устроен следующим образом:

Примитив pcap-выражения, как правило, состоит из одного или нескольких *квалификаторов*, за которыми следует *идентификатор* (имя или число). Квалификаторы бывают следующих типов:

- Тип – определяет тип идентификатора. Может принимать следующие значения: "host" (хост), "net" (сеть), "port" (порт), "portrange" (диапазон портов). По умолчанию предполагается квалификатор host.
- Направление – определяет направление, к которому относится идентификатор. Может принимать следующие значения: "src", "dst", "src or dst", "src and dst". По умолчанию предполагается направление "src or dst".
- Протокол – определяет протокол, к которому относится идентификатор. Может принимать значения "ip", "ip6", "tcp", "udp". По умолчанию выбираются все подходящие протоколы (так, если написать выражение "port 53", под него будут подходить как UDP-, так и TCP-пакеты, у которых либо порт источника, либо порт назначения равен 53).

Существуют некоторые примитивы, которые не удовлетворяют указанному шаблону. Смотрите описание примитивов ниже.

Более сложные выражения составляются путём объединения примитивов при помощи операторов и арифметических выражений.

8.4.5.1 Примитивы pcap

В таблице ниже заглавными буквами указаны параметры, вместо которых необходимо подставить конкретное значение. В качестве примера для [src/dst] host HOST расписаны три варианта применения с разными квалификаторами направления, далее в таблице эти три варианта указаны в одной строчке с опциональными квалификаторами в квадратных скобках.

Примитив	Сопоставление	Пример
dst host HOST	По ip-адресу назначения	dst host 10.0.0.1
src host HOST	По ip-адресу источника	src host 10.0.0.2
host HOST	Либо по ip-адресу источника, либо по ip-адресу назначения (эквивалентно (src or dst) host HOST	host 10.0.0.2
[src/dst] net NET/LEN	По сети источника либо назначения, в зависимости от квалификаторов	src net 10.0.0.0/24
[src/dst] net NET mask NETMASK	По сети с использованием маски сети	src net 10.0.0.0 mask 255.255.255.255
[tcp/udp] [src/dst] port PORT	По порту	src port 42
[tcp/udp] [src/dst] portrange PORT1-PORT2	По диапазону портов	dst portrange 1-1000
less LENGTH	По длине пакета. Эквивалентно len <= LENGTH	less 400
greater LENGTH	По длине пакета. Эквивалентно len >= LENGTH	greater 100
[ip/ip6] proto PROTO	По протоколу, вложенному в ip(6)-пакет. PROTO – либо число, либо название протокола. Названия протоколов tcp, udp, icmp должны быть экранированы обратным слэшом (), так как являются ключевыми словами. Сопоставление происходит только по первому протоколу, если используется цепочка заголовков.	ip proto esp
tcp, udp, icmp	Сокращения для proto \tcp, proto \udp, proto \icmp	udp
[ip/ip6] protochain PROTO	Аналогично proto, однако с прохождением всей цепочки заголовков. Работает гораздо медленнее.	ip6 protochain \tcp
ip broadcast	Broadcast-пакеты	ip broadcast
ip/ip6 multicast	Multicast-пакеты	ip multicast

8.4.5.2 Арифметические выражения

Язык pcap позволяет использовать арифметические выражения с использованием следующих операторов:

Оператор	Описание
>, <, >=, <=, =, !=	Сравнение
+, -, *, /, %	Арифметические операции (/ – деление нацело, % – взятие остатка)
&	Побитовое И
|	Побитовое ИЛИ
^	Побитовое исключающее ИЛИ
<<, >>	Битовые сдвиги

Арифметические выражения могут быть сгруппированы при помощи скобок.

В качестве аргументов к операторам используются либо числа (в обычной или шестнадцатеричной записи: 100, 0x35), либо данные пакета. Для доступа к данным пакета используется конструкция PROTO[EXPR:SIZE], где:

- PROTO – один из ip, ip6, udp, tcp, icmp. Определяет стартовую позицию, от которой считается отступ.
- EXPR – выражение, определяющее отступ от **начала заголовка** протокола, указанного в PROTO
- SIZE – размер фрагмента в байтах, может быть 1 (по умолчанию), 2 или 4

Некоторые выражения для отступов уже добавлены в качестве ключевых слов: icmptype (поле типа ICMP), icmpcode (поле кода ICMP) и tcpflags (поле флагов TCP)

Также некоторые значения полей добавлены в качестве ключевых слов:

- Для icmp[icmptype]: icmp-echoreply, icmp-unreach, icmp-sourcequench, icmp-redirect, icmp-echo, icmp-routeradvert, icmp-routersolicit, icmp-timxceed, icmp-paramprob, icmp-tstamp, icmp-tstampreply, icmp-ireq, icmp-ireqreply, icmp-maskreq, icmp-maskreply
- Для tcp[tcpflags]: tcp-fin, tcp-syn, tcp-rst, tcp-push, tcp-ack, tcp-urg

Для проверки длины пакета доступно специальное выражение length

Примеры:

- icmp[icmptype] = icmp—echo – этому выражению соответствуют echo-запросы ICMP.
- ip[0]&0xf!=5 – этому выражению соответствуют IPv4-пакеты с опциями (т.е. с длиной заголовка, не равной 5*4=20 байт).

8.4.5.3 Объединение примитивов

Примитивы могут быть объединены при помощи следующих операторов:

Оператор	Описание
!, not	Отрицание
&&, and	И
||, or	ИЛИ

Также можно использовать скобки для задания приоритетов операторов.

Пример:

```
”(ip src 10.0.0.1 and tcp port 20) or (ip dst 10.0.0.2 and udp port 33)”
```

8.4.5.4 Макросы Pcap

Доступно только в Dionis-NX DPS

Для упрощения работы с выражениями Pcap в Dionis-NX доступен механизм макросов. Макрос pcap – это фрагмент выражения pcap, который подставится в pcap-выражение при использовании pcap-macro.

Пример:

```
ip pcap—macro—list modbus
  macro mb—len tcp[((tcp[12:1]&0xf0)>>2)+4:2]
  ...
```

При использовании pcap-macro с параметром modbus будет использоваться библиотека макросов modbus, и mb-len будет заменяться на tcp[((tcp[12:1]&0xf0)>>2)+4:2].

8.4.5.5 Использование Pcap в Dionis-NX

Для использования pcap в Dionis-NX есть два критерия отбора: pcap и pcap-macro (**только в Dionis-NX DPS**). Критерий pcap-macro позволяет использовать библиотеку макросов для подстановки в pcap-выражение. Примеры:

```
permit pcap ”ip src 10.0.0.1”
permit pcap—macro modbus mb—len>0
```

8.4.6 Списки недавних пакетов

В Dionis-NX существует возможность создавать правила относительно накопленной информации о предшествующих пакетах, для этого используется критерий recent.

Для работы с критерием recent, вводится понятие списка недавних пакетов (ip recent-list). Каждый такой список идентифицируется по имени и заполняется динамически с помощью критерия recent <имя> set, например:

```
DionisNX(config—acl—ping)# permit icmp recent pings set
```

В данном случае, пакеты с протоколом icmp будут допущены к маршрутизации, при этом информация о пакетах (исходный адрес датаграммы и время) будет заноситься в список pings. Каждый список может сохранять до 1024 записей (адресов), в каждой записи может храниться информация о 20 последних пакетов.

Критерий recent может быть использован для проверки наличия записи в списке недавних пакетов. Для этого используется критерий: recent <имя> update|check.

Так, в следующем примере:

```
DionisNX(config—acl—ping)# permit tcp dport ssh recent pings update
DionisNX(config—acl—ping)# drop tcp dport ssh
```

Пакет, направляемый на порт службы ssh будет пропущен только в том случае, если ранее от хоста источника были получены icmp-пакеты.

Полный синтаксис критерия recent:

recent <имя списка> <операция> [аргументы]

Операции:

Название	Назначение
set	Добавить информацию о пакете в список
remove	Удалить информацию о пакете в список
check	Проверить информацию о пакете в списке
update	Проверить информацию о пакете в списке и обновить временную метку

Возможные аргументы для операции set:

Название	Назначение
dest	Запоминать адрес назначения, а не адрес источника

Возможные аргументы для операции remove:

Название	Назначение
dest	Работать относительно адреса назначения, а не адреса источника

Возможные аргументы для операций check и update:

Название	Назначение
dest	Проверять адрес назначения, а не адрес источника
seconds <число>	Проверять запись не старее заданного числа секунд
hitcount <число>	Проверять запись, число пакетов в которой больше или равно заданного числа
rttl	Проверять корректность ttl (соответствие ttl пакета с информацией в записи из списка)

Для работы с списками ip recent-list используются следующие команды в режиме enable:

Команда	Параметры	Назначение
show ip recent list	<* или имя списка> [IP адрес]	Просмотреть информацию о записях в recent списках
clear ip recent list	<* или имя списка> [IP адрес]	Очистить информацию о записях в recent списках

8.4.7 Состояние соединения

Ядро Dionis-NX содержит средства, обеспечивающие отслеживание состояния соединений и классификацию пакетов с точки зрения принадлежности к соединениям, что позволяет осуществлять полноценную фильтрацию трафика.

При этом, ядром поддерживаются следующие функции:

1. Отслеживание состояний отдельных соединений с тем, чтобы классифицировать каждый пакет либо как относящийся к уже установленному соединению, либо как открывающий новое соединение. При этом понятие «состояние соединения» искусственно вводится для протоколов, в которых оно изначально отсутствует (UDP, ICMP). При работе же с протоколами, поддерживающими состояния (например, TCP), активно используется эта возможность.
2. Отслеживание связанных соединений, например, ICMP-ответов на TCP- и UDP-пакеты.

В правилах отбора следует использовать критерий state для того, чтобы использовать информацию о состоянии соединения. При этом, можно указывать 4 состояния.

Название	Смысл
invalid	Пакет связан с неизвестным потоком или соединением и, возможно, содержит ошибку в данных или в заголовке
established	Состояние указывает на то, что пакет принадлежит уже установленному соединению, через которое пакеты идут в обоих направлениях
new	Пакет открывает новое соединение или пакет принадлежит однонаправленному потоку.
related	Пакет принадлежит уже существующему соединению, но при этом он открывает новое соединение.

В качестве примера, рассмотрим правила фильтрации для внешнего сетевого интерфейса, разрешающие только исходящие соединения (соединения из внутренней сети во внешнюю сеть).

```
DionisNX(config)# ip access—list wan
DionisNX(config—acl—wan)# permit state established
DionisNX(config—acl—wan)# permit state related
```

```
DionisNX(config—acl—wan)# deny
```

8.4.8 Мандатные метки МСВС и Astra Linux

Существует возможность задавать в критериях отбора диапазон мандатных меток пакетов, которые используются в ОС МСВС 3.0 и Astra Linux. Для этого используется критерий maclabel, который задает диапазон уровней мандатной метки, а также может содержать диапазон категорий.

Для задания диапазона уровней используется конструкция: <минимальный уровень>:<максимальный уровень>, например:

```
DionisNX(config)# ip access—list mac
DionisNX(config—acl—mac)# permit maclabel level 0:2
DionisNX(config—acl—mac)# deny
```

Если в качестве диапазона задано одно значение, то метка проверяется на совпадение со значением.

Для отрицания диапазона , можно использовать символ ~, например:

```
DionisNX(config)# ip access—list mac
DionisNX(config—acl—mac)# permit maclabel level ~0
DionisNX(config—acl—mac)# deny
```

Аналогично, для задания диапазона категорий, используется параметр category.

```
DionisNX(config)# ip access—list mac
DionisNX(config—acl—mac)# permit maclabel level 0 category 0:ffff
DionisNX(config—acl—mac)# deny
```

Значения диапазона категорий задаются в шестнадцатеричной форме, для отрицания используйте символ ~.

```
DionisNX(config)# ip access—list mac
DionisNX(config—acl—mac)# deny maclabel level ~0 category ~0
```

8.4.9 Другие критерии отбора

Синтаксис других критериев отбора описан в списке команд Dionis-NX, ниже приводятся некоторые из них.

Название параметра	Критерий
connlimit	Ограничение числа соединений с одного клиента(или из сети)
syn	syn флаг в TCP-пакете
mac	MAC-адрес источника
tos	Значение TOS

Название параметра	Критерий
dscp	Значение DSCP
datestart, datestop, timestart, timestop, monthdays, weekdays	Время

8.4.10 Протоколирование

Существует возможность протоколирования факта выполнения выбранных правил фильтрации. Для этого необходимо указать параметр: log [all]. Необязательный параметр all указывает на необходимость протоколирования полного тела пакета, а не только заголовка.

```
deny tcp dport 22 log
```

Параметр log - гарантируется запись всех событий в журнал. В случае когда пропускная способность системы выше скорости записи на диск и включена функция протоколирования пакетов - в архитектуру заложена фукнция уничтожения всех пакетов. Данное поведение можно изменить см. раздел Фильтрация и модификация.

Когда процессорное ядро перегружено, траффик, который уже невозможно обработать - уничтожается (drop).”

Настройка подсистемы протоколирования и выборка из журнала рассмотрены в соответствующей главе.

8.4.11 Комментарии

Администратор может комментировать отдельные правила отбора с помощью параметра remark, например:

```
deny tcp dport 22 log remark ”Stop port scanning”
```

8.5 Другие правила списков контроля доступа

Кроме запрета и разрешения трафика (permit/deny), правила в списках могут содержать следующие действия:

Название правила	Параметры	Действие
call	access-list	Передать управление на другой access-list с возвратом
goto	access-list	Передать управление на другой access-list без возврата
log	правила отбора	Протоколировать пакет, не выполняя запрета/разрешения
clone	IP-адрес шлюза	Клонировать пакет и отправить копию в шлюз

8.6 Модификация

Для модификации пакетов используются списки модификации (ip mangle-list). По синтаксису они похожи на списки контроля доступа, однако позволяют отбор пакетов по классу (ip class-map, см. руководство по QoS). Однако при изменении классов возможен небольшой период (несколько секунд), когда классы определяются неверно, поэтому отбор пакетов по классу не реализован в подситстеме фильтрации, связанной с безопасностью.

Для модификации пакетов используется команда mangle в секции ip mangle-list. Возможные операции:

Операция	действие
adjust-mss	Поменять поле mss в пакете SYN
set-tos/set-dscp	Установить поле TOS/DSCP пакета
set-df/clear-df	Установить поле don’t fragment IP-пакета
set/inc/dec-ttl	Изменить поле TTL
set/clear-bso	Изменить поле DoD BSO
set-cos	Установить поле 802.1q CoS

Для отбора пакетов по классу используется параметр class, затем перечисляются классы через запятую. Восклицательный знак перед классом означает отрицание класса. Пример:

```
mangle inc—ttl 1 class cla,!clb
```

Увеличит ttl на 1 для пакетов, попавших в класс cla и одновременно не попавших в класс clb.

При необходимости отбора модифицируемых сообщений по tcp флагам используется ключевое слово tcp-flags, после которого сначала указываются флаги, которые будут исследоваться в процессе отбора, а затем флаги, которые должны быть выставлены, т.е. равны 1 (по умолчанию флаги берутся равными 0).

Например, мы хотим задать tos 0x50 для всех сообщений, которые имеют значение флага syn=1, а значение флага ack=0:

```
adm@DionisNX(config—mangle—mss)# mangle set—tos 0x50 tcp tcp—flags syn,ack syn
```

Если же мы хотим задать tos 0x50 для сообщений, которые имеют значение флага syn=1 и значение флага ack=1, то используем команду:

```
adm@DionisNX(config—mangle—mss)# mangle set—tos 0x50 tcp tcp—flags syn,ack syn,ack
```

8.7 IPv6

Для создания и модификации списков доступа/модификации IPv6 используются команды, аналогичные командам IPv4, с префиксом ip6 вместо ip. Например:

```
ip6 access—list acl
 1 deny src 2001:db8::21
!
interface ethernet 0
 ip6 access—group acl
!
```

8.8 Группирование IP адресов

Зачастую необходимо применить одно и то же правило для достаточно большого набора IP адресов. При этом создание отдельного правила для каждой подсети имеет недостаток: слишком много времени затрачивается на прохождение пакета через эти правила. Команда ip(6) set позволяет создать список подсетей, принадлежность которому проверяется при помощи одного правила ip(6) access/mangle/trace/nat-list, class-map, или policy-route.

ip set состоит из правил match и exclude; единственный аргумент правила — подсеть. Если маска подсети не указана, подразумевается подсеть из одного ip адреса. Если одному ip адресу соответствует несколько правил, из них будет действовать правило с большей длиной префикса подсети (т.е. с меньшей подсетью).

Пример:

```
ip set ipset
 match 192.168.0.0/16
 exclude 192.168.33.0/24
 match 192.168.33.25
!
ip access—list acl
 deny src—ipset ipset
!
```

8.9 Группирование портов

Для транспортных протоколов так же можно создавать группы портов. Для этого необходимо использовать команду ip port-set.

ip port-set состоит только из правил match; единственный аргумент правила — номер порта из диапозона 0-65535.

Пример:

```
ip port—set web
 match 80
 match 443
!
```

```
ip access—list no—web
 1 deny tcp dst—portset web
!
```

8.10 Группирование протоколов и портов

Для транспортных протоколов так же можно создавать группы протоколов и портов с явным указанием транспортного протокола, а так же диапазона портов. Для этого необходимо использовать команду ip proto-port-set.

ip proto-port-set состоит только из правил match; аргументы правила — протокол, порт из диапозона 0-65535 или диапазон портов, комментарий.

Пример:

```
ip proto—port—set myset
 match tcp 80
 match tcp 443
 match udp 67 68 remark ”match dhcp”
!
ip access—list deny—myset
 1 deny dst—proto—portset myset
!
```

8.11 Группирование IP адресов, протоколов и портов

Так же можно группировать IP адрес, протокол и порт. Для этого необходимо использовать команду ip ip-port-set

ip ip-port-set состоит из правил match и exclude; аргументы правила — подсеть, протокол, порт из диапозона 0-65535, комментарий.

Пример:

```
ip ip—port—set myset
 exclude 192.168.1.0/24 tcp 443
 match 192.168.1.0/24 tcp 53 remark ”match dns”
 match 192.168.1.0/24 tcp 80
 match 192.168.1.0/24 udp 53 remark ”match dns”
!
ip access—list deny—myset
 1 deny dst—ipportset myset
!
```

9. Многоадресная передача

Dionis-NX может использоваться для организации многоадресной передачи (далее МП) на уровне IP стека TCP/IP.

МП используется в тех случаях, когда по сети необходимо передавать нескольким пользователям одну и ту же информацию. Такая потребность может возникнуть, например, при распространении через IP-сеть телевизионного или радиосигнала, при организации телеконференций или селекторных совещаний. В этом случае можно не открывать отдельные соединения с каждым клиентом сети, с тем чтобы передавать по ним одинаковую информацию, а рассылать IP-пакеты без лишнего дублирования, но так, чтобы их получали все клиенты.

В Dionis-NX можно настраивать динамическую и статическую маршрутизацию.

В данном подразделе МП будет рассматриваться на основе динамической многоадресной маршрутизации(далее ММ).

Основные понятия МП:

- передача пакетов: пакеты МП рассылаются не отдельным узлам, а группе узлов;
- адресация группы: группа определяется одним групповым IP-адресом класса D в диапазоне 224.0.0.0–239.255.255.255:
  - 224.0.0.0/8 - зарезервированные IANA-адреса;
  - 232.0.0.0/8 - глобальное адресное пространство для МП, специфичной для конкретного источника (SSM, реализован в IGMPv3);
  - 233.0.0.0/8 - глобальное адресное пространство GLOP для групп внутри автономных систем (232.AA.AA.GG, где AAAA - 16 бит, включающих номер AS, GG номер группы в AS);
  - 239.0.0.0/8 - локальное адресное пространство для закрытых (частных) сетей; аналог LAN одноадресного пространства адресов;
- адрес источника не может быть адресом класса D;
- членство в группе: узлы сети могут входить в группу МП (далее группу) и выходить из нее по своему желанию;
- адресность: многоадресные датаграммы (далее МД) посылаются группе и только члены этой группы получают их.

Организация МП в Dionis-NX осуществляется тремя протоколами:

- управление группами при помощи протокола IGMPv3 (Протокол управления группами Интернет, версия 3);
- маршрутизация МД при помощи протокола DVMRP (Дистанционно-векторный протокол многоадресной маршрутизации);
- маршрутизация МД при помощи протокола PIM-SM (Независимая от протокола многоадресная передача (Разреженный режим)) с поддержкой SSM (МП с заданным источником)

Рассмотрим в общих чертах механизм МП:

- исходный сетевой узел (например узел,транслирующий видеофильм) посылает МД на групповой адрес А (например по UDP- или RTP-протоколу);
- сетевые узлы назначения (клиенты, желающие смотреть данный видеофильм) сообщают маршрутизатору по протоколу IGMP о желании присоединиться к группе А;
- маршрутизатор при получении МД определяет, нужно ли ее пересылать дальше (протокол DVMRP или PIM);
- если МД нужно пересылать дальше, маршрутизатор определяет, на какой именно интерфейс ее послать, чтобы она быстрее достигла получателей (протокол DVMRP или PIM) из группы А.

Основные особенности протокола IGMP:

- служит для обмена информацией о членстве в группах между IP-маршрутизаторами, поддерживающими МП, и членами групп;
- узлы сами сообщают маршрутизаторам о своем членстве в группах (IGMP-сообщение REPORT);
- узлы сами сообщают маршрутизаторам о выходе из группы (IGMP-сообщение LEAVE);
- состояние членства узлов в группах периодически проверяется маршрутизаторами, поддерживающими МП (IGMP-сообщение QUERY)

Основные особенности протокола DVMRP:

- относится к внутренним протоколам маршрутизации, пригодным для использования в пределах автономной системы;
- обеспечивает эффективный механизм доставки МД хостам,входящим в группы, без организации соединений;
- использует сообщения протокола IGMP для обмена информацией с другими маршрутизаторами,поддерживающими МП.

Эффективность маршрутизации в протоколе DVMRP осуществляется путем использования алгоритма RPM (Reverse Path Multicasting):

- динамически генерирует деревья групповой доставки МД;
- если в зоне ответственности маршрутизатора нет членов группы, тогда маршрутизатор отсекает ненужные ветки дерева рассылки (pruning);
- сохраняет информации о пути возврата к отправителю МД (передача маршрутов).

Основные особенности протокола PIM-SM:

- используется для сетей с произвольным рассредоточением пользователей с ограниченной пропускной способностью сетевых каналов;
- эффективная поддержка работы «рассеянных» мультикастинг-групп: группы из разных автономных систем, находящихся на разных континентах;
- построение дерева маршрутов, разветвляющегося как можно ближе к получателям МД;
- передача трафика идет только по явному запросу.
- для групп 232.0.0.0/8 используется режим SSM - клиент, помимо указания группы, может указать адрес источника МП, от которого хочет принимать МД.

9.1 Общие сведения о настройке многоадресной маршрутизации

В Dionis-NX имеется возможность настраивать:

- статическую ММ: командой ip mroute;
- динамическую ММ на основе IGMPv2/v3: командой router igmp;
- динамическую ММ на основе DVMRP (с поддержкой IGMPv3): командой router dvmrp;
- динамическую ММ на основе PIM-SM (с поддержкой IGMPv3): командой router pim.

Одновременно в системе может быть настроена только одна из четырех типов ММ.

9.2 Настройка протокола DVMRP

Данная настройка включает также поддержку протокола IGMPv3, нужного для работы DVMRP.

Включение ММ на основе DVMRP осуществляется командой:

```
(config)# router dvmrp
```

Если данная команда успешно выполнилась, по умолчанию все доступные для МП интрефейсы НЕ будут участвовать в МП.

Чтобы интерфейс участвовал в МП необходимо выполнить команду

```
(config—dvmrp)# iface ethernet 0
```

Это включит интерфейс ethernet0 в участие в МП и ММ.

9.2.1 Настройка параметров интерфейсов

Рассмотрим настройки на примере интерфейса ethernet0. Для настройки параметров интерфейса сначала нужно войти в конфигурацию соответствующего интерфейса:

```
(config)# router dvmrp
(config—dvmrp)# iface ethernet 0
```

Основные параметры МП, которые могут быть настроены для интерфейсов:

- метрика: задает стоимость прохождения МД через данный интерфейс;
- порог TTL: минимальное значение IP TTL для МД, нужное для прохода этой МД через данный интерфейс;
- пропускная способность многоадресного трафика.

Чтобы настроить метрику для интерфейса, следует выполнить команду:

```
(config—dvmrp—ethernet0)# metric 1
```

Для метрики следует устанавливать как можно меньшее значение, т.к. максимально сумма всех метрик маршрута МД в сети не может превышать 31. По умолчанию: 1.

Чтобы настроить порог TTL для интерфейса, следует выполнить команду:

```
(config—dvmrp—ethernet0)# threshold 5
```

По умолчанию: 1.

Если upstream-интерфейс (интерфейс к источнику) соединен с разными подсетями, опишите эти подсети следующей командой::

```
(config—dvmrp—ethernet0)# subnet 10.0.0.0/24
```

Полностью запретить ММ на интерфейсе можно с помощью следующей команды:

```
(config—dvmrp—ethernet0)# disable
```

9.2.2 Настройка ограничений

Обычно настройки порога TTL, приведенные в предыдущем подразделе, используются для достижения следующих целей:

- ограничить время жизни МД;
- уменьшить трафик из-за ограничений пропускной способности сети;
- уменьшить трафик для целей повторного использование адресов и приватности.

Для третьей цели лучше подходит использование не ограничения по полю TTL, а назначение определённого группового адреса как административной границы (далее АГ). Интерфейс, которому назначена АГ, не будет принимать и передавать МД, направленные адресам, принадлежащими АГ.

АГ определяется следующим интервалом групповых адресов: 239.0.0.0-239.255.255.255. Эти адреса могут быть использованы и назначены только внутри автономных систем, где гарантируется их уникальность.

Рассмотрим пример назначения АГ интерфейсу:

```
(config—dvmrp)# boundary bo1 239.255.1.0/24
(config—dvmrp—ethernet0)# boundary bo1
```

Эти команды определяют для интерфейса ethernet0 АГ 239.255.1.0/24. Любой трафик МП с адресом назначения из указанной АГ не разрешён к передаче через данный интерфейс в обоих направлениях.

9.2.3 Настройка протоколирования

Для включения режима протоколирования динамической ММ следует выполнить команду:

```
(config—dvmrp)# log <TYPE>
```

Параметр TYPE задает тип протоколируемой информации и может принимать следующие значения:

- packet : входящие/исходящие пакеты;
- prune : отсечение маршрутов;
- route : маршрутизационные сообщения;
- route-details : более детальная информация о маршрутизации;
- peer : взаимодействие соседей (маршрутизаторов) между собой;
- route-cache : кэширование маршрутов;
- timeout : таймауты;
- interface : виртуальные интерфейсы;
- group : группы;
- mtrace : многоадресный traceroute;
- igmp : IGMP-сообщения;
- icmp : ICMP-сообщения;
- rsrr : RSRR-сообщения;
- default : igmp,route,route-cache,prune,peer,interface,group информация;
- all : все перечисленные типы информации.

9.3 Настройка протокола PIM

Далее под протоколом PIM будем понимать этот протокол в режиме SM (Sparse mode). Данная настройка включает также поддержку протокола IGMPv3, нужного для работы PIM. Основное отличие от PIM DM режима в том, что в PIM SM маршрутизаторы добавляются в дерево МП, только если они явно отправляют PIM-Join. Однако, поскольку нет флудинга информация в SM режиме по PIM-домену, то необходимо как-то сообщать маршрутизаторам о том, кому направлять PIM-Join сообщение. Для этого выбирается RP-маршрутизатор (точка встречи, Rendezvous Point) - либо выборами, либо статическим указанием IP-адреса точки встречи на всех PIM-маршрутизаторах.

Рассмотрим основные понятия протокола:

- режим SM протокола PIM - используется для сетей с произвольным рассредоточением пользователей с ограниченной пропускной способностью сетевых каналов;
- PIM-маршрутизатор - маршрутизатор,например Dionis-NX, поддерживающий протокол PIM (далее будем под PIM понимать PIM-SM);
- PIM-домен - набор смежных PIM-маршрутизаторов, сконфигурированных для совместной работы в рамках границ, определенных пограничными маршрутизаторами PMBR, соединяющими PIM-домен с остальным Интернет;

- PMBR-маршрутизатор - пограничный PIM-маршрутизатор; размещается на границе PIM-домена и взаимодействует с другими типами мультикаст-маршрутизаторов;
- точка встречи (RP) - PIM-маршрутизатор разветвления маршрута для потока данных; каждая мультикаст-группа должна иметь RP; RP выбираются динамически, либо назначаются статически; отправители используют RP для объявления о своем существовании, а получатели, чтобы узнать о новых отправителях путем посылки Join PIM-сообщений;
- дерево кратчайших маршрутов (SPT) - описывает кратчайший путь от RP к источнику МД; обозначается как (S,G) - для каждой пары источник(S)-группа(G) строится свое SPT;
- дерево точки встречи (RPT): дерево кратчайших маршрутов от RP к получателям МД; обозначается как (*,G) - т.к. строится вне зависимости от адреса источника S, а только в зависимости от группы G;
- RPF - Маршрутизация Обратного Пути: МД пересылается на все интерфейсы, кроме того, с которого пришел, только если источник МД доступен через интерфейс получения данной МД (есть маршрут к источнику через интерфейс получения);
- выделенный маршрутизатор (DR) выбирается (по приоритету и затем по максимальному IP-адресу) из маршрутизаторов,подсоединенных к одной и той же сети с МП; он ответственен за посылку сообщений Join,Prune,Register к RP в данном сегменте сети; выбирается для того,чтобы только он передавал МД данной группы в данную сеть,что бы избежать дублирования пакетов;
- NHR - ближайший к получателю PIM-маршрутизатор;
- NHS - ближайший к источнику PIM-маршрутизатор;
- вышестоящий маршрутизатор - маршрутизатор, расположенный ближе к источнику МД;
- нижестоящий маршрутизатор - маршрутизатор, расположенный ближе к получателю МД;
- BSR - маршрутизатор, отвечающий за рассылку bootstrap сообщений; содержит полный список C-RP домена,который рассылается по домену на адрес 224.0.0.13; должен быть хотя бы один BSR, иначе информация о C-RP будет неизвестна маршрутизаторам сети;
- C-RP - кандидат в RP-маршрутизаторы; среди маршрутизаторов,объявленных как C-RP, происходит выбор RP по приоритету и затем по величине IP-адреса;
- C-BSR - кандидат в BSR-маршрутизаторы; среди маршрутизаторов,объявленных как C-BSR, происходит выбор BSR по приоритету и затем по величине IP-адреса;
- основные PIM-сообщения (юникастные):
  - Join - присоединение маршрутизатора к дереву маршрутов; сообщение посылается,если пакет получен на интерфейсе, прошедшем проверку RPF и есть локально присоединенные хосты или нижестоящие маршрутизаторы, желающие получать трафик данной группы;
  - Prune - отсоединение маршрутизатора от дерева маршрутов; сообщение посылается,если пакет получен на интерфейсе, не прошедшем проверку RPF и/или нет локально присоединенных хостов или нижестоящих маршрутизаторов, желающих получать трафик данной группы.
  - Register - сообщение посылается, когда источник отправляет данные группе в первый раз, его DR посылает это сообщение, в которое вкладывает МД источника
  - Register-stop - сообщение от RP к DR, в котором говорится,что не нужно больше инкапсулировать МД в Register-сообщение, т.к.:
    * в случае если есть получатели МД данной группы: на RP уже передается МД от источника по SPT, построенного после получения Register сообщения;
    * нет получателей МД данной группы;
  - Candidate-RP-Advertisement - C-RP периодически высылает в адрес BSR данное сообщение

об обслуживаемых группах; BSR собирает эти данные и распространяет их далее по PIM-домену в сообщении bootstrap;

- bootstrap - сообщения, которые воспринимаются всеми PIM-маршрутизаторами для получения RP-информации (о том, какие RP отвечают за какие группы) и для динамического выбора BSR-маршрутизатора; это многоадресные сообщения на адрес 224.0.0.13 (All-PIM-routers).

Таким образом:

- каждая мультикаст-группа должна иметь хотя бы один C-RP;
- PIM-SM-домен должен иметь хотя бы один C-BSR, если только все маршрутизаторы домена не имеют статически заданной информации о RP всех доменов;
- каждая подсеть должна иметь хотя бы один DR.

Рассмотрим подробнее алгоритм работы PIM-SM:

Фаза 1. Выбор кратчайшего маршрута.

1. пусть хост посылает IGMP-Join-сообщение J1 на DR, который не является RP для указанной в сообщении группы G1;
2. DR отправителя шлет сообщение J1 по направлению к RP для группы G1 («upstream»), он определяет это из последнего присланного от BSR bootstrap-сообщения;
3. каждый PIM-маршрутизатор, через который проходит сообщение J1, записывает, что существуют члены группы J1 на входящем интерфейсе;
4. в результате J1 доходит либо до RP, либо до другого маршрутизатора, за которым есть члены группы;
5. если сообщения сходятся в RP и есть много членов группы G1, то эти сообщения Join формируют RPT, на основе информации от DR получателей, в результате образуются эффективные короткие маршруты пересылки МД к получателям
6. DR отправителя начинет посылку МД, вкладывая МД-данные источника МД в юникаст-пакет PIM Register и посылая данный PIM Register на RP группы;
7. RP группы, получив пакет PIM Register, дэинкапсулирует МД из пакета и посылает его по сформированному на основе RPT маршруту.

Фаза 2. Повышение эффективности и скорости посылки.

1. получив PIM Register, RP выбирает SPT к отправителю, в результате чего МД больше не нужно регистрировать на RP и, как следствие, инкапсулировать в PIM Register;
2. RP отправляет PIM RegisterStop сообщение в ответ на следующее инкапсулированное сообщение от DR отправителя.
3. DR, получив PIM RegisterStop сообщение, прекращает регистрацию/инкапсуляцию МД и посылает нулевое Register-сообщение, которое является вопросом к RP: «Все еще не требуются Register-сообщения?»;
4. если RP отвечает на нулевое Register сообщение сообщением RegisterStop, то DR начинает посылку оригинальных (неинкапсулированных) МД;

5. если DR не получает другого RegisterStop сообщения в течение некоторого периода времени, то DR продолжает посылать Register сообщения.
6. как только RP начинает получать оригинальные МД от источника, RP начинает перенаправлять их по кратчайшему RPT-маршруту к получателям.

Рассмотрим пример МП посредством PIM-SM (см. рис. 9.1).

Рис. 9.1: Схема МП PIM-SM

1. предположим,что все PIM-маршрутизаторы уже имеют информацию о расположении RP и поддерживаемых ими групп (посредством bootstrap-сообщений или путем статического назначения RP на PIM-маршрутизаторах);

2. SRC - источник начинает трансляцию группы G

3. RCV - получатель заявляет о желании получать трафик группы G: шлет сообщение IGMP-Join(G) в сторону ближайшего PIM-маршрутизатора

4. NHR - ближайшие к получателю PIM-маршрутизаторы, получив IGMP-Join(G) сообщение:

   - шлют сообщение PIM-Join(*,G) в сторону RP на адрес 224.0.0.13;
   - в результате строится дерево RPT (дерево точек встречи: дерево с вершиной RP и ветками в виде PIM-маршрутизаторов связанных с получателями трафика);
   - в каждой вершине дерева находятся RP или PIM-маршрутизаторы и все они имеют информацию о получателях трафика в виде (*, G) (см. команду ‘show multicast pim’, таблица (*, G));

5. NHS - ближайший к источнику PIM-маршрутизатор, который выбирается в качестве DR (выбор через сообщения PIM-Hello по старшинству IP-адреса), регистрируется на RP (из п.1 он знает,какая RP отвечает за группу G): это называется регистрация источника:

   - посылка пакета PIM-Register, в которое инкапуслирована МД источника, в сторону RP

   - в результате на RP будет информация о паре (S,G), т.е. пара “источник-группа” в соответствующей (S,G)-таблице (см. команду ‘show multicast pim’, таблица (S, G));
   - далее RP шлет сообщение PIM-Register-Stop в сторону NHS

   - NHS получив PIM-Register-Stop запускает 1-минутный таймер Register-Suppression, по истечении 55 сек. отправляет на RP Register пакет с флагом Null-Register (без инкапсуляции), которое является своего рода вопросом к RP: «Все еще не требуются Register-сообщения?»

- если у RP нет данных о получателях трафика, то снова отправляется Register-Stop (нужно же ответить на Null-Register пакет) и переходим к предыдущему пункту (запуск на NHS таймера Register-Suppression)

- если у RP есть данные о получателе трафика, то RP не отвечает на Null-Register, в результате по истечении таймера переход на начало п.3 - посылка инакпулированного PIM-Register

- МП идет по деревьям: инкапсулированные МД - между NHS и RP ;и чистые МД через RPT - между RP и NHR; но нужно передавать везде чистые МД

- далее NHS должен начать передавать обычные,не инкапуслированные МД, для этого происходит переключение на SPT дерево после получения определенного количества данных от одного источника получателем - это задается командой spt-threshold; если RP знает о наличии клиентов-получателей трафика (есть записи (*, G), см. п.5), то RP построит дерево SPT к источнику

- дерево SPT: RP отправляет сообщение PIM-Join(S,G) в сторону NHS, причем S = IP адресу NHS (от кого приходят Register-сообщения), IP получателя = 224.0.0.13 (все PIM-маршрутизаторы) с ttl=1; путь по которому прошел Join от RP к источнику превращается в дерево SPT и далее NHS начинает вещать чистыми МД; однако параллельно передается МД инкапуслированные в Register

- RP получает чистый МД, видя по (S,G) и по интерфейсу получения, что это тот же МД-трафик, что приходит к ней в Register, она отправляет PIM-Register-Stop на NHS - говорит,что нужно прекратить посылку инкапуслированных МД в Register.

Включение ММ на основе PIM осуществляется командой:

```
(config)# router pim
```

В результате все доступные для МП интерфейсы НЕ будут участвовать в МП.

Чтобы интерфейс участвовал в МП необходимо выполнить команду

```
(config—pim)# iface ethernet 0
```

Это включит интерфейс ethernet0 в участие в МП и ММ.

9.3.1 Глобальные настройки

iface <IFACE>

Добавляет интерфейс IFACE к участию в МП по протоколу PIM и входит в настройки интерфейса по части МП. Если не выполнить данную команду, указанный интерфейс не будет участвовать в МП.

default-route-distance <VAL>

Задает административное расстояние при выборе оптимального PIM-forwarder маршрутизатора, в случае когда существует несколько PIM-маршрутизаторов, перенаправляющих поток МД от источника.

Выборы осуществляются посредством PIM Assert механизма: PIM-маршрутизатор с меньшим значением расстояния будет избран как PIM-forwarder для данной подсети. Если default-route-distance одинакова, то выбирается маршрутизатор с численно максимальным IP-адресом на мультикаст-интерфейсе, через который идет поток МД.

По умолчанию: 101.

bsr-cand [LOCAL_IP] [PRIO]

Устанавливает параметры C-BSR. Данная система Dionis-NX объявляется как кандидат в BSR.

Параметры:

- LOCAL_IP - один из локальных IP-адресов; по умолчанию: выбирается максимальный IP-адрес;
- PRIO - приоритет C-BSR; указывает насколько важен данный кандидат при выборе; чем ниже значение,тем выше приоритет.

По умолчанию: 255

rp-cand [LOCAL_IP] [period TIME] [priority PRIO]

Устанавливает параметры C-RP данной системы. Данная система Dionis-NX объявляется как кандидат в RP.

Параметры:

- LOCAL_IP - один из локальных IP-адресов; по умолчанию: выбирается максимальный IP-адрес;
- PRIO - приоритет C-RP; указывает насколько важен данный кандидат при выборе; чем ниже значение,тем выше приоритет;
- TIME - период времени между посылками PIM-сообщения Candidate-RP-Advertisement (сообщает BSR об RP); данное PIM сообщение ,будучи полученным,воспринимается только BSR для обновления знания об RP-узлах и поддерживаемы ими группах.

По умолчанию: приоритет: 0, период: 60сек.

group <IP/MSK>

Задает группу, за которую будет отвечать данная C-RP. Имеет смысл,только если указана команда rp-cand.

rp-static <IP> <GRP> [PRIO]

Задает статически C-RP и поддерживаемую ей группу.

Параметры:

- IP - IP-адрес RP;
- GRP - многоадресная группа, которую обслуживает указанная RP;
- PRIO - приоритет C-RP; указывает насколько важен данный кандидат при выборе; чем ниже значение,тем выше приоритет.

При использовании rp-static необходимо задать ее на каждом PIM-маршрутизаторе.

По умолчанию: приоритет: 0.

log [TYPE{1,12} | all]

Включает лог:

- TYPE - различные подсистемы (до 12 шт.)
- all - самый подробный лог.

Без параметров означает log default (pim, igmp, dvmrp).

spt-threshold <packets|rate> <VAL> [INTERVAL]

Устанавливает порог перехода с RPT на SPT для DR- и RP-маршрутизаторов.

Параметры:

- VAL - порог скорости трафика (байт/сек) для режима rate и количество пакетов для режима packets (0);
- INTERVAL - интервал проверки превышения порога (100 сек.)

Если сообщения Register приходят на RP со скоростью выше RATE, RP шлет DR-сообщение RegisterStop и добавляет SPT-маршрут для передачи МД от источника.

Если сообщения Register приходят на NHR со скоростью выше RATE, DR также переходит на SPT-маршрут.

Количество пакетов обозначает, сколько пакетов PimRegister должно придти начиная с 1го PimRegister пакета за указанный интервал. Поэтому значение packets 0 означает, что переход на SPT начинается уже после первого PimRegister пакета.

По умолчанию: packets: 0; rate: 6250 байт/сек; интервал: 100 сек.

9.3.2 Настройка параметров интерфейса

Рассмотрим настройки на примере интерфейса ethernet0. Для настройки параметров интерфейса сначала нужно войти в конфигурацию соответствующего интерфейса:

```
(config)# router pim
(config—pim)# iface ethernet 0
```

Чтобы настроить приоритет для интерфейса выполните команду

```
(config—pim—ethernet0)# priority 1
```

По умолчанию: 1.

Приоритет влияет на выбор данного маршуртизатора как DR для данной сети LAN. Чем меньше данное значение, тем более вероятен выбор данного маршрутизатора как DR.

Чтобы настроить порог TTL для интерфейса выполните команду

```
(config—pim—ethernet0)# threshold 5
```

По умолчанию: 1. Команда задает минимальное значение IP TTL, требуемое для пересылки МД через данный интерфейс.

Если upstream-интерфейс (интерфейс к источнику) соединен с разными подсетями, опишите эти подсети следующей командой:

```
(config—pim—ethernet0)# subnet 10.0.1.0/24
```

Если не указывать данной команды, то данный интерфейс будет обслуживать трафик только первичной подсети (заданной командой ip address в настройках интерфейса).

Чтобы запретить распространение МД указанной группы через интерфейс выполните команду:

```
(config—pim—ethernet0)# boundary 239.0.0.0/24
```

В данном случае запрещается передача МД группе 239.0.0.0/24. Команду рекомендуется использовать на PMBR-маршрутиазторах для создания границ распространения МД определенных групп.

Следующей командой вы можете полностью запретить МП на интерфейсе

```
(config—pim—ethernet0)# disable
```

Это равносильно удалению интерфейса из конфигурации router pim, однако в данном случае все прочие настройки интерфейса сохраняются,что более удобно, если в дальнейшем потребуется вновь включить интерфейс.

9.3.3 Пример настройки

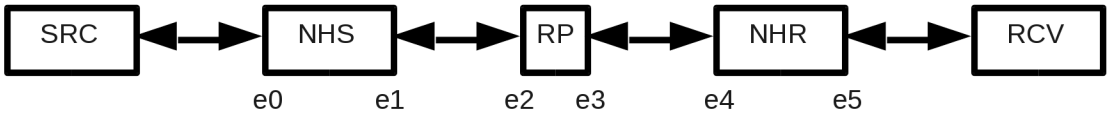

Рис. 9.2: Пример стенда PIM-SM

Пусть вещателя SRC и клиента RCV разделяет сеть из 3х маршрутизаторов (см. рис. 9.2). SRC вещает на группу 239.0.0.1/32. На схеме eN - обозначается интерфейс ethernetN, где N - его номер. Возможный упрощенный вариант настройки представлен далее.

Настройки NHS:

```
NHS(config—pim)# iface ethernet0
NHS(config—pim)# iface ethernet1
```

NHS работает как обычный PIM-маршрутизатор.

Настройки RP:

```
RP(config—pim)# iface ethernet2
RP(config—pim)# iface ethernet3
RP(config—pim)# rp—cand
RP(config—pim)# bsr—cand
RP(config—pim)# group 239.0.0.1/32
```

Настройки NHR:

```
RP(config—pim)# iface ethernet4
RP(config—pim)# iface ethernet5
RP(config—pim)# rp—cand
RP(config—pim)# bsr—cand
RP(config—pim)# group 239.0.0.1/32
```

RP и NHR работают как C-RP и C-BSR.

9.4 Настройка протокола IGMP

Включение ММ на основе только IGMP (без протокола динамической ММ) осуществляется командой:

```
(config)# router igmp
```

Если данная команда успешно выполнилась, по умолчанию все доступные для МП интрефейсы НЕ будут участвовать в МП.

Чтобы интерфейс участвовал в МП, необходимо определить один входящий интерфейс МП и один или более исходящих интерфейсов МП.

Определяем входящий интерфейс (интерфейс от источника многоадресного трафика):

```
(config—igmp)# input—iface
(config—igmp—in)# iface gre 1
```

Определяем исходящий интерфейс (интерфейс к потенциальным слушателям многоадресного трафика):

```
(config—igmp)# output—iface ethernet 0
```

Если в результате каких-либо настроек ММ на основе IGMP включится, будучи до этого выключенной (из-за недостаточных настроек), будет выведено сообщение: «Info: [igmp] igmp multicast routing enabled»

Если в результате каких-либо настроек ММ на основе IGMP выключится, будучи до этого включенной, будет выведено сообщение: «Info: [igmp] igmp multicast routing disabled»

9.4.1 Настройка интерфейса

Рассмотрим настройки интерфейса на примере исходящего интерфейса ethernet0:

```
(config)# router igmp
(config—igmp)# iface ethernet 0
```

Чтобы настроить порог TTL для интерфейса, следует выполнить команду

```
(config—igmp—out—ethernet0)# threshold 5
```

МД с TTL меньше указанного будут отбрасываться. По умолчанию: 1.

Чтобы настроить максимальную пропускную способность многоадресного трафика для интерфейса, следует выполнить команду

```
(config—igmp—out—ethernet0)# rate 100
```

Значение задается в Кбит/сек. По умолчанию: неограниченно.

Данные опции возможны и для входящего интерфейса. Однако для него добавляется дополнительная опция.

Если upstream-интерфейс (интерфейс к источнику) соединен с разными подсетями, опишите эти подсети следующей командой:

```
(config—igmp—in)# subnet 10.0.0.0/24
```

9.4.2 Настройка протоколирования

Для включения протоколирования следует использовать следующую команду:

```
(config—igmp)# log [debug]
```

Необязательный параметр debug задает более подробный протокол.

9.4.3 Пример

Рассмотрим пример настройки IGMP (см. рис 9.3).

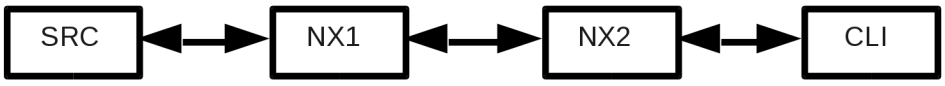

Рис. 9.3: Схема МП на основе IGMP

В данном примере:

- src - это источник МП, с адреса 10.0.0.0/24;
- cli - получатель МП;
- nx1,nx2 - маршрутизаторы Dionis-NX;
- между src и nx1: интерфейс ethernet0 на nx1;

- между nx1 и nx2: интерфейс gre1 на nx1 и nx2;
- между nx2 и cli: интерфейс ethernet1 на nx2.

Настройка МП на основе двух igmp-маршрутизаторов Dionis-NX:

- настройки nx1:

```
(config—igmp)# input—iface
(config—igmp—in)# iface ethernet 0
(config—igmp)# output—iface gre 1
```

- настройки nx2:

```
(config—igmp)# input—iface
(config—igmp—in)# iface gre 1
(config—igmp—in)# subnet 10.0.0.0/24
(config—igmp)# output—iface ethernet 1
```

9.5 Настройка статической многоадресной маршрутизации

Статическая ММ возможна только при отключенной динамической ММ (отсутствуют команды router pim, router igmp, router dvmrp).

Для настройки статического маршрута следует ввести команду:

```
(config)# ip mroute <IFACE_IN> <MC_SRC_IP> <GROUP_IP> <IFACE_OUT>{1,8}
```

Параметры:

- IFACE_IN : интерфейс, откуда приходят МД (должен иметь IP-адрес)
- MC_SRC_IP : IP-адрес источника МД
- GROUP_IP : групповой адрес
- IFACE_OUT : интерфейсы, куда должны направляться МД: может быть до 8 штук (должны иметь IP-адреса)

9.6 Мониторинг работы многоадресной маршрутизации

Мониторинг работы многоадресной маршрутизации осуществляется командой show multicast режима enable и ее подкомандами.

```
# show multicast \<log | routes | vifs | groups | pim | dvmrp [groups|cache] \>
```

Параметры:

- log - протоколы, в которых регистрируется работа МП;
- routes - таблица многоадресных маршрутов (для динамической и статической МП);
- vifs - таблица VIF-ов: интерфейсов используемых в МП (для динамической и статической МП);
- groups - IGMP-информация о МП, количество присоединений к мультикаст-группам на интерфейсах;
- pim - PIM-информация о МП;
- dvmrp - DVMRP-информация о МП:
  - groups - информация о группах DVMRP;
  - cache - информация о маршрутах DVMRP.

VIF - это виртуальный интерфейс, участвующий в МП и который на самом деле отображается на реальный интерфейс или локальный конец туннеля в системе.

9.7 Работа со службой

Для перезапуска службы МП выполните команду

```
# router <igmp|pim|dvmrp> restart
```

9.8 Пример настройки видео-трансляции при помощи VideoLAN Vlc

В примерах данной главы использовались источники и получатели МП.

Покажем, как можно осуществить передачу и прием МП при помощи ПО VidoeLAN Vlc.

Для трансляции видео выполните на источнике МП:

- **vlc -vvv c:\video.avi --sout “#udp{dst=239.0.0.1:1234}” --ttl 5 --sout-all**

Сервер осуществляет МП из файла c:\video.avi на мультикаст-адрес 239.0.0.1:1234 по протоколу UDP, причем TTL для МД устанавливается в значение 5, чтобы указанная МД достигла получателей.

Для получения трансляции выполните на получателе МП:

- **vlc udp://@239.0.0.1:1234**

9.9 Сокращения и термины

В данной главе используются следующие сокращения:

- ММ - многоадресная маршрутизация
- МП - многоадресная передача
- МД - многоадресная датаграмма

10. NAT

NAT (Network Address Translation) — это механизм в сетях TCP/IP, позволяющий преобразовывать IP-адреса в заголовках пакетов. См. рис. 10.1. Различают два типа NAT: SNAT – замена адреса источника, и DNAT – замена адреса назначения.

SNAT используется для предоставления пользователям локальной сети с внутренними адресами доступа к внешней сети. DNAT используется для доступа из внешней к ресурсам внутренней.

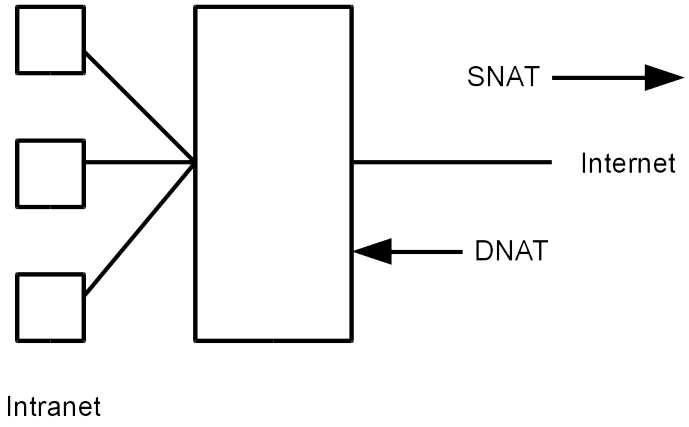

Рис. 10.1: Механизм NAT

В Dionis-NX NAT выполняется всегда на внешнем интерфейсе. При этом, SNAT начинает применяться для исходящего трафика, а DNAT – для входящего.

Очевидно, что и при SNAT и при DNAT для разных направлений трафика меняются как адреса источника, так и назначения. Например, в случае с преобразованием SNAT, исходящий с внешнего интерфейса пакет будет подвергнут изменению – его адрес источника будет заменен. При ответе, пакет также попадет в логику SNAT (для того, чтобы попасть во внутреннюю сеть), однако логика сопоставления внутренних адресов выполняется один раз для всего потока, и это происходит при выходе пакета с внешнего интерфейса.

Таким образом, преобразование SNAT для входящих во внешний интерфейс пакетов выполняется только для пакетов уже установленных изнутри соединений. Аналогично, преобразование DNAT для исходящих из внешнего интерфейса пакетов выполняется только если эти пакеты ассоциированы с установленным ранее соединением извне, подверженному преобразованию DNAT.

Следует иметь в виду, что если задано преобразования DNAT относительно какого-либо IP-адреса назначения, предполагается, что сетевые пакеты с таким адресом назначения дойдут до сетевого интерфейса. Чаще всего это означает необходимость задания дополнительного ip-адреса (ip secondary-address) для интерфейса, с которым связано преобразование DNAT.

Для созданий правил трансляции NAT в Dionis-NX используются списки NAT (ip nat-list).

Следует иметь в виду, что правила отбора в списках NAT применяются не для каждого пакета в отдельности, а только для первого пакета, устанавливающего соединение. NAT-преобразования пакетов выполняется только тогда, когда они принадлежат какому-либо соединению. Пакеты, не попадающие ни в одно соединение, считаются некорректными и не будут подвержены NAT-преобразованию.

10.1 Создание ip nat-list

Работа со списками NAT во многом совпадает с работой со списками контроля доступа. Так, для создания (редактирования) списка NAT необходимо в режиме configure выполнить следующую команду:

```
DionisNX(config)# ip nat—list mynat
```

При этом, произойдет переход в режим редактирования списка. Список содержит набор NAT-правил. Синтаксис правила выглядит следующим образом: nat <критерии отбора> <тип NAT> [параметры NAT], где критерии отбора являются подмножеством критериев списков контроля доступа и могут содержать:

Название	Значение
Протокол	IP-протокол потока
src	Адрес(а) источника потока
dst	Адрес(а) приемника потока
sport	Порт(ы) источника потока (для TCP и UDP)
dport	Порт(ы) приемника потока (для TCP и UDP)

Параметр <тип NAT> задает тип преобразования. Основные типы: snat, dnat или masquerade. MASQUERADE - это такой тип SNAT, который меняет адрес источника пакета (при выходе с внешнего интерфейса) на текущий адрес интерфейса.

Для преобразований snat и dnat необходимо указать ip-адрес замены, для преобразования masquerade этого не требуется. Например:

```
DionisNX(config)# ip nat—list mynat
DionisNX(config—nat—mynat)# nat src 192.168.0.0/24 masquerade
DionisNX(config—nat—mynat)# nat tcp dport 80 dnat ip 192.168.0.1
```

Для работы с элементами списка можно использовать числовые префиксы, также как и при работе со списками контроля доступа. Например:

```
DionisNX(config—nat—mynat)# do show
1 nat src 192.168.0.0/24 masquerade
2 nat tcp dport 80 dnat ip 192.168.0.1
DionisNX(config—nat—mynat)# no 1
DionisNX(config—nat—mynat)# do show
1 nat tcp dport 80 dnat ip 192.168.0.1
```

Для просмотра информации о NAT-списках, существует две команды, доступные из enable-режима.

show ip nat-list <имя|*> config	Информация о действующей конфигурации
show ip nat-list [имя]	Низкоуровневая информация из ядра ОС

Если в командах имя списка задано как *, будет показана информация о всех списках.

Например (из режима configure):

```
DionisNX(config)# do show ip nat—list mynat config
```

10.2 Другие типы NAT

Кроме основных типов преобразований (snat, dnat, masquerade) существуют также другие:

Название	Параметры	Действие
exclude in	критерии отбора	Исключает входящий трафик из NAT преобразования
exclude out	критерии отбора	Исключает исходящий трафик из NAT преобразования
redirect	для протоколов tcp/udp задается port <номер порта>	Перенаправлять трафик на локальный хост:порт (DNAT)
netmap src	критерии отбора, ip <адрес сети>	Отобразить целую сеть (SNAT)
netmap dst	критерии отбора, ip <адрес сети>	Отобразить целую сеть (DNAT)

10.3 Привязка ip nat-list

Создание списка преобразований NAT не означает то, что список начинает действовать. Для того, чтобы правила NAT начали действовать на проходящий трафик, список NAT должен быть привязан к интерфейсу.

10.3.1 Привязка к интерфейсу

Для того, чтобы привязать список к интерфейсу, нужно войти в режим конфигурации интерфейса и выполнить команду ip nat-group <имя списка>

Следует обратить внимание, что привязка nat списка осуществляется всегда к внешнему интерфейсу! Например:

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip nat—group mynat
```

Для удаления связи с интерфейсом, необходимо выполнить команду: no ip nat-group <имя>, например:

```
DionisNX(config—if—ethernet0)# no ip nat—group mynat
DionisNX(config—if—ethernet0)# do show
```

Иногда возникает необходимость делать nat для трафика, который уходит в туннель DISEC/IPSEC. В этом случае, преобразование адресов должно выполняться до шифрования перед отправкой на интерфейс (SNAT) и после расшифрования, после приема на интерфейсе (DNAT). В этом случае, привязка списков осуществляется командой: ip nat-group-xfrm, синтаксис которой аналогичен ip nat-group. Удаление связи осуществляется командой: no ip-nat-group-xfrm.

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip nat—group—xfrm mynat
DionisNX(config—if—ethernet0)# no ip nat—group—xfrm mynat
DionisNX(config—if—ethernet0)# do show
```

10.4 Просмотр и удаление активных соединений

Все проходящие через DionisNX соединения отслеживаются маршрутизатором и доступны для просмотра администратором. Для этого необходимо выполнить команду: show ip connections [параметры] из enable режима (или из режима configure, но с префиксом do).

Существует возможность просмотра соединений, над которыми выполняются NAT-преобразования, например:

```
DionisNX(config)# do show ip connections dnat tcp
```

В результате будут показаны TCP-соединения, над которыми выполнены преобразования DNAT.

При изменении параметров преобразований nat может оказаться необходимым очистить часть активных соединений, над которыми выполняются еще старые преобразования, для этого можно воспользоваться командой: clear ip connections [параметры] из enable режима, например:

```
DionisNX(config)# do clear ip connections dnat
```

Будут очищены все DNAT-соединения.

11. Helper

Некоторые протоколы используют несколько соединений в одной сессии (например, FTP). Для правильной обработки таких протоколов используются специальные модули (helper), которые анализируют трафик на уровне протокола и извлекают информацию о новых соединениях, что позволяет обеспечить их корректную фильтрацию, маршрутизацию и пропускание через NAT.

При этом, в ядре ОС создаются специальные записи (expectation) для ожидаемых будущих соединений.

```
# show ip connections expect
```

Так, например, если создаётся FTP-соединение в активном режиме из внешней сети во внутреннюю, то FTP-сервер создаст соединение из внутренней сети во внешнюю. Чтобы это соединение корректно обработалось подсистемой NAT, необходимо создать helper. Тогда в процессе согласования параметров через управляющее соединение NAT-подсистема сможет настроить трансляцию адресов для соединения, через которое передаются команды.

Кроме того, helper'ы позволяют фильтровать пакеты новых соединений, по признаку принадлежности их ранее открытой сессии протокола. Такие пакеты можно распознать в списках доступа по правилу отбора state related. Таким образом, даже если запрещены входящие соединения в некоторый сегмент сети, можно использовать некоторые протоколы, требующие таких соединений.

Во многом команды управления helper-list'ами похожи на команды управления access-list'ами.

11.1 Создание helper-list

Для создания helper-list'ов используется команда ip helper-list. Для добавления в неё правил используется команда helper. Пример:

```
ip helper—list ftp—irc—helper
 1 helper ftp tcp dst any dport 21
 2 helper irc tcp dst any dport 194
```

Аналогично при помощи команды ip6 helper-list создаются helper'ы для IPv6.

11.2 Применение helper-list

Для применения helper-list'ов используются команды ip helper-group и ip6 helper-group.

12. Журналирование и отладка

В Dionis-NX существует несколько механизмов, которые могут быть использованы администратором для диагностики проблем, а также для выявления нарушений.

12.1 tcpdump

Существует возможность мониторинга активности сети с помощью прослушивания сегмента Ethernet с выбранного интерфейса. Эти данные, в том или ином виде, в зависимости от указанных параметров выводятся на консоль. Для этого необходимо воспользоваться командой tcpdump в режиме enable.

У tcpdump существует множество параметров, с помощью которых можно выбирать формат вывода данных. Среди основных параметров можно выделить следующие:

Название параметра Описание	
proto <имя или числовое значение>	Задать интересуемый IP-протокол
тип интерфейса и его номер	Слушать на выбранном интерфейсе
dump <режим>	Вывод содержимого пакетов в заданном формате
host <IP-адрес или имя хоста>	Показ трафика относящегося к заданному хосту
src <IP-адрес или имя хоста>	Показ трафика с заданным исходным адресом
dst <IP-адрес или имя хоста>	Показ трафика с заданным целевым адресом
net <IP-адрес с маской>	Показ трафика относящегося к заданной сети
snet <IP-адрес с маской>	Показ трафика с адресом источника из заданной сети
dnet <IP-адрес с маской>	Показ трафика с адресом назначения из заданной сети
port <номер порта>	Показ трафика, относящемуся к заданному порту
sport <номер порта>	Показ трафика с заданным портом источника
dport <номер порта>	Показ трафика с заданным портом назначения
numeric	Не делать DNS-запросов и показывать адреса в числовом виде
ether	Выводить информацию по ethernet-заголовкам
count <число>	Закончить мониторинг после принятия <числа> пакетов, удовлетворяющих правилам выборки
write	Запись дампа в формате pcap (например для Wireshark)
export	Запись дампа в файл

Например:

```
DionisNX# tcpdump ethernet 0 numeric dump hex—ascii
```

Кроме того понимаются выражения со следующими служебными словами: **port**, **portrange**, **host** и **and**, **or**, **not**.

Например:

```
DionisNX# tcpdump ethernet 0 numeric dump hex—ascii not port 23
```

Для прерывания режима мониторинга необходимо нажать на клавиатуре Ctrl-C.

12.2 Трассировка

Механизм трассировки применяется для выявления проблем и ошибок администрирования, и позволяет проследить прохождение сетевого пакета внутри маршрутизатора.

Существует два режима работы трассировки:

1. Отладка фильтров;
2. Полная трассировка.

В режиме отладки фильтров протоколируется прохождение пакета по действующим фильтрам (входным и выходным). В режиме полной трассировки протоколируется весь путь прохождения пакета. Для задания режима трассировки необходимо перейти в режим конфигурации сервиса журналов (service log), для этого в режиме configure следует выполнить команду:

```
DionisNX(config)# service log
DionisNX(config—service—log)#
```

Режим работы трассировки задается с помощью команды trace. Так, например, для включения режима полной трассировки используется команда: trace all:

```
DionisNX(config—service—log)# trace all
DionisNX(config—service—log)# do show
log
trace all
size 262144 131072
```

Для отладки фильтров следует выполнить команду trace без параметров:

```
DionisNX(config—service—log)# trace
DionisNX(config—service—log)# do show
log
trace
size 262144 131072
```

Существует возможность отключения механизма трассировки полностью, для этого используется команда no trace:

```
DionisNX(config—service—log)# no trace
DionisNX(config—service—log)# do show
log
size 262144 131072
```

Для указания того, какой трафик должен быть подвержен трассировке, используются списки трассировки (ip trace-list и ip6 trace-list для протоколов IPv4 и IPv6 соответственно).

12.2.1 Создание списка трассировки

Для создания списка трассировки IPv4, в режиме configure необходимо выполнить команду ip trace-list <имя>, где <имя> – это имя создаваемого списка, например:

```
DionisNX(config)# ip trace—list ping
```

После выполнения команды, задаются (или модифицируются) правила трассировки списка. Каждое правило содержит критерии отбора трафика для трассировки. Если ни одно из правил не удовлетворяет критериям, трафик не будет трассироваться.

```
DionisNX(config—trace—ping)# trace icmp
```

В данном примере трассировке будет подвержен протокол icmp.

В качестве критериев отбора используется подмножество критериев списков контроля доступа (см. ip access-list).

Для того, чтобы просмотреть правила отбора текущего редактируемого списка, достаточно выполнить команду:

```
DionisNX(config—trace—ping)# do show
```

При этом будут выведены все правила текущего списка. Каждая строка снабжена числовым префиксом, указывающим позицию правила в списке.

Для того, чтобы удалить правило с конкретным номером, следует ввести команду: no <номер правила> Например:

```
DionisNX(config—trace—ping)# no 1
```

Удаление всего содержимого текущего списка может быть осуществлено командой: no all. Для удаления списка, используется команда: no ip trace-list <имя списка>.

Для просмотра информации о списках, существует две команды, доступные из enable-режима.

show ip trace-list <имя|*> config (Информация о действующей конфигурации) show ip trace-list <имя|*> (Низкоуровневая информация из ядра ОС) Если в командах имя списка задано как *, будет показана информация о всех списках.

Например (из режима configure):

```
DionisNX(config)# do show ip trace—list ping config
```

Для работы со списками трассировки IPv6, используйте команды с префиксом ip6: ip6 trace-list, no ip6 trace-list и do show ip6 trace-list.

12.2.2 Применение списка трассировки

Для того, чтобы трассировка начала действовать, необходимо применить какой-то из списков трассировки. Применение списка осуществляется командой ip trace-group <имя списка> из режима configure для IPv4 и ip6 trace-group <имя списка> для IPv6.

Например:

```
DionisNX(config)# ip trace—group ping
```

Для отмена действия списка трассировки следует выполнить команду no ip trace-group <имя списка> (no ip6 trace-group <имя списка>):

```
DionisNX(config)# no ip trace—group ping
```

12.2.3 Выборка из журнала IP-пакетов

Если механизм трассировки включен, и в действующее правило отбора для трассировки попали пакеты, то содержимое пакетов и информация об их прохождении попадает в журнал IP-пакетов.

Для просмотра и выборки журнала используется команда show ip log [дополнительные параметры] из режима enable. Например:

```
DionisNX# show ip log
2011—12—08 16:50:46.749509 @in(ethernet0) 'TRACE: PREROUTING:rule:1' IP (tos 0x60, ttl 59, id
      24044, offset 0, flags [none], proto ICMP (1), length 84)
    www.yandex.ru > 83.220.32.68: ICMP echo reply, id 7530, seq 1, length 64
2011—12—08 16:50:46.749517 @in(ethernet0) 'TRACE: outside:rule:4' IP (tos 0x60, ttl 59, id 24044,
      offset 0, flags [none], proto ICMP (1), length 84)
    www.yandex.ru > 83.220.32.68: ICMP echo reply, id 7530, seq 1, length 64
2011—12—08 16:50:46.749533 @fwd(ethernet0—>ethernet2) 'TRACE: FORWARD:policy:1' IP (tos 0x60,
      ttl 58, id 24044, offset 0, flags [none], proto ICMP (1), length 84)
    www.yandex.ru > 192.168.33.41: ICMP echo reply, id 7530, seq 1, length 64
2011—12—08 16:50:46.749541 @out(ethernet2) 'TRACE: POSTROUTING:policy:1' IP (tos 0x60, ttl 58, id
      24044, offset 0, flags [none], proto ICMP (1), length 84)
```

Запись в журнале содержит в себе: время, цепочку обработки, интерфейс, попадание в фильтры и информацию о пакете. Цепочка обработки может принимать значения, приведенные в таблице:

название	описание
in	вход в маршрутизатор
out	выход из маршрутизатора
fwd	логика маршрутизации
local_in	пакет предназначен для локального процесса
local_out	пакет порождается локальным процессом

К команде show ip log могут передаваться параметры, приведенные в таблице:

название	описание
число записей	показать последние n записей
all	показать все записи
proto <протокол>	выборка заданного IP-протокола
src <маска источника>	выборка для заданного источника
dst <маска приемника>	выборка для заданного приемника

название	описание
in <тип интерфейса> <номер интерфейса>	выборка для пакетов, входящих в заданный интерфейс
out <ти интерфейса> <номер интерфейса>	выборка для пакетов, выходящих из заданного интерфейса
stat	вывод статистики
follow	режим непрерывного показа (слежение)
numeric	не разрешать адреса по DNS
quiet	кратко
dump <режим>	режим вывода содержимого пакета
date <дата или диапазон>	выборка по дате в формате T1 (конкретное время) или T1-T2(диапазон), где T1 или T2 записываются в формате: yy[mm[dd[hh[mm[ss]]]]]
file <файл>	выбор файла, из которого читать записи
export <файл>	сохранить вывод журнала в файл

12.3 Протоколирование правил фильтрации

Существует возможность протоколирования выбранных правил фильтрации. Для этого, в критериях отбора указывается параметр: log [all] [alert].

Например:

```
DionisNX(config)# ip access—list noicmp
DionisNX(config—acl—noicmp)# deny icmp log all
```

Присутствие ключевого слова all означает, что все данные пакета попадут в журнал. По умолчанию в журнал попадет только информация о заголовке пакета.

Присутствие ключевого слова alert означает повышенную важность сообщения.

Для произвольной выборки информации из журнала IP-пакетов используется команда: show ip log, которая описана в разделе «Трассировка».

12.4 Системные журналы

Кроме журнала IP-пакетов, в Dionis-NX поддерживается набор различных системных журналов. Для выборки данных журналов применяется команда: show log [имя журнала] [параметры] в режиме enable.

В качестве параметров могут присутствовать:

название	описание
число записей	показать последние n записей
all	показать все записи
follow	режим непрерывного показа (слежение)
archive <номер>	смотреть записи из архивных (старых)журналов

Для администратора доступны следующие журналы:

название	назначение
messages (журнал по умолчанию)	общесистемный журнал
system	системные события, относящиеся к конфигурации
dish	действия администратора
daemon	сообщения сервисов
kernel	сообщения ядра
router	сообщения от сервисов динамической маршрутизации
alert	сообщения, требующие внимания
auth	сообщения безопасности
monitor	сообщения о запусках и остановках сервисов

Следует отметить, что в журнале действий администратора dish, кроме факта выполнения команды, времени выполнения и имени учётной записи администратора, присутствует статус выполнения операции. Статус выводится в виде цифры в конце строки после двоеточия. "0" – означает отсутствие ошибки (успешно). Например:

```
Apr 11 11:50:58 RAUL klish[31906]: 0(adm) enable : 0
```

Кроме этих системных журналов, существуют журналы для подсистем dhcp, dns, ntp, snmp и другие. Просмотр журнала для них возможен с помощью команды show service <название сервиса> log [параметры] в enable режиме:

```
DionisNX# show service dns log
```

12.5 Сигнал тревоги

Для оперативного информирования администратора о возникновении в системе важных событий предусмотрен сигнал тревоги (alert). Администратору следует отреагировать на сигнал тревоги. До этого момента сигнал тревоги снят не будет. Сигнал тревоги может проявляться следующим образом:

- Красный цвет лампочки на LCD-панели;
- Знак «!» вместо «#» в строке приглашения командной оболочки dish в привилегированном режиме;
- Звуковой сигнал от встроенного динамика;
- Отправка сообщений по протоколу syslog на удаленные syslog-сервера.

Важными событиями считаются все системные события (см. раздел 12.4), уровень важности которых не ниже «критического». Такие события дополнительно попадают в системный журнал «alert». Для просмотра перечня важных событий из привилегированного режима (режим enable) необходимо выполнить следующую команду:

```
Router! show log alert
```

При этом сигнал тревоги будет снят. Также можно снять сигнал тревоги с помощью команды привилегированного режима:

```
Router! clear alert
```

При снятии сигнала тревоги системный журнал «alert» не очищается, поэтому администратор всегда может просмотреть важные события, происходившие в системе ранее.

12.5.1 Настройка звукового сигнала

Администратор имеет возможность настраивать возникновение звукового сигнала. Для включения звукового сигнала при возникновении сигнала тревоги в режиме конфигурации необходимо выполнить следующие команды:

```
Router(config)# service log
Router(config—service—log)# alert beep
```

Для выключения звукового сигнала при возникновении сигнала тревоги в режиме конфигурации необходимо выполнить следующие команды:

```
Router(config)# service log
Router(config—service—log)# no alert beep
```

При заводской установке системы звуковой сигнал по умолчанию включен.

12.5.2 Настройка удаленных оповещений

Сообщения о событиях в системе могут пересылаться на удаленные сервера. Для этого используется сетевой протокол syslog. На удаленном сервере для приема сообщений должен быть установлен и настроен syslog-сервер.

Администратор имеет возможность указать удаленные узлы, на которые будут отсылаться оповещения, и правила отбора сообщений для посылки:

```
Router(config)# service log
Router(config—service—log)# remote 192.168.1.2 *.crit
Router(config—service—log)# remote myhost.org dish.*
```

Приведенные выше команды добавят в список узлов два сервера: 192.168.1.2 и myhost.org. На первый из них будут отсылаться сообщения с приоритетом не меньшим, чем критический (что соответствует сигналу тревоги - alert). На второй сервер будут отсылаться все сообщения службы (facility) dish, что соответствует регистрации всех команд, введенных администратором в командной оболочке.

Для удаления узла из списка рассылки сообщений необходимо выполнить следующие команды в режиме конфигурации:

```
Router(config)# service log
Router(config—service—log)# no remote myhost.org dish.*
```

Правила отбора сообщений для отсылки на удаленный сервер соответствует формату syslog: »[служба].[приоритет]». Список допустимых служб:

- auth;
- authpriv;
- daemon;
- kern;
- dish - соответствует local0 в конфигурационном файле syslog;
- router - соответствует local1 в конфигурационном файле syslog;
- dhcp - соответствует local2 в конфигурационном файле syslog;
- dns - соответствует local3 в конфигурационном файле syslog;
- watcher - соответствует local7 в конфигурационном файле syslog;
- * - любая служба.

Список допустимых приоритетов:

- info;
- notice;
- warning;
- err;
- crit;
- alert;
- emerg;
- * - любой приоритет;
- none.

Также допустимы правила с использованием »,», »;», »!», »=». Более подробную информацию по формату правил отбора и использованию специальных символов можно найти в документации по стандартному сервису syslog.

12.5.3 Настройка оптимизации производительности системы

При настройке логирования пакетов (см. раздел Фильтрация и модификация) по умолчанию, гарантируется запись всех событий в журнал, вследствие чего происходит сброс пакетов, которые не успевают туда записаться.

Это поведение можно изменить командой:

```
Router(config)# service log
Router(config—service—log)# optimize throughput
```

При наличии данной настройки гарантируется пропускная способность, при этом не все пакеты записываются в журнал, в kernel.log для информирования администратора дополнительно записываются сообщения “packet NOT LOGGED!!!”.

Отключить опцию оптимизации можно командой:

```
Router(config)# service log
Router(config—service—log)# no optimize
```

13. AAA - аутентификация и авторизация с помощью протоколов семейства AAA.

13.1 Введение

AAA (Authentication, Authorization, Accounting) — протокол, используемый для описания процесса предоставления доступа и контроля за ним.

DionisNX поддерживает аутентификацию и авторизацию с помощью удаленного сервера по протоколам RADIUS и TACACS+ (протоколы семейства AAA). Это означает, что управление учетными записями пользователей возможно производить централизованно на удаленном сервере, а не на каждой машине в отдельности. В данном случае DionisNX является AAA-клиентом (radius или tacacs+). AAA клиенты также называются серверами доступа NAS (Network Access Server)

Таким образом, когда пользователь хочет подключиться к DionisNX, вначале проверяется, существует ли пользователь на устройстве NAS и не является ли он AAA-пользователем. Если пользователь существует и не является AAA-пользователем, то аутентификация и авторизация происходит по обычной схеме. Если на устройстве NAS данный пользователь отсутствует или пользователь является AAA-пользователем, то происходит подключение к AAA-серверу, где проверяются данные подключаемого пользователя и принимается решение о предоставлении доступа к системе. В случае положительного результата на устройстве NAS создается новый AAA-пользователь или обновляются полномочия существующего.

13.2 Настройка для подключения к RADIUS серверу

Настройка подключения к RADIUS серверу происходит из режима **configure**:

```
adm@DionisNX(config)# radius

adm@DionisNX(config—radius)# server 10.10.10.1 testing123
adm@DionisNX(config—radius)# enable
```

Здесь указывается ip адрес RADIUS-сервера и пароль для подключения с нему. Дополнительно можно указать порт RADIUS-сервера. По умолчанию подключение происходит к порту 1812.

```
adm@DionisNX(config—radius)# port 1813
```

13.2.1 Настройка RADIUS-сервера

Для того, чтобы RADIUS сервер мог принимать и обслуживать запросы от DionisNX необходимо подключить словарь, содержащий специфичные для DionisNX атрибуты. Словарь минимальный должен содержать следующее:

```
VENDOR          Factor—TS          1156
BEGIN—VENDOR    Factor—TS
ATTRIBUTE       Nx—Usr—Realname      1 string
ATTRIBUTE       Nx—Usr—Status        2 string
ATTRIBUTE       Nx—Usr—Description   3 string
ATTRIBUTE       Nx—Role—Deps         4 string
END—VENDOR      Factor—TS
```

Пример конфигурирования пользователя u1 на RADIUS-сервере. В качестве примера используется фрагмент конфигурации сервера FreeRADIUS (http://freeradius.org):

```
u1      Cleartext—Password      := ”555”
        Nx—Usr—Realname          := ”user1”,
        Nx—Usr—Status           := ”supervisor”,
        Nx—Usr—Description       := ”Default administrator USER1”,
        Nx—Role—Deps            := ”@default”
```

Где:

Cleartext-Password - пароль пользователя;

Nx-Usr-Realname - Настройка реального имени владельца учетной записи. Соответствует команде realname (смотри раздел “Учетные записи” данного руководства);

Nx-Usr-Status - Установка статуса супервизора (смотри раздел “Учетные записи” данного руководства);

Nx-Usr-Description - Настройка описания учетной записи пользователя. Соответствует команде description (смотри раздел “Учетные записи” данного руководства);

Nx-Role-Deps - Настройка полномочий пользователя. Соответствует команде delegate (смотри раздел “Учетные записи” данного руководства).

13.3 Настройка для подключения к TACACS+ серверу

Настройка подключения к TACACS+ серверу происходит из режима **configure**:

```
adm@DionisNX(config)# tacacs

adm@DionisNX(config—tacacs)# server 10.10.10.1 testing123
adm@DionisNX(config—tacacs)# enable
```

Здесь указывается ip адрес TACACS+ сервера и пароль для подключения с нему. Дополнительно можно указать порт TACACS+ сервера. По умолчанию подключение происходит к порту 49.

```
adm@DionisNX(config—tacacs)# port 4949
```

13.3.1 Настройка TACACS+ сервера

Пример конфигурирования пользователя utac на TACACS+ сервере. В качестве примера используется фрагмент конфигурации сервера tac_plus, работа с другими серверами tacacs не гарантированна (http://www.pro-bono-publico.de/projects):

```
host = 10.10.10.10 {
    address = 10.10.10.10
    prompt = ”Welcome\n”
    key = nas_pass_tacacs
}

group = adm {
    service = login {
        set Nx—Usr—Realname = ”user”
        set Nx—Usr—Status = ”supervisor”
        set Nx—Usr—Description = ”Tacplus user”
        set Nx—Role—Deps = ”@default”
    }
}

user = utac {
   member = adm
   password = clear utac_pass
}
```

Где:

key - пароль NAS;

address - ip адрес NAS;

Nx-Usr-Realname - Настройка реального имени владельца учетной записи. Соответствует команде realname (смотри раздел “Учетные записи” данного руководства);

Nx-Usr-Status - Установка статуса супервизора (смотри раздел “Учетные записи” данного руководства);

Nx-Usr-Description - Настройка описания учетной записи пользователя. Соответствует команде description (смотри раздел “Учетные записи” данного руководства);

Nx-Role-Deps - Настройка полномочий пользователя. Соответствует команде delegate (смотри раздел “Учетные записи” данного руководства).

password - пароль пользователя.

14. FTP-сервер

14.1 Введение

В DionisNX реализована поддержка FTP-сервера. FTP-сервер поддерживает как анонимный так и приватный доступ.

14.2 Настройка анонимного FTP-сервера

Настройка анонимного FTP-сервера происходит из режима **configure**.

```
adm@DionisNX(config)# service ftp
```

Для включения анонимного доступа необходимо выполнить команды:

```
adm@DionisNX(config—service—ftp)# allow anonymous
adm@DionisNX(config—service—ftp)# enable
```

Данная команда позволит подключаться к FTP-серверу без пароля, используя логин **ftp**.

В этом режиме пользователю будут доступны для чтения директории **incoming** и **public**. Чтобы разрешить запись в директорию **incoming** необходимо дополнить команду опцией **write**:

```
adm@DionisNX(config—service—ftp)# allow anonymous write
```

Чтобы разрешить перезапись существующих файлов необходимо выполнить команду:

```
adm@DionisNX(config—service—ftp)# user anonymous overwrite
```

Кроме того можно ограничить размер дискового пространства, доступного для записи анонимному пользователю. Для этого необходимо дополнить следующую команду:

```
adm@DionisNX(config—service—ftp)# user anonymous space 1mb
```

В данном примере размер анонимному пользователю доступен 1 МБ.

14.3 Настройка приватного FTP-сервера

14.3.1 Настройка учетных записей

Настройка учетных записей происходит из **enable** режима.

Добавление учетной записи:

```
adm@DionisNX# service ftp account create <NAME> <PASSWORD>
```

Удаление учетной записи:

```
adm@DionisNX# service ftp account remove <NAME>
```

Изменение пароля учетной записи:

```
adm@DionisNX# service ftp account password <NAME> <NEWPASSWORD>
```

Просмотр списка учетных записией:

```
adm@DionisNX# show service ftp account
```

Просмотр хеша пароля учетной записии:

```
adm@DionisNX# show service ftp account <NAME> passwd—hash
```

14.3.2 Включение учетных записей

Настройка приватного FTP-сервера происходит из режима **configure**.

```
adm@DionisNX(config)# service ftp
```

Для включения приватного доступа необходимо выполнить команды:

```
adm@DionisNX(config—service—ftp)# allow adm
adm@DionisNX(config—service—ftp)# enable
```

В данном режиме пользователю с логином *adm* запрещена запись с свою домашнюю директорию на сервере. Чтобы разрешить запись необходимо дополнить команду опцией **write**:

```
adm@DionisNX(config—service—ftp)# allow adm write
```

Чтобы разрешить перезапись существующих файлов необходимо выполнить команду:

```
adm@DionisNX(config—service—ftp)# user adm overwrite
```

Кроме того есть возможность ограничить дисковое пространство для записи. В примере ниже пользователю *adm* выделено 10MB:

```
adm@DionisNX(config—service—ftp)# user adm space 10mb
```

14.4 Дополнительные команды конфигурации FTP-сервера

По умолчанию сервер использует порт 21 и слушает на всех доступных адресах. Можно изменить порт по умолчанию и указать конкретные адреса для прослушивания. Например:

```
adm@DionisNX(config—service—ftp)# port 33
adm@DionisNX(config—service—ftp)# listen 192.168.33.155 21
adm@DionisNX(config—service—ftp)# listen 192.168.40.155
```

Для более детального разграничения доступа с FTP-серверу можно узазать IP адреса разрешенных/запрешенных узлов и/или сетей. Рассмотрим пример:

```
adm@DionisNX(config—service—ftp)# allow anonymous@192.168.33.84/32 write
adm@DionisNX(config—service—ftp)# allow adm
adm@DionisNX(config—service—ftp)# deny adm@192.168.33.0/24
```

В данном примере анонимный доступ разрешен только для узла с адресом *192.168.33.84* (кроме того разрешена запись для данного узла), а доступ для пользователя *adm* разрешен только для чтения из любых сетей кроме *192.168.33.0/24*.

По умолчанию в журнал работы сервиса заносятся только основные действия, такие как запуск и остановка FTP-сервера. Для ведения более подробного журнала необходимо выполнить команду:

```
adm@DionisNX(config—service—ftp)# log <'none'|'low'|'middle'|'high'>
```

где:

- **none** - соответствует поведению по умолчанию;
- **low** - дополнительно записываются события аутентификации и завершения сессии (ACCT,PASS,REIN,USER,EXIT);
- **middle** - вместе с событиями уровня **low** журналируются события следующего типа: FEAT, HELP, MDTM, QUIT, PWD, STAT, SIZE, SYST, XPWD, CDUP, CWD, LIST, MKD, MLSD, MLST, NLST, RMD, XCWD, XCUP, XMKD, XRMD, RETR, APPE, MFF, MFMT, MKD, RMD, RNFR, RNTO, STOR, STOU, XMKD, XRMD;
- **high** - все события уровня **middle**, а также события ABOR, ALLO, EPRT, EPSV, MODE, NOOP, OPTS, PASV, PORT, REST, RNFR, RNTO, SITE, SMNT, STRU, TYPE, AUTH, CCC, PBSZ и PROT. Кроме того записывается информация о передачи файлов. Пример такой записи:

  ```
  Sun Nov 10 05:42:21 2019 0 DionisNX.cuba.int 16971 /home/ftp/.files/adm/qwe144 b _ i r adm
          ftp 0 * c
  ```

  где:

  - *Sun Nov 10 05:42:21 2019* - дата;
  - *0* - длительность операции в секундах;
  - *DionisNX.cuba.int* - адрес инициализации трансфера;
  - *16971* - размер файла;
  - */home/ftp/.files/adm/qwe144* - пункт назначения;
  - *b* - тип трансфера (a: в ASCII-типе; b: в бинарном типе);
  - *** - флаг сжатия (C: файл был сжат; U: файл был рассжат; T: файл был затарен;* : никаких действий не произведено);
  - *i* - направление (i: входящий; o: исзодящий; d: удаление);
  - *r* - тип доступа (r: неанонимный; a: анонимный);
  - *adm* - имя пользователя;
  - *ftp* - имя сервиса;
  - *0* - аутентификация (0: обычная; 1: RFC931);
  - * - id пользователя (недоступно - т.к. все пользователи виртуальные);
  - *c* - статус опарации (c: выполнено; i: прервано).

15. VLAN

Dionis-NX поддерживает стандарт IEEE 802.1Q (VLAN). Для создания интерфейса, выходящий трафик с которого будет помечаться идентификатором vlan-сети, а входящий трафик соответствующей vlan-сети, перенаправляться на вход этого интерфейса, используется команда: interface ethernet <номер интерфейса>.<номер vlan> (режим configure).

Например:

```
DionisNX(config)# interface ethernet 0.1
DionisNX(config—if—ethernet0.1)#
```

В дальнейшем интерфейс настраивается так же, как и любой другой ethernet-интерфейс. Однако следует иметь в виду, что для активизации vlan-интерфейса необходимо, чтобы и соответствующий обычный интерфейс был активирован.

16. IEEE 802.1ad (QinQ)

Dionis-NX поддерживает стандарт IEEE 802.1ad (QinQ - VLAN в VLAN).

Для создания QinQ интерфейса сначала нужно создать родительский 802.1ad интерфейс:

```
DionisNX(config)# interface ethernet <номер интерфейса>.<номер vlan>ad
```

Затем, чтобы создать QinQ интерфейс, используется следующая команда:

```
DionisNX(config)# interface ethernet <номер интерфейса>.<номер vlan>ad.<номер>
```

При этом, выходящий с этого интерфейса трафик будет помечаться идентификатором vlan-сети и перенаправляться в родительский 802.1ad интерфейс, а входящий трафик с 802.1ad интерфейса, соответствующей VLAN-сети, перенаправляться на вход QinQ интерфейса,

В дальнейшем, QinQ интерфейс настраивается так же, как и любой другой ethernet-интерфейс. Однако следует иметь в виду, что для активизации QinQ-интерфейса необходимо, чтобы соответствующие родительский 802.1ad и ethernet интерфейс были созданы и активированы.

Например (при условии, что VLAN ethernet 0.1ad был создан и активирован ранее):

```
DionisNX(config)# interface ethernet 0.1ad.2
DionisNX(config—if—ethernet0.1ad.2)# ip address 10.0.0.1/24
DionisNX(config—if—ethernet0.1ad.2)# enable
```

17. VXLAN

VXLAN является технологией сетевой виртуализации, созданной для решения проблем масштабируемости в больших системах. Она использует схожую с VLAN технику для MAC инкапсуляции Layer 2 Ethernet-кадров в UDP-пакеты, порт 4789.

Для создания интерфейса vxlan для работы с IPv4 используйте команду: interface vxlan <номер интерфейса>

```
DionisNX(config)# interface vxlan 0
DionisNX(config—if—vxlan0)#
```

Для создания IPv6 vxlan интерфейса используйте команду: interface ip6vxlan <номер интерфейса>.

Интерфейс vxlan в целом настраивается так же, как и интерфейсы других типов, однако имеется набор параметров, которые применимы только для vxlan интерфейсов. Эти параметры приведены в таблице:

команда	параметр
id <ID>	Идентификатор (обязательный параметр)
local <IP-адрес>	IP-адрес локального конца туннеля (unicast режим)
remote <IP-адрес>	IP-адрес удаленного конца туннеля (unicast режим)
ttl <значение>	Принудительно устанавливать значение ttl для UDP-датаграммы, а не наследовать его из инкапсулируемого пакета
tos <тип_трафика>|inherit	Настройка типа трафика (по качеству обслуживания)
group <мультикаст группа>	Задать группу мультикаст (multicast режим)
srcport <порт источник>	Явное задание UDP порта источника
dstport <порт назначения>	Явное задание UDP порта назначения

18. Bonding-интерфейсы

Dionis-NX поддерживает агрегацию (bonding) интерфейсов путем объединения нескольких физических интерфейсов в один логический (мастер), что можно использовать для повышения пропускной способности или в целях резервирования.

Создание/редактирование мастер-интерфейса осуществляется с помощью команды: interface bond <номер> в режиме configure.

После создания bonding-интерфейса, управление им осуществляется в целом так же, как и любым другим интерфейсом, за исключением следующих особенностей:

1. С помощью команды slave должны быть указаны подчиненные интерфейсы (или один интерфейс), которые будут использоваться для агрегации;
2. Подчиненные интерфейсы не должны использоваться маршрутизатором (кроме как в агрегации) и должны находиться в отключенном состоянии;
3. С помощью команды mode должен быть выбран режим агрегации (если режим не задан – используется режим по умолчанию);
4. С помощью команды monitor желательно задать режим мониторинга состояния подчиненных интерфейсов;

18.1 Режимы агрегации

Существуют следующие режимы агрегации:

режим	описание
balance-rr	режим по умолчанию, циклическое использование подчиненных интерфейсов
active-backup	режим резервирования
balance-xor	распределение зависит от хеш-функции
broadcast	одновременная передача по всем интерфейсам
802.3ad	IEEE 802.3ad
balance-tlb	адаптивная балансировка передачи
balance-alb	адаптивная балансировка (в том числе и на приеме)

18.1.1 balance-rr

Этот режим используется по умолчанию, если в настройках не указано другое. balance-rr обеспечивает балансировку нагрузки и отказоустойчивость. В данном режиме пакеты отправляются «по кругу» от первого интерфейса к последнему и сначала. В случае выхода из строя одного из интерфейсов, входящего в Bond, пакеты отправляются на оставшиеся в работе. При восстановлении работоспособности отказавшего интерфейса, он продолжает использоваться для передачи трафика в составе Bond интерфейса. При подключении портов к разным коммутаторам необходимо выполнить их настройку.

18.1.2 active-backup

Работает только один интерфейс, остальные находятся в очереди горячей замены. Если ведущий интерфейс перестает функционировать, то его нагрузку подхватывает следующий (присвоив mac-адрес) и становится активным. При восстановлении работоспособности отказавшего интерфейса, маршрутизатор продолжает использовать для передачи трафика интерфейс, на который трафик перешел после отказа. Восстановившийся интерфейс находится в очереди горячей замены. Дополнительная настройка коммутатора не требуется.

В данном режиме присутствует подпараметры fail-over-mac и arp-validate. Настраиваются они в процессе выбора режима балансировки с помощью команды: **adm@DionisNX(config-if-bond0)# mode active-backup fail-over-mac[none active follow] arp-validate [none active backup all]**

fail-over-mac

- None - Bond интерфейсу назначается адрес первого добавленного в bonding slave интерфейса. Для мгновенного перевода исходящего трафика на другие интерфейсы MAC адрес bond интерфейса назначается всем физическим интерфейсам, участвующим в Bonding.
- active – Bond интерфейсу назначается адрес первого добавленного в bonding slave интерфейса. В случае падения активного интерфейса, в работу включается следующий, входящий в bonding, и bond присваивает себе MAC адрес нового активного интерфейса
- follow – Bond интерфейсу назначается адрес первого добавленного в bonding slave интерфейса. Активному физическому интерфейсу назначается такой же MAC, как и bond интерфейсу. В случае падения активного интерфейса трафик переходит на следующий, входящий в bonding, и этому интерфейсу присваивается MAC адрес интерфейса bond.

Внимание: подпараметр fail-over-mac может быть изменен только в том случае, если в текущем bonding-интерфейсе нет подчиненных интерфейсов.

Arp-validate

Используется при выборе ARP мониторинга состояния интерфейса (описан далее). Команда используется для проверки входящих ARP ответов и их сопоставления с IP адресами, на которые направлялись ARP запросы. Интерфейс рассматривается работоспособным только в случае получения соответствующих ответов от опрашиваемых узлов.

- none – проверка arp отключена
- active –для активного подчиненного интерфейса проверяется, что ARP ответы приходят от узлов, указанных в команде monitoring arp.
- backup – для бэкап интерфейсов с помощью широковещательного ARP запроса проверяется, что ARP ответы приходят от узлов, указанных в команде monitoring arp. Мониторинг бэкап интерфейсов нужен для того, чтобы определить наилучшего кандидата на замену активному интерфейсу в случае его падения.
- all – проверка включена для всех интерфейсов в bonding

18.1.3 balance-xor

XOR-политика: Выбор подчиненного интерфейса выполняется на основе хеш-функции [по умолчанию: (исходный MAC-адрес ☐ XOR ☐ MAC-адрес получателя) %число интерфейсов]. Режим обеспечивает балансировку нагрузки и отказоустойчивость. При выборе данного режима балансировки можно задать режим хеширования для хеш-функции, которая используется для выбора подчиненного интерфейса. Сделать это можно во время выбора режима балансировки: **adm@DionisNX(config-if-bond0)# mode balance-xor [layer2 layer2+3 layer3+4]**

18.1.4 broadcast

Все пакеты передаются на все интерфейсы в группе. Режим обеспечивает отказоустойчивость.

Данный режим балансировки следует использовать с осторожностью – возможно увеличение нагрузки на маршрутизатор и его ответную часть пропорционально количеству участвующих в Bond интерфейсе линков. Также в случае некорректного подключения возможно дублирование ответного трафика. Данный режим можно использовать при необходимости подключения на ответной стороне нескольких устройств, которые вместе с интерфейсом Bond должны принадлежать одной подсети (например, для горячего резервирования).

18.1.5 802.3ad

IEEE 802.3ad Dynamic Link aggregation (динамическое объединение каналов). Создает агрегации групп, имеющие одни и те же скорости и дуплексные настройки. Использует все включенные интерфейсы в активном агрегаторе согласно спецификации 802.3ad.

Необходимы коммутаторы с поддержкой IEEE 802.3ad Dynamic Link aggregation. Большинство параметров потребует некоторой конфигурации для режима 802.3ad.

Изменить логику выбора активных интерфейсов и частоту передачи LACPDU для изменения времени реакции протокола LACP можно с помощью команды: **Mode 802.3ad [ad-select] [lacp-rate] Ad-select** – при наличии нескольких каналов на разные (и только на разные) коммутаторы позволяет использовать различную логику выбора активного канала - stable – выбор осуществляется по наибольшей полосе пропускания - bandwidth – то же самое, в случае добавления новой пропускной способности в один из каналов происходят перевыборы активного объединенного канала - count – выбор осуществляется исходя из наибольшего количества объединенных портов **Lacp-rate** – позволяет выбрать частоту, с которой будут посылаться LACPDU сообщения. - Fast – каждую секунду - Slow – каждые 30 секунд (по умолчанию)

При выборе данного режима балансировки можно задать режим хеширования для хеш-функции, которая используется для выбора подчиненного интерфейса. Сделать это можно во время выбора режима балансировки: **adm@DionisNX(config-if-bond0)# mode 802.3ad [layer2 layer2+3 layer3+4]**

18.1.6 balance-tlb

Адаптивная балансировка передаваемой нагрузки: канал связи не требует какой либо специальной настройки. Исходящий трафик распределяется в соответствии с текущей нагрузкой (вычисляется по скоростям) для каждого интерфейса. Входящий трафик принимается текущим интерфейсом, выбранным активным на данный момент. Если принимающий интерфейс выходит из строя, то следующий занимает его место, приватизировав его mac-адрес.

Данный режим не имеет понятия backup интерфейса – все интерфейсы, включенные в bonding, могут быть выбраны в качестве активных для передачи трафика. Поэтому в случае подключения физических интерфейсов, входящих в bonding, к нескольким ответным коммутаторам надо убедиться в корректном подключении соединительных кабелей и правильности настройки.

18.1.7 balance-alb

Адаптивное перераспределение нагрузки: включает balance-tlb плюс receive load balancing (rlb) для трафика IPv4 и не требует специального конфигурирования. То есть все так же как и при режиме balance-tlb, только дополнительно и входящий трафик тоже балансируется между интерфейсами. Полученная балансировка нагрузки достигается опросом ARP. Драйвер перехватывает ответы ARP, направленные в локальной системе в поисках выхода, и перезаписывает исходный адрес сетевой карты с уникальным аппаратным адресом одного из интерфейсов в группе.

Исходящий трафик распределяется также как и в режиме TLB в соответствии с текущей нагрузкой (вычисляется по скоростям) для каждого интерфейса. В отличие от режима TLB, где трафик мог принимать только один интерфейс, в режиме ALB входящий трафик тоже балансируется между интерфейсами. Полученная балансировка нагрузки достигается опросом ARP. Драйвер анализирует ARP-запросы, направленные маршрутизатору в поисках выхода, и в ARP-ответе отправляет не MAC адрес bond интерфейса, а адрес конкретного физического интерфейса.

18.1.8 Настройка режимов хеширования

Если выбран один из режимов: 802.3ad или balance-xor, то можно задать режим хеширования для хеш-функции, которая используется для выбора подчиненного интерфейса. Доступны следующие режимы хеширования:

- layer2: исключающее «или» по MAC-адресам источника и приемника (функция по умолчанию);
- layer2+3: исключающее «или» по MAC-адресам источника и приемника, а также по IP-адресам приемника и источника;
- layer3+4: исключающее «или» по IP-адресам приемника и источника, а также портам источника и назначения. Порты источника и назначения используются при расчете для протоколов TCP и UDP только в случае нефрагментированных пакетов.

Далее более подробно рассмотрен выбор подчиненных интерфейсов для каждого режима хеширования.

Layer2

При использовании этого режима используется функция исключающее «или» по MAC-адресам источника и приемника (функция по умолчанию). Используемая формула для генерации хеша:

(source_MAC_address XOR destination_MAC_address) MODULO slave_count

Source_MAC_address – MAC адреса источника Destination_MAC_address – MAC адрес получателя Slave_count – количество интерфейсов, входящих в Bonding

Данный алгоритм будет распределять пакеты по интерфейсам для балансировки нагрузки.

Пример: Предположим, что интерфейсы в bonding добавлялись последовательно (Ethernet 0 – Ethernet 1 – Ethernet 2 – Ethernet 3). Предположим, что имеются два получателя трафика – с MAC адресами 0021CE080ED9 и 0021CE08328A. MAC адрес интерфейса Bond 0 = 00187D1E7E13. Для того, чтобы определить, на какой физический интерфейс маршрутизатор будет посылать пакеты для получателя с MAC адресом 0021CE080ED9 происходит следующее. В bonding входит 4 интерфейса, значит у маршрутизатора есть всего четыре варианта выбора – физические интерфейсы Ethernet 0 - Ethernet 3. Source_MAC_address – 00187D1E7E13 – адрес интерфейса Bond 0 Destination_MAC_address – 0021CE080ED9 – адрес получателя Slave_count – 4 (в формулу подставляем в шестнадцатеричном формате) Подставим в алгоритм и получим: (00187D1E7E13 XOR 0021CE080ED9) MODULO 4 = 2 , где XOR – исключающее побитовое ИЛИ MODULO – остаток от деления по модулю В логике маршрутизатора все интерфейсы, претендующие на роль активного для перенаправления трафика, нумеруются начиная с нулевого и далее подряд в порядке добавления их в bonding. Т.е. в данном случае для маршрутизатора Ethernet 0 – это интерфейс выбора 0, Ethernet 1 – интерфейс выбора 1 итд. Так будет для случая, когда интерфейсы добавлялись в таком порядке: Eth0-Eth1-Eth2-Eth3. Если же интерфейсы добавлялись, например, так – Eth0-Eth1-Eth3-Eth2, то Eth 0 – интерфейс выбора 0, Eth 1 – 1, Eth 3 – 2, Eth 2 – 3. Соответственно, все кадры для данной пары MAC отправителя – MAC получателя будут отправляться через интерфейс Eth 2. Для второй пары имеем:

(00187D1E7E13 XOR 0021CE08328A) MODULO 2 = 1

Для этой пары будет использоваться интерфейс E1. При появлении других MAC адресов расчеты выполняются аналогично.

Layer2+3

Этот режим балансировки использует исключающее «или» по MAC-адресам источника и приемника, а также по IP-адресам приемника и источника. Используемая формула для генерации хеша: **(((source_IP XOR dest_IP) AND 0xffff) XORb(source_MAC XOR destination_MAC)) MODULO slave_count**

Source_IP_address – IP адрес источника Dest_IP_address – IP адрес получателя Source_MAC_address – MAC адреса источника Destination_MAC_address – MAC адрес получателя Slave_count – количество интерфейсов, входящих в Bonding

Данный алгоритм будет распределять пакеты по интерфейсам для балансировки нагрузки. В случае попадания на интерфейс не IP трафика будет использоваться формула для метода балансировки Layer 2.

Пример: Предположим, что DionisNX придется передавать трафик для двух получателей - получателя 1 с MAC адресом 0021CE08329A и IP адресом 10.0.0.1 от отправителя с IP адресом 10.0.1.1

- получателя 2 с MAC адресом 0021CE080ED5 и IP адресом 192.168.3.1 от отправителя с IP адресом 192.168.2.1 MAC адрес интерфейса Bond 0 на DionisNX = 00187D1E7E13. Для того, чтобы определить, на какой физический интерфейс маршрутизатор будет посылать пакеты для получателя 1 происходит следующее. У маршрутизатора есть всего четыре выбора – физические интерфейсы Ethernet 0 - Ethernet 3. Source_IP_address – 10.0.1.1 – IP адрес отправителя Dest_IP_address – 10.0.0.1 - IP адрес получателя Source_MAC_address – 00187D1E7E13 – адрес интерфейса Bond 0 Destination_MAC_address – 0021CE08329A – адрес получателя Slave_count – 2 (в формулу подставляем в шестнадцатеричном формате) Все вычисления осуществляются маршрутизатором автоматически. Данные выкладки приведены в целях полного понимания логики выбора того или иного интерфейса для перенаправления трафика. Если имеется необходимость ручного расчета: - каждый октет IP адреса должен быть дополнен нулями до 3-х значного числа - далее точки отбрасываются и цифры воспринимаются как одно число - все операции должны производиться только между числами в одном формате т.е. десятичные с десятичными, шестнадцатеричные с шестнадцатеричными. В расчете, описанном ниже, IP адреса приведены в десятичной системе, MAC – адреса – в шестнадцатеричной. Подставим в алгоритм и получим: (((010.000.001.001 XOR 010.000.000.001) AND 0xffff) XOR (00187D1E7E13 XOR 0021CE08329A)) MODULO 4 = 1 , где XOR – исключающее побитовое ИЛИ AND – логическое И MODULO – остаток от деления по модулю

Соответственно, все кадры для получателя 1 будут отправляться через интерфейс E1. Для второй пары имеем: (((192.168.002.001 XOR 192.168.003.001) AND 0xffff) XOR (00187D1E7E13 XOR 0021CE080ED5)) MODULO 4 = 2

Для этой пары будет использоваться интерфейс E2.

Layer3+4

Этот режим балансировки использует (там, где доступно) информацию протоколов верхних уровней для генерации хеша. Это позволяет трафику, предназначенного для одного хоста, балансироваться по нескольким подчиненным интерфейсам, т.к. балансировка осуществляется по правилу, использующему исключающее «или» по IP-адресам приемника и источника и портам источника и назначения. Для нефрагментированных TCP и UDP пакетов используется следующая формула:

((source_port XOR dest_port) XOR ((source_IP XOR dest_IP) AND 0xffff) MODULO slave_count Source_port – TCP или UDP порт источника Dest_port – TCP или UDP порт получателя Source_IP_address – IP адрес источника Dest_IP_address – IP адрес получателя

Для фрагментированных TCP или UDP пакетов и всего остального трафика IP информация о портах не используется. В случае попадания на интерфейс не IP трафика будет использоваться формула для метода балансировки Layer 2.

Пример: Предположим, что DionisNX придется передавать трафик для одного получателя с IP 10.0.0.1 от одного отправителя с IP 10.1.1.1, но с разными парами UDP портов, что имитирует, например, коммуникацию одного хоста с одним и тем же сервером и работу с несколькими приложениями. Пары UDP портов следующие: - А: порт источника: 60001 порт получателя: 50001 - Б: порт источника: 6546 порт получателя: 6545

Для того, чтобы определить, на какой физический интерфейс маршрутизатор будет посылать пакеты для пары А происходит следующее. Source_port – 60001 – порт источника Dest_port – 50001 – порт получателя Source_IP_address – 10.0.1.1 – IP адрес отправителя Dest_IP_address – 10.0.0.1 - IP адрес получателя Slave_count – 4 (в формулу подставляем в шестнадцатеричном формате) Все вычисления осуществляются маршрутизатором автоматически. Данные выкладки приведены в целях полного понимания логики выбора того или иного интерфейса для перенаправления трафика. Если

имеется необходимость ручного расчета: - каждый октет IP адреса должен быть дополнен нулями до 3-х значного числа - далее точки отбрасываются и цифры воспринимаются как одно число - все операции должны производиться только между числами в одном формате т.е. десятичные с десятичными, шестнадцатеричные с шестнадцатеричными. В расчете, описанном ниже, часть данных приведена в десятичной системе, часть - в шестнадцатеричной.
Подставим в алгоритм и получим: ((60001 XOR 50001) XOR ((10.0.1.1 XOR 10.0.0.1) AND 0xffff) MODULO 4 = 0 , где XOR – исключающее побитовое ИЛИ AND – логическое И MODULO – остаток от деления по модулю

Соответственно, все кадры для получателя 1 будут отправляться через интерфейс E0. Для второй пары имеем:

((6546 XOR 6545) XOR ((10.0.1.1 XOR 10.0.0.1) AND 0xffff) MODULO 4 = 3

Для этой пары будет использоваться интерфейс E3.

18.2 Режимы мониторинга

Мониторинг состояния линка контролирует работоспособность текущего линка. Всего на данный момент существует два режима мониторинга: ARP мониторинг и MII (Media Independent Interface) мониторинг. Включить два вида мониторинга сразу нельзя.

ARP мониторинг При выборе этого вида мониторинга, на выбранные узлы сети (неважно какие, главное с возможностью ответа на ARP сообщения) посылаются ARP запросы и на основании ответов делается заключение, работоспособен ли линк. Это дает гарантию того, что трафик действительно передается между двумя узлами. ARP мониторинг может быть настроен как на проверку доступности одного узла, так и нескольких что увеличивает надежность мониторинга. Включается этот тип мониторинга командой конфигурации интерфейса: **adm@DionisNX(config-if-bond0)# monitor arp [interface IP, ...]** frequency – частота отсылки ARP запросов interface IP – IP адрес, куда будет посылаться ARP запрос

MII мониторинг Этот тип мониторинга проверяет только состояние несущей на локальном интерфейсе маршрутизатора. Включается этот тип мониторинга командой конфигурации интерфейса: **adm@DionisNX(config-if-bond0)# monitor mii [downdelay updelay]** frequency – частота проверки работоспособности интерфейса donwdelay – задает время между объявлением физического интерфейса неработоспособным до объявления slave-интерфейса неактивным updelay - задает время между объявлением физического интерфейса работоспособным до объявления slave-интерфейса активным

Для удаления подчиненных интерфейсов можно использовать команду no slave <интерфейс>:

```
DionisNX(config)# interface bond 0
DionisNX(config—if—bond0)# slave ethernet 0
DionisNX(config—if—bond0)# slave ethernet 2
DionisNX(config—if—bond0)# no slave ethernet 2
DionisNX(config—if—bond0)# slave ethernet 1
DionisNX(config—if—bond0)# monitor mii 100
DionisNX(config—if—bond0)# mode active—backup
DionisNX(config—if—bond0)# enable
```

19. Сетевые мосты

Dionis-NX может выступать в роли Ethernet-коммутатора, позволяя объединять сегменты сети с помощью сетевых мостов. Сетевой мост представлен в системе интерфейсом особого типа – bridge. Затем, в интерфейс добавляются порты, трафик с которых будет пересылаться.

Например:

```
DionisNX(config)# interface bridge 0
DionisNX(config—if—bridge0)# port ethernet 0
DionisNX(config—if—bridge0)# port ethernet 1
DionisNX(config—if—bridge0)# enable
```

В данном примере, порты ethernet 0 и etherent 1 объединяются в сетевой мост. Пакеты, приходящие в порты, передаются на основе Ethernet-адресов, а не IP-адресов (как в маршрутизаторе). Поскольку передача выполняется на канальном уровне (уровень 2 модели OSI), все протоколы более высокого уровня прозрачно проходят через мост.

Для удаления портов из моста следует использовать команду no port <интерфейс>. Существует также команда no port all – для удаления всех портов.

У сетевого моста существуют следующие параметры:

Команда	Значение
[no] ageing	Задание времени ageing - времени жизни MAC-адреса в базе данных маршрутизации в секундах
[no] fd	Задание задержки маршрутизации в секундах

Сетевой мост в Dionis-NX поддерживает протокол STP (Spanning Tree Protocol), который используется для того, чтобы избежать петель коммутации. Для настройки STP следует использовать следующие команды:

Команда	Описание
[no] stp	Включение/выключение протокола stp
[no] hello	Задание времени hello (stp)
[no] maxage	Задание времени maxage (stp)

Сетевой мост в Dionis-NX поддерживает более тонкую настройку его работы с многоадресным трафиком. В дальнейшем описании будем использовать обозначение МР для роутера осуществляющего многоадресную передачу (МП),а многоадресный трафик - МП-трафиком. Для данной настройки следует использовать следующие команды:

Команда	Описание
igmp-routing on	система сама определяет (отслеживание IGMP Query) пристуствие МР подключенных к порту сетевого моста; если МР обнаружен - разрешается МП через данный порт, иначе - МП через данный порт не будет осуществляться

Команда	Описание
igmp-routing off	порт не видит МР, подключенных к нему и МП через порты моста не будет осуществляться
no igmp-routing	по умолчанию: разрешается МП через порты моста, независимо от того, есть ли МР, подключенные к портам моста
igmp-snooping on	сетевой мост начинает анализировать все IGMP-пакеты между подключенными к нему потребителями и поставщиками МП-трафика; обнаружив запрос IGMP-Join (присоединиться к группе) потребителя, мост включает порт, к которому тот подключён; обнаружив запрос IGMP-Leave (покинуть группу), удаляет соответствующий порт из списка группы.
igmp-snooping off	сетевой мост без разбора ретранслирует МП-трафик по всем своим портам
no igmp-snooping	по умолчанию поведение соответсвует igmp-snooping on.

Для просмотра информации о сетевом мосте используйте команду show bridge [<iface>], например:

```
DionisNX# show bridge 0
```

20. Интерфейсы E1

Интерфейсы E1 предназначены для передачи голоса или данных. Система Dionis-NX использует интерфейсы E1 для передачи данных. Такой интерфейс имеет 30 (или 31) каналов по 64 кбит/сек для данных и 2 (или 1) канала для служебной информации. Общая пропускная способность интерфейса E1 - 2048 Кбит/с.

Так как интерфейс E1 может использовать сразу несколько независимых каналов передачи данных через одно физическое соединение, то физический интерфейс E1 будем называть контроллером. А интерфейсом будем называть логический канал (или объединенную группу каналов) передачи данных. Т.е. имея один физический контроллер E1, можно организовать от 1 до 31 логических каналов передачи данных, каждый из которых в системе Dionis-NX будет выглядеть как интерфейс.

Для работы с E1 в системе Dionis-NX необходимо сначала задать параметры контроллера E1, а уже затем, на основании параметров контроллера, создавать и настраивать интерфейсы.

20.1 Установка и подключение платы контроллера E1

Ниже изображена лицевая панель контроллера E1 и таблица подключения портов E1.

20.1.1 Лицевая панель контроллера E1

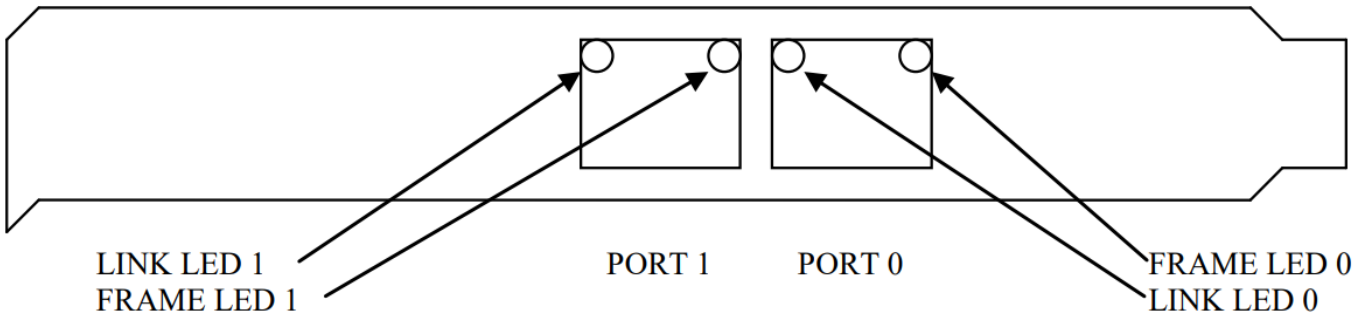

Рис. 20.1: Лицевая панель контроллера E1

LINK LED – индикатор наличия сигнала E1

FRAME LED – индикатор наличия кадровой синхронизации E1

20.1.2 Таблица подключения портов E1

Порт\Вывод	1	2	3	4	5	6	7	8
Port 0	RX 0+	RX 0-		TX0+	TX0-			
Port 1	RX 1+	RX 1-		TX1+	TX1-			

Примечание:

- RX – приёмник (вход), TX – передатчик (выход)

20.1.3 Нумерация выводов порта контроллера E1.

Тип используемого соединения RJ-45.

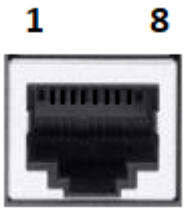

Рис. 20.2: Нумерация выводов порта контроллера E1

20.2 Настройка контроллера

Настройка контроллера E1 производится в режиме конфигурации.

```
DionisNX(config)# controller e1 1
DionisNX(config—e1—1)#
```

20.2.1 Формат кадра (фреймирование)

Поток E1 может быть фреймированным или нефреймированным. Во фреймированном режиме кадр делится на 32 временных слота, которые соответствуют каналам передачи данных. Один или два слота используются для передачи служебной информации. В нефреймированном режиме поток не делится на кадры. Тем самым достигается наиболее полное использование физического канала передачи данных, но в этом случае возможен только один логический канал передачи. Т.е. на основании такого контроллера в системе Dionis-NX возможно создать только один интерфейс.

Чтобы задать нефреймированный режим работы контроллера E1 используется следующая команда:

```
DionisNX(config—e1—1)# unframed
```

Чтобы опять включить фреймированный режим используется команда:

```
DionisNX(config—e1—1)# no unframed
```

Если выбран фреймированный режим, то необходимо также задать формат кадра. Возможные форматы кадра:

- CAS (Channel Associated Signaling) - В этом формате временной слот №16 также является служебным и его нельзя использовать для передачи данных. В русскоязычной литературе этот формат упоминается как «сигнализация по выделенному каналу»;
- CCS (Common Channel Signaling) - В этом формате для передачи данных доступен 31 временной слот. В русскоязычной литературе этот формат называется ОКС-7 (Обще-Канальная Сигнализация №7).

Чтобы задать формат кадра используются следующие команды.

Для формата CAS:

```
DionisNX(config—e1—1)# framing cas
```

Для формата CCS:

```
DionisNX(config—e1—1)# framing ccs
```

20.2.2 Кодирование сигнала

Для кодирования сигнала могут использоваться следующие методы:

- AMI (Alternate Mark Inversion)- попеременная инверсия сигнала;
- HDB3 (High Density Bipolar 3) - усовершенствованная версия AMI.

Чтобы задать кодирование сигнала, используются специальные команды.

Для кодирования AMI:

```
DionisNX(config—e1—1)# coding ami
```

Для кодирования HDB3:

```
DionisNX(config—e1—1)# coding hdb3
```

20.2.3 Детектирование ошибок

Для детектирования ошибок может использоваться режим CRC4 (Cyclic Redundancy Checking). Для включения этого режима используется следующая команда:

```
DionisNX(config—e1—1)# crc4
```

Для выключения режима CRC4 используется следующая команда:

```
DionisNX(config—e1—1)# no crc4
```

20.2.4 Источник синхронизации

С помощью команды «timing» задается приоритет источника синхронизации. Возможные значения:

- 1 - Оборудование на удаленном конце соединения имеет наивысший приоритет в качестве источника синхронизации;
- 2 - Оборудование на удаленном конце соединения имеет приоритет 2 в качестве источника синхронизации. Будет использоваться, если станет недоступен первичный источник синхронизации;
- 3-255 - Более низкие приоритеты;
- 0 - Никогда не использовать оборудование на удаленном конце соединения в качестве источника синхронизации. Это означает, что оборудование на удаленном конце соединения всегда будет подчиненным.

Задать приоритет источника синхронизации можно с помощью команды:

```
DionisNX(config—e1—1)# timing 1
```

20.2.5 Состояние контроллера E1

Текущее состояние контроллера E1 можно узнать с помощью команды «show controller».

```
DionisNX# show controller e1 1
```

20.2.6 Настройки по умолчанию

По умолчанию (если параметр не указан явно в настройках контроллера), поля принимают следующие значения:

- Фреймированный поток no unframed
- Источник синхронизации timing 0
- Формат кадра framing ccs
- Кодирование сигнала coding hdb3
- Детектирование ошибок CRC4 выключено no crc4

20.3 Настройка интерфейса

Для работы с E1 в системе Dionis-NX необходимо создать интерфейс типа HDLC (High Level data Link Control) и привязать его к настроенному контроллеру E1. Привязка осуществляется с помощью указания каналов E1, которые будут использоваться для передачи данных этого интерфейса. В зависимости от выбранного для контроллера E1 формата кадра таких каналов может быть 30 или 31. Для HDLC-интерфейса можно использовать как один такой канал, так и группу каналов. Если используется группа каналов, то пропускная способность такого интерфейса увеличивается.

20.3.1 Привязка интерфейса к каналам E1

Привязка интерфейса к каналам E1 задается с помощью команды «backend» в режиме конфигурирования HDLC интерфейса. Пример привязки интерфейса:

```
DionisNX(config)# controller e1 1
DionisNX(config—e1—1)# framing cas
DionisNX(config—e1—1)# coding hdb3
DionisNX(config—e1—1)# timing 1
DionisNX(config—e1—1)# crc4
DionisNX(config—e1—1)# exit
DionisNX(config)# interface hdlc 3
DionisNX(config—if—hdlc3)# backend e1 1 1—15,17—31
```

В данном примере интерфейс hdlc3 будет использовать все доступные каналы (режим CAS) контроллера №1. В режиме CAS канал №16 является служебным.

Следующий пример использует канал №1 для интерфейса hdlc1 и все остальные каналы контроллера для интерфейса hdlc2. Формат кадра CCS.

```
DionisNX(config)# controller e1 1
DionisNX(config—e1—1)# framing ccs
DionisNX(config—e1—1)# coding hdb3
DionisNX(config—e1—1)# timing 1
DionisNX(config—e1—1)# crc4
DionisNX(config—e1—1)# exit
DionisNX(config)# interface hdlc 1
DionisNX(config—if—hdlc3)# backend e1 1 1
DionisNX(config)# interface hdlc 2
DionisNX(config—if—hdlc3)# backend e1 1 2—31
```

В случае нефреймированного потока номера каналов не задаются, так как используется весь поток.

```
DionisNX(config)# controller e1 1
DionisNX(config—e1—1)# unframed
DionisNX(config—e1—1)# coding hdb3
DionisNX(config—e1—1)# timing 1
DionisNX(config—e1—1)# crc4
DionisNX(config—e1—1)# exit
DionisNX(config)# interface hdlc 3
DionisNX(config—if—hdlc3)# backend e1 1
```

20.3.2 Протокол

Для HDLC-интерфейса необходимо также задать протокол передачи данных канального уровня. Доступные протоколы:

- ppp - Протокол PPP (Point-to-Point Protocol). Двухточечный протокол;
- cisco - Версия протокола HDLC, совместимая с маршрутизаторами Cisco.

Задать протокол PPP можно с помощью команды:

```
DionisNX(config—if—hdlc3)# encapsulation ppp
```

Задать протокол Cisco HDLC можно с помощью команды:

```
DionisNX(config—if—hdlc3)# encapsulation cisco
```

В случае использования протокола cisco, существует дополнительный параметр “keepalive”. Для проверки работоспособности интерфейса в канал периодически посылается специальный пакет. Если в течение некоторого тайм-аута на интерфейс не поступило ни одного специального пакета, то интерфейс считается неработоспособным.

```
DionisNX(config—if—hdlc3)# keepalive 10 3
```

Первый аргумент команды “keepalive” - это периодичность отправки специального пакета в секундах. Второй аргумент - количество попыток послать пакет. Тайм-аут, в течении которого ожидаются специальные пакеты от удаленного оборудования, расчитывается как период, умноженный на количество попыток.

20.3.3 Другие настройки

Интерфейс HDLC имеет такие же дополнительные настройки, как и любой другой интерфейс. В том числе HDLC интерфейс может быть включен (команда «enable») или выключен (команда «disable»).

20.3.4 Состояние интерфейса

Текущее состояние HDLC интерфейса можно узнать с помощью команды «show interface».

```
DionisNX# show interface hdlc 1
```

20.3.5 Настройки по умолчанию

По умолчанию (если параметр не указан явно в настройках интерфейса), поля принимают следующие значения:

- Протокол encapsulation cisco
- Отправка пакета “keepalive” (только для cisco) keepalive 10 5

21. Фильтрация и NAT в интерфейсах bridge

В интерфейсах bridge возможно производить фильтрацию на уровне фреймов. Для этого используются списки bridge access-list. Принцип их работы схож с принципом работы ip access-list, однако критерии отбора и команды отличаются.

Кроме того, в Dionis-NX реализован NAT для интерфейсов bridge, позволяющий прозрачно менять MAC-адреса проходящих фреймов. Для этого используются списки bridge nat-list.

21.1 Создание bridge access-list

Создание bridge access-list в целом аналогично созданию ip access-list, однако критерии src/dst/proto будут проверять соответствующие поля в заголовке Ethernet, а не IP. Пример создания bridge access-list:

```
adm@DionisNX(config)# bridge access—list mylist
adm@DionisNX(config—bracl—mylist)# permit src 12:34:56:78:9a:bc
adm@DionisNX(config—bracl—mylist)# permit IPv4 ip—src 192.168.0.0/24
adm@DionisNX(config—bracl—mylist)# permit IPv4 ip—proto icmp
adm@DionisNX(config—bracl—mylist)# deny
adm@DionisNX(config—bracl—mylist)# do show
1 permit src 12:34:56:78:9a:bc
2 permit IPv4 ip—src 192.168.0.0/24
3 permit IPv4 ip—proto icmp
4 deny
```

Доступные команды в списке bridge access-list:

Название правила	Параметры	Действие
deny	правила отбора	Запретить дальнейшее прохождение фрейма
permit	правила отбора	Разрешить дальнейшее прохождение фрейма
call	access-list	Передать управление на другой access-list с возвратом
pass	правила отбора	Ничего не делать с фреймом, перейти к следующему правилу. Может использоваться для подсчёта количества прошедших фреймов, удовлетворяющих определённым критериям.

Название правила	Параметры	Действие
return	правила отбора	Прекратить обработку фрейма в данном access-list'е. Вернуться к родительскому access-list в случае, если в данный access-list фрейм попал при помощи команды call.

21.2 Привязка bridge access-list

В отличие от ip access-list, bridge access-list может использоваться только на конкретном интерфейсе bridge. Доступны три точки привязки bridge access-list:

1. local-in. В этом случае фильтроваться будут фреймы, которые получены в одном из портов сетевого моста, но покидают его (например, фреймы, предназначенные самому маршрутизатору, или фреймы, которые покинут маршрутизатор через порт, не входящий в сетевой мост).
2. local-out. В этом случае фильтроваться будут фреймы, которые попадают в сетевой мост из другого интерфейса или из самого маршрутизатора.
3. forward. В этом случае фильтроваться будут фреймы, которые получены в одном из портов сетевого моста и уходят через порт того же сетевого моста.

Пример привязки bridge access-list:

```
adm@DionisNX(config)# interface bridge 1
adm@DionisNX(config—if—bridge1)# bridge access—group mylist forward
```

21.3 Создание bridge nat-list

Bridge nat-list позволяет прозрачно заменять MAC-адреса проходящих через него фреймов. Для этого доступны два режима работы nat: dnat и snat. Snat используется для замены MAC-адреса источника во фрейме. Применяется к пакетам, покидающим маршрутизатор через сетевой мост. Dnat используется для замеры MAC-адреса получателя. Применяется к пакетам, полученным через сетевой мост.

Кроме того, доступна команда exclude, позволяющая прекратить прохождение фрейма через nat-list.

Пример bridge nat-list:

```
adm@DionisNX(config)# bridge nat—list mynat
adm@DionisNX(config—brnat—mynat)# nat snat mac 22:22:22:22:22:22 src 11:11:11:11:11:11
        snat—arp
adm@DionisNX(config—brnat—mynat)# nat dnat mac 11:11:11:11:11:11 dst 22:22:22:22:22:22
```

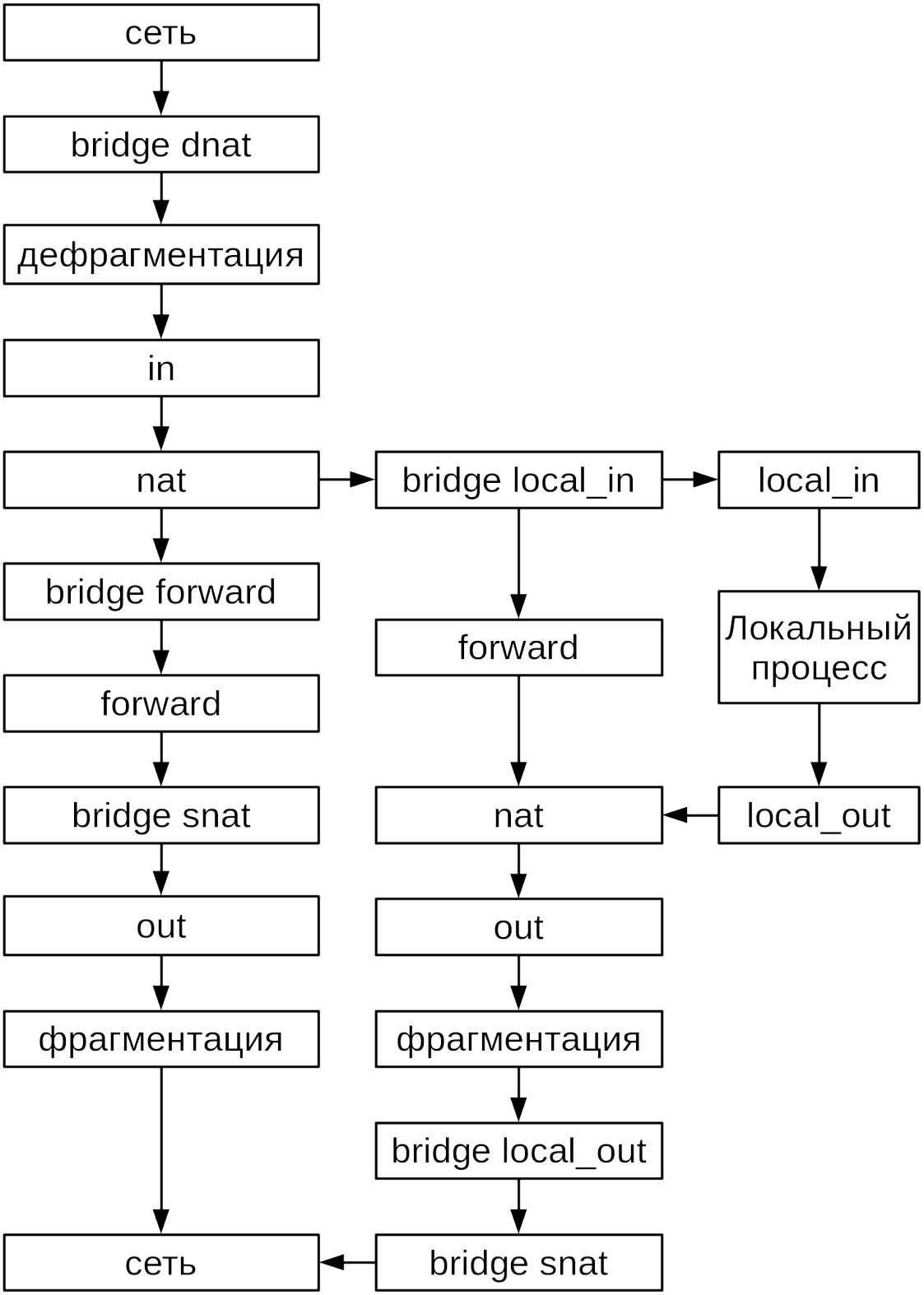

Рис. 21.1: Фильтрация и NAT в bridge

Snat-arp – специальная опция для режима snat, включающая замену mac-адреса в заголовке ARP.

Важным отличием bridge nat-list от ip nat-list является то, что через bridge nat-list проходит каждый фрейм и правила работают только в одну сторону. То есть если написать правило snat mac 22:22:22:22:22:22 src 11:11:11:11:11:11 snat—arp, адрес источника 11:11:11:11:11:11 заменится на 22:22:22:22:22:22. Однако в ответе с адресом назначения 22:22:22:22:22:22 адрес назначения этим правилом изменён не будет.

21.4 Привязка bridge nat-list

bridge nat-list привязываются к интерфейсам bridge аналогично ip nat-list:

```
adm@DionisNX(config)# interface bridge 1
adm@DionisNX(config—if—bridge1)# bridge nat—group mynat
```

21.5 Правила отбора

21.5.1 MAC-адреса источника и назначения

Для задания MAC-адресов источника и назначения используются соответственно параметры src и dst. Можно также указать маску в том же формате, что и MAC-адрес. Маска определяет, какие биты соответственно источника или назначения будут проверяться. Примеры:

```
adm@DionisNX(config—bracl—mylist)# permit src 11:22:33:44:55:66
adm@DionisNX(config—bracl—mylist)# deny dst 11:22:33:44:55:66
adm@DionisNX(config—bracl—mylist)# deny dst 44:44:44:00:00:00/ff:ff:ff:00:00:00
```

списке команд Dionis-NX ### pkttype Параметр pkttype позволяет определить тип адреса назначения Ethernet-фрейма (broadcast/multicast/host/otherhost). Параметры broadcast/multicast означают соответственно broadcast и multicast-фреймы. Параметр host означает пакет, адрес назначения которого совпадает с MAC-адресом устройства, на котором он был принят. Параметр otherhost означает прочие пакеты.

21.5.2 Протокол (EtherType)

Для задания протокола нужно просто написать его как критерий отбора:

```
adm@DionisNX(config—bracl—mylist)# deny IPv6
```

21.5.3 Параметры IPv4/IPv6

При задании протокола IPv4 становятся доступны параметры, позволяющие анализировать заголовок IP. Аналогично, при задании протокола IPv6 можно анализировать заголовок IP6.

Команда для IPv4	Команда для IPv6	Аналог из ip(6) access-list
ip-src	ip6-src	src
ip-dst	ip6-dst	dst
ip-proto	ip6-proto	протокол (указывается как самостоятельный параметр)
ip-sport	ip6-sport	sport
ip-dport	ip6-dport	dport
ip-tos	ip6-tclass	tos

Пример использования:

```
adm@DionisNX(config—bracl—mylist)# permit IPv4 ip—src 192.168.0.0/24
adm@DionisNX(config—bracl—mylist)# deny IPv4 ip—proto tcp ip—dport 80
```

21.5.4 Параметр content

Параметр content аналогичен параметру content в ip access-list, за исключением того, что смещения считаются с начала Ethernet-заголовка:

```
adm@DionisNX(config—bracl—mylist)# permit content 10&0xFFFF=0x800
```

Можно задать режим net-header, при котором смещения будут считаться от начала заголовка IP/IPv6/ARP:

```
adm@DionisNX(config—bracl—mylist)# permit content net—header ”6&0xFF=1 && 4&0x1FFF=0 &&
        0>>22&0x3C@0>>16=0x0301”
```

Режим net-header удобен тем, что упрощает нахождение IP-заголовка в случае, когда во фрейме могут быть VLAN-теги.

21.5.5 Другие критерии отбора

Полный список критериев отбора можно найти в списке команд Dionis-NX.

22. GRE-туннели

Dionis-NX поддерживает туннелирование по протоколу GRE. Для этого используются интерфейсы типа gre. После создания такого виртуального интерфейса, весь трафик, попадающий по правилам маршрутизации на этот туннель, инкапсулируется в GRE-пакеты.

Для создания gre туннеля используется команда interface gre <номер> из режима configure. При этом номер интерфейса может начинаться с 1.

```
DionisNX(config)# interface gre 1
DionisNX(config—if—gre1)#
```

Интерфейс gre в целом настраивается так же, как и интерфейсы других типов, однако имеется набор параметров, которые применимы только для gre-туннелей. Эти параметры приведены в таблице:

команда	параметр
local <IP-адрес>	IP-адрес локального конца туннеля (обязательный параметр)
remote <IP-адрес>	IP-адрес удаленного конца туннеля (обязательный параметр)
ttl <значение>	Принудительно устанавливать значение ttl для GRE-датаграммы, а не наследовать его из инкапсулируемого пакета
keepalive <период> <число повторных попыток>	Включить механизм проб. Пробы посылаются через <период> секунд, и <число повторных попыток> + 1 раз **
tos <тип_трафика>|inherit	Настройка типа трафика (по качеству обслуживания)
checksum	Включить генерацию/проверку контрольных сумм
sequence	Включить упорядочивание пакетов
key <id>	Установить ID для туннеля

** ВНИМАНИЕ! Состояние GRE туннеля с включённым контролем состояния (keepalive) после перезагрузки может иметь значение UNKNOWN - исправить данное поведение можно выполнив перезапуск GRE туннеля командами disable/enable.

Реальное создание интерфейса происходит после того, как заданы параметры local и remote. Например:

```
DionisNX(config)# interface gre 1
DionisNX(config—if—gre1)# local 192.168.0.1
DionisNX(config—if—gre1)# remote 192.168.1.1
DionisNX(config—if—gre1)# ip address 10.0.0.1/32
DionisNX(config—if—gre1)# enable
```

23. GRETAP-туннели

Существует возможность инкапсуляции ethernet-кадров на уровень IP, для этого используется особая разновидность туннелей GRETAP. Синтаксис команд для работы с GRETAP-туннелями полностью повторяет команды для работы с GRE-туннелями, за исключением того, что тип интерфейса задается как gretap, а не как gre. Например:

```
DionisNX(config)# interface gretap 1
DionisNX(config—if—gretap1)# local 192.168.0.1
DionisNX(config—if—gretap1)# remote 192.168.1.1
DionisNX(config—if—gretap1)# keepalive 5
DionisNX(config—if—gretap1)# ip address 10.0.0.1/32
DionisNX(config—if—gretap1)# enable
DionisNX(config—if—gretap1)# do show interface gretap 1 link
```

24. IP6GRE-туннели

Настройка IP6GRE-туннелей аналогична настройке GRE-туннелей. Например:

```
DionisNX(config)# interface ip6gre 2
DionisNX(config—if—ip6gre2)# local 2001:db8::5:8
DionisNX(config—if—ip6gre2)# remote 2001:db8::3:5
DionisNX(config—if—ip6gre2)# keepalive 5
DionisNX(config—if—ip6gre2)# ip address 50.0.0.7/32
DionisNX(config—if—ip6gre2)# enable
```

25. IP6GRETAP-туннели

Настройка IP6GRETAP-туннелей аналогична настройке GRETAP-туннелей. Например:

```
DionisNX(config)# interface ip6gretap 5
DionisNX(config—if—ip6gretap5)# local 2001:db8::12:1
DionisNX(config—if—ip6gretap5)# remote 2001:db8::11:2
DionisNX(config—if—ip6gretap5)# keepalive 5
DionisNX(config—if—ip6gretap5)# ip address 30.0.0.4/32
DionisNX(config—if—ip6gretap5)# enable
```

26. VPN-туннели

26.1 Введение

Dionis-NX имеет поддержку технологии OpenVPN для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами.

Настройка OpenVPN в системе Dionis-NX сводится к настройке динамических интерфейсов типа svpn (серверный интерфейс) и vpn (клиентский интерфейс).

Для использования vpn и svpn интерфейсов в режиме с TLS-аутентификацией или pre-shared ключом защиты необходимо предварительно добавить соответствующие ключи и сертификаты в хранилище сертиаикатов Dionis-NX. Для аутентификации пользователей с помощью связки логин/пароль необходимо предварительно настроить на сервере учетные записи vpn пользователей.

26.2 VPN-интерфейс

Для создания vpn-интерфейса используется команда: interface vpn <номер> из режима configure. При этом необходимо указать на каком уровне модели OSI будет работать данный интерфейс (сетевой или канальный). Выбор уровня задается командой **mode** в режиме конфигурации интерфейса.

Для работы на сетевом уровне модели OSI необходимо ввести команду **mode tun**:

```
adm@DionisNX(config)# interface vpn 1
adm@DionisNX(config—if—vpn1)# mode tun
```

Для работы на канальном уровне модели OSI необходимо ввести команду **mode tap**:

```
adm@DionisNX(config)# interface vpn 1
adm@DionisNX(config—if—vpn1)# mode tap
```

Режимы работы vpn-интерфейса:

1. Простой туннель без защиты;
2. Туннель с pre-shared ключом защиты;
3. Туннель с TLS-аутентификацией;
4. Работа в режиме клиента openvpn для подключения к мультиклиент-серверу openvpn.

Настройка интерфейса происходит в два этапа:

1. Настройка connection-блока;
2. Прочая настройка.

Настройка connection-блока.

Для интерфейса vpn доступно несколько профилей подключения (connection-блоков). Интерфейс поочередно пытается установить соединение с удаленным концом из каждого блока.

Для входа в connection-блок необходимо выполнить команду connection <Имя>

```
adm@DionisNX(config—if—vpn1)# connection block1
adm@DionisNX(config—if—vpn1—block1)
```

Команды доступные для настройки connection-блоков

команда	параметр
lport <Номер порта>	Номер порта на локальном конце туннеля. По умолчанию 1194 (Необязательный параметр)
rport <Номер порта>	Номер порта на удаленном конце туннеля. По умолчанию 1194 (Необязательный параметр)
port <Номер порта>	Номер порта на локальном и удаленном конце туннеля. По умолчанию 1194 (Необязательный параметр)
local <ip или имя хоста>	Локальный ip или имя хоста (Необязательный параметр)
proto <Протокол>	Протокол работы интерфейса. По умолчанию udp. Должен совпадать с протоколом на удаленном конце туннеля
bind	Команда связывает локальный адрес и порт
remote <ip или имя хоста>	Удаленный ip или имя хоста, к которому будет происходить подключение (Обязательный параметр)

Примечание: В connection-блоке может быть несколько remote. Для удаления конкретного remote необходимо выполнить команду:

```
no remote <IP>
```

Например:

```
adm@DionisNX(config—if—conn)# no remote 192.168.10.1
adm@DionisNX(config—if—conn)
```

Прочие настройки.

Дополнительные команды для настройки vpn-интерфейса:

команда	параметр
tls-client	Включить TLS и быть клиентом во время handshake. Эта команда подразумевает обязательный ввод команд <ca>, <cert>, <key>
ca <Корневой сертификат>	Корневой сертификат. Ca-файл должен быть такой же как на сервере. Должен находиться в папке “cert:/” на устройстве.

команда	параметр
cert <Сертификат клиента>	Сертификат клиента. Должен находиться в папке “cert:/” на устройстве.
key <Ключ клиента>	Ключ сертификата клиента. Должен находиться в папке “cert:/” на устройстве.
tls-auth <Дополнительный ключ>	Tls-auth ключ. Данная команда добавляет дополнительный слой аутентификации. Tls-auth-файл должен быть такой же как на сервере (Необязательный параметр. Используется совместно с <tls-client>)
secret <Pre-shared ключ>	Pre-shared ключ в режиме работы туннеля с pre-shared ключом защиты. Файл ключа должен быть одинаковым на обоих концах туннеля
ifconfig <l_IP:r_IP>	Приватный адрес локального и удаленного конца туннеля. (Обязательная команда для всех режимов работы интерфейса кроме режима клиента openvpn. В данном режиме команда не используется)
cipher <Алгоритм>	Алгоритм шифрования.
auth <Алгоритм>	Алгоритм Аутентификации.
ping <Интервал в секундах>	ping удаленного конца туннеля, если нет передачи пакетов в течение промежутка времени, большего чем указанный интервал
ping-exit <Интервал в секундах>	Завершение соедининия с удаленным концом туннеля, если от него не приходило пакетов в течение промежутка времени, большего чем указанный интервал
ping-restart <Интервал в секундах>	Перезагрузка подключения к удаленному концу туннеля, если от него не приходило пакетов в течение промежутка времени, большего чем указанный интервал
pull	Обязательная команда при работе интерфейса в режиме клиента openvpn
ns-cert-type <nsCertType>	Требовать <nsCertType> в поле nsCertType сертификата соседа
tls-cipher <Alg>	Алгоритм tls-шифра. Используется для повышения уровня безопасности контроля канала управления (TLS used only as a control channel).
mode <tun|tap>	Выбор уровня модели OSI на котором будет работать интерфейс. Обязательная команда.
user <NAME> <PASS>	Использовать связку логин/пароль для аутентификации на сервере, если на нем включена поддержка данного режима.
max-routes <N>	Максимальное количество маршрутов получаемых от сервера (по умолчанию 100).

Ниже приводятся примеры конфигурации интерфейса для различных режимов работы. (В данных примерах настраиваемый интерфейс является локальным концом туннеля).

1. Простой туннель без защиты. В данном режиме необходимо указать имя удаленного хоста или его ip-адрес, а так же локальный и удаленный ip-адреса туннеля. Пример настройки интерфейса:

```
adm@DionisNX(config)# interface vpn 1
adm@DionisNX(config—if—vpn1)# mode tun
adm@DionisNX(config—if—vpn1)# connection block1
adm@DionisNX(config—if—vpn1—block1)
adm@DionisNX(config—if—vpn1—block1)# remote 192.168.33.232
adm@DionisNX(config—if—vpn1—block1)# exit
adm@DionisNX(config—if—vpn1)# ifconfig 10.8.0.1:10.8.0.2
adm@DionisNX(config—if—vpn1)# enable
```

Здесь 192.168.33.232 - это адрес удаленного конца. 10.8.0.1 - локальный адрес туннеля, 10.8.0.2 - удаленный адрес туннеля. Удаленный конец туннеля необходимо также настроить для подключения к vpn-туннелю, указав remote и ifconfig. ifconfig на удаленном конце в данном случае будет 10.8.0.2:10.8.0.1

2. Туннель с pre-shared ключом защиты. Данный режим повторяет настройку простого туннеля без защиты. Кроме этого, необходимо на обоих концах туннеля указать pre-shared ключ.

```
adm@DionisNX(config—if—vpn1)# secret cert:/key.pem
```

Ключ должен быть указан как на локальном, так и на удаленном конце туннеля, храниться в секрете и передаваться по защищенному каналу.

3. Туннель с TLS - аутентификацией. Данный режим повторяет настройку простого туннеля без защиты. Кроме того, необходимо выполнить следующее: На локальном конце туннеля выполнить команду <tls-client> а также указать корневой сертификат, и ключ локального конца.

```
adm@DionisNX(config—if—vpn1)# tls—client
adm@DionisNX(config—if—vpn1)# ca cert:/ca.crt
adm@DionisNX(config—if—vpn1)# cert cert:/client.crt
adm@DionisNX(config—if—vpn1)# key cert:/client.key
```

В этом случае удаленный конец туннеля будет выступать в качестве tls-сервера и должен быть настроен соответствующим образом.

4. Работа в режиме клиента openvpn для подключения к мульти-клиент-серверу openvpn. Для подключения к мульти-клиент-серверу openvpn необходимо выполнить следующие команды:

```
adm@DionisNX(config)# interface vpn 1
adm@DionisNX(config—if—vpn1)# mode tun
adm@DionisNX(config—if—vpn1)# connection conn
adm@DionisNX(config—if—vpn1—conn)# remote 192.168.33.232
adm@DionisNX(config—if—vpn1—conn)# exit
adm@DionisNX(config—if—vpn1)# tls—client
adm@DionisNX(config—if—vpn1)# pull
adm@DionisNX(config—if—vpn1)# ca cert:/ca.crt
adm@DionisNX(config—if—vpn1)# cert cert:/client.crt
```

```
adm@DionisNX(config—if—vpn1)# key cert:/client.key
adm@DionisNX(config—if—vpn1)# enable
```

26.3 SVPN-интерфейс

26.3.1 Настройка учетных записей VPN-пользователей

Серверный интерфейс (svpn) предоставляет возможность производить аутенцификацию пользователей с использованием связки логин/пароль. Для этого необходимо создать пользователя, а затем добавить его в настройки svpn-интерфейса и включить соотсетствующий режим аутентификации.

Настройка учетных записей происходит из **enable** режима.

Добавление учетной записи:

```
adm@DionisNX# vpn account create <NAME> <PASSWORD>
```

Удаление учетной записи:

```
adm@DionisNX# vpn account remove <NAME>
```

Изменение пароля учетной записи:

```
adm@DionisNX# vpn account password <NAME> <NEWPASSWORD>
```

Просмотр списка учетных записией:

```
adm@DionisNX# show vpn account
```

Просмотр хеша пароля учетной записии:

```
adm@DionisNX# show vpn account <NAME> passwd—hash
```

Для просмотра информации о подключенных VPN-клиентах необходимо выполнить команду:

```
adm@DionisNX# show interface svpn 1 clients
```

26.3.2 Настройка SVPN-интерфейса

Для создания svpn-интерфейса используется команда: interface svpn <номер> из режима configure. При этом необходимо указать на каком уровне модели OSI будет работать данный интерфейс (сетевой или канальный). Выбор уровня задается командой **mode** в режиме конфигурации интерфейса.

Для работы на сетевом уровне модели OSI необходимо ввести команду **mode tun**:

```
adm@DionisNX(config)# interface svpn 0
adm@DionisNX(config—if—svpn0)# mode tun
```

Для работы на канальном уровне модели OSI необходимо ввести команду **mode tap**:

```
adm@DionisNX(config)# interface svpn 0
adm@DionisNX(config—if—svpn0)# mode tap
```

Режимы работы интерфейса svpn:

1. Сервер для туннеля с TLS-аутентификацией,
2. Мульти-клиент-сервер.

Команды для настройки svpn-интерфейса

команда	параметр
proto <Протокол>	Протокол работы интерфейса. По умолчанию udp. Должен совпадать с протоколом на удаленном конце туннеля
port <Номер порта>	Номер порта. По умолчанию 1194
local <ip или имя хоста>	Локальный ip или имя хоста (Необязательный параметр)
bind	Команда связывает локальный адрес и порт
server <ip сети:маска сети>	IP-адрес и маска создаваемой частной сети.(Только для режима работы Мульти-клиент-сервер)
remote <ip или имя хоста>	Удаленный ip или имя хоста, к которому будет происходить подключение (Режим сервера с TLS-аутентификацией)
ca <Корневой сертификат>	Корневой сертификат. Должен находиться в папке “cert:/” на устройстве.
cert <Сертификат сервера>	Сертификат сервера. Должен находиться в папке “cert:/” на устройстве.
key <Ключ сервера>	Ключ сертификата сервера. Должен находиться в папке “cert:/” на устройстве.
dh <Ключ Диффи-Хеллмана>	Ключ Диффи-Хеллмана. Должен находиться в папке “cert:/” на устройстве.
tls-auth <Дополнительный ключ>	Tls-auth-ключ. Данная команда добавляет дополнительный слой аутентификации. Tls-auth-файл должен быть такой же, как на клиенте (Необязательный параметр)
ifconfig <l_IP:r_IP>	Приватный адрес локального и удаленного конца туннеля. (Используется только для режима сервера в туннеле с TLS-аутентификацией)
cipher <Алгоритм>	Алгоритм шифрования.
auth <Алгоритм>	Алгоритм Аутентификации.
ping <Интервал в секундах>	ping удаленного конца, если нет передачи пакетов в течение промежутка времени, большего чем указанный интервал

команда	параметр
ping-exit <Интервал в секундах>	Завершение соедининия с удаленным концом туннеля, если от него не приходило пакетов в течение промежутка времени, большего чем указанный интервал
ping-restart <Интервал в секундах>	Перезагрузка подключения к удаленному концу туннеля, если от него не приходило пакетов в течение промежутка времени, большего чем указанный интервал
push ping <Интервал в секундах>	Установка значения ping для подключаемых клиентов
push ping-exit <Интервал в секундах>	Установка значения ping-exit для подключаемых клиентов
push ping-restart <Интервал в секундах>	Установка значения ping-restart для подключаемых клиентов
push route <ip сети:маска сети>	Передача клиенту маршрутов, чтобы позволить ему связаться с другими частными подсетями
ns-cert-type <nsCertType>	Требовать <nsCertType> в поле nsCertType сертификата соседа
tls-cipher <Alg>	Алгоритм tls-шифра. Используется для повышения уровня безопасности контроля канала управления.
client-to-client	Команда позволяет подключенным клиентам видеть друг друга
duplicate-cn	Команда позволяет подключаться нескольким клиентам с одинаковым common name в поле сертификата
client-net <Common Name>	Добавление подсети клиента с Common Name в поле сертификата
mode <‘tun’|‘tap’>	Выбор уровня модели OSI на котором будет работать интерфейс. Обязательная команда.
verify-client-cert <‘none’|‘require’|‘optional’> [‘replace-cn’]	Настройка режима аутентификации пользователей. **none**: аутентификация с использованием связки логин/пароль, **require**: аутентификация с использованием сертификатов (режим по умолчанию), **optional**: оба режима аутентификации.
allow <USER>	Добавить пользователя в конфигурацию.
max-routes <N>	Максимальное количество маршрутов для сети клиента (по умолчанию 100).
max-routes-per-client <N>	Максимальное количество внутренних маршрутов клиента (по умолчанию 256).
ifconfig-pool <begin_IP end_IP> [mask <M>]	Задать пул адресов для выделения клиентам.
push dhcp-option dns <IP>	Передача клиенту адреса DNS сервера.
verify-client radius	Настройка аутентификации клиентов с помощью удаленного RADIUS-сервера.

Примечание. Команда:

```
adm@DionisNX(config—if—svpn0)# mode tun
adm@DionisNX(config—if—svpn0)# server 10.8.0.0/24
```

означает, что интерфейсу svpn0 будет назначен адрес 10.8.0.1, а подключаемым клиентам адреса с 10.8.0.4 по 10.8.0.251

Ниже приводятся примеры конфигурации интерфейса для различных режимов работы. (В данных примерах настраиваемый интерфейс является локальным концом туннеля).

1. Сервер для туннеля с TLS-аутентификацией.

В данном режиме необходимо указать имя удаленного хоста или его ip-адрес, а так же локальный и удаленный ip туннеля. Пример настройки интерфейса:

```
adm@DionisNX(config)# interface svpn 0
adm@DionisNX(config—if—svpn0)# mode tun
adm@DionisNX(config—if—svpn0)# remote 192.168.33.232
adm@DionisNX(config—if—svpn0)# ifconfig 10.8.0.1:10.8.0.2
adm@DionisNX(config—if—svpn0)# ca cert:/ca.crt
adm@DionisNX(config—if—svpn0)# cert cert:/server.crt
adm@DionisNX(config—if—svpn0)# key cert:/server.key
adm@DionisNX(config—if—svpn0)# dh cert:/dh1024.key
adm@DionisNX(config—if—svpn0)# enable
```

2. Мульти-клиент-сервер

```
adm@DionisNX(config)# interface svpn 1
adm@DionisNX(config—if—svpn1)# mode tun
adm@DionisNX(config—if—svpn1)# server 10.8.0.0/24
adm@DionisNX(config—if—svpn1)# ca cert:/ca.crt
adm@DionisNX(config—if—svpn1)# cert cert:/server.crt
adm@DionisNX(config—if—svpn1)# key cert:/server.key
adm@DionisNX(config—if—svpn1)# dh cert:/dh1024.key
adm@DionisNX(config—if—svpn1)# ping 10
adm@DionisNX(config—if—svpn1)# ping—restart 120
adm@DionisNX(config—if—svpn1)# push ping 10
adm@DionisNX(config—if—svpn1)# push ping—restart 60
adm@DionisNX(config—if—svpn1)# enable
```

Включение нескольких машин на стороне клиента.

Допустим, что локальная сеть клиента использует адреса 192.168.4.0/24 и что VPN-клиент использует сертификат с common name = Users.

Для включения этой сети на сервере необходимо выполнить следующие команды:

```
adm@DionisNX(config—if—svpn1)# client—net Users
adm@DionisNX(config—if—svpn1—Users)# iroute 192.168.4.0/24
```

Если требуется, чтобы другие клиенты могли видеть данную подсеть, необходимо также выполнить команды:

```
adm@DionisNX(config—if—svpn1)# client—to—client
adm@DionisNX(config—if—svpn1)# push route 192.168.4.0/24
```

Если требуется назначить конкретный ip-адрес клиенту, необходимо выполнить следующие команды (в данном примере VPN-клиент использует сертификат с common name = Users. Требуется назначить ip 10.8.0.113):

```
adm@DionisNX(config—if—svpn1)# client—net Users
adm@DionisNX(config—if—svpn1—Users)# ifconfig—push 10.8.0.113/24
```

Включение аутентификации с помощью удаленного RADIUS-сервера.

Авторицазия клиентов, назначение им IP адресов и передача маршрутной информации может происходить с помощью удаленного RADIUS-сервера. Для этого необходимо настроить интерфейс в режиме **Мульти-клиент-сервер**, установить режим **verify-client-cert require**, а также дополнительно настроить параметры подключения с RADIUS-серверу.

Ниже приведены команды для настройки данного подключения:

```
adm@DionisNX(config—if—svpn1)# verify—client radius
adm@DionisNX(config—if—svpn1—radius)# nas—id OpenVPN
adm@DionisNX(config—if—svpn1—radius)# nas—ip 127.0.0.1
adm@DionisNX(config—if—svpn1—radius)# radius—ip 192.168.1.1
adm@DionisNX(config—if—svpn1—radius)# radius—pass BigPassword
```

Рассмотрим подробнее команды:

- первая команда: включает режим аутентификации с помощью RADIUS-сервера;
- вторая команда: задает идентификатор NAS (необязательная команда);
- третья команда: задает ip адрес NAS (необязательная команда);
- четвертая команда: задает ip адрес RADIUS-сервера;
- пятая команда: задает пароль для подключения к RADIUS-серверу.

Такая конфигурация предполагает, что vpn-клиент будет подключаться с использованием связки логин/пароль (команда ‘**user <NAME> <PASS>**’ в настройке vpn-интерфейса).

Ниже приведен пример конфигурирования пользователя *u1* с паролем *555* на RADIUS-сервере. В данном случае используется фрагмент конфигурации сервера FreeRADIUS (http://freeradius.org):

```
u1      Cleartext—Password:= ”555”
        Service—Type = Framed—User,
        Framed—IP—Netmask = 255.255.255.0,
        Framed—IP—Address = 10.8.0.33,
        Framed—Routing = Broadcast—Listen,
        Framed—Compression = Van—Jacobsen—TCP—IP,
        Framed—Route += ’192.168.101.0/24 10.8.0.1/32 1’,
        Acct—Interim—Interval=5,
        Ascend—Data—Rate=100,
        Ascend—Xmit—Rate=200,
        Framed—Protocol = PPP
```

В данном примене клиенту *u1* при успешной авторизации будет назначен адрес *10.8.0.33/24*, а также передан маршрут до сети *192.168.101.0/24* через шлюз *10.8.0.1/32*.

27. Экспорт статистики по Netflow

Экспорт статистики по протоколу Netflow предоставляет возможность анализа сетевого трафика на уровне сеансов, делая запись о каждой транзакции TCP/IP. Dionis-NX может выступать в роли сенсора и передавать информацию на коллектор для выбранных интерфейсов. Поддерживаемые версии протоколов: 3, 5, 9.

Для того чтобы активировать экспорт статистики, необходимо для всех выбранных интерфейсов включить flow cache с помощью команды ip flow:

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip flow
```

Затем, нужно войти в режим конфигурации ip flow-export и настроить параметры передачи по Netflow:

```
DionisNX(config)# ip flow—export
DionisNX(config—ip—flow—export)#
```

Ниже перечислены все параметры конфигурации flow-export:

destination <ip> <port>	Указать адрес коллектора
no ip destination	Удалить адреса коллектора
version <3|5|9>	Задать версию протокола Netflow
no version	Версия протокола Netflow по умолчанию
max-flows <число>	Максимальное число потоков
no max-flows	Значение по умолчанию 8192
hoplimit <ttl>	Значение TTL
no hoplimit	По умолчанию 1
level <full|proto|ip>	Уровень информации
no level	По умолчанию full
timeout <general|tcp|tcp.rst|tcp.fin|udp|maxlife|expiry> <интервал>	Задать временной интервал
[no] timeout <general|tcp|tcp.rst|tcp.fin|udp|maxlife|expiry>	Установить значение по умолчанию для временного интервала
enable	Включить экспорт
disable	Выключить экспорт

Основными настройками являются:

- destination;
- version.

С помощью параметра timeout настраиваются временные промежутки устаревания потока. Когда поток устаревает, информация о нём пересылается на коллектор. Например, если TCP соединение закрывается, то оно устаревает и информация о нём отправляется на коллектор. Если же TCP соединение не закрывается долгое время, то информация об этом соединении может быть отправлена очень

нескоро. Поэтому, иногда возникает необходимость задавать меньшие интервалы устаревания. Ниже приводится подробное описание интервалов.

general	Интервал, относящийся ко всему трафику, если только не определён интервал для конкретного типа трафика
tcp	Интервал для открытых TCP соединений
tcp.rst	Интервал для TCP соединений, после того как пакет RST был послан одной из сторон
tcp.fin	Интервал для TCP соединений, после того как пакеты FIN были посланы обеими сторонами
udp	Интервал для UDP соединений
maxlife	Максимальное время жизни потока. Для отключения - задайте 0
expint	Задать интервал между проверками. Увеличение параметра приводит к тому, что в NetFlow пакет будет помещено больше потоков. Для отключения – задайте 0

После настройки, необходимо активировать экспорт командой enable, например:

```
DionisNX(config)# ip flow—export
DionisNX(config—ip—flow—export)# destination 192.168.0.254 8854
DionisNX(config—ip—flow—export)# version 9
DionisNX(config—ip—flow—export)# enable
```

Для диагностики работы flow кэша (кэша статистики соединений), можно воспользоваться командой show interface:

```
DionisNX# show interface ethernet 0 flow dump
```

28. Служба NTP

В Dionis-NX реализована служба синхронизации времени. Данная служба может получать информацию о точном времени от других NTP-серверов, а также может являться NTP-сервером для других узлов.

Чтобы войти в режим настройки службы NTP, нужно выполнить команду в режиме конфигурации:

```
(config)# service ntp
```

Далее нужно добавить имена (IP-адреса)серверов времени, с которыми будет осуществляться синхронизация. Пример:

```
(config—service—ntp)# server 1.1.1.1
(config—service—ntp)# server myntpserver
(config—service—ntp)# servers 0.ru.pool.ntp.org
```

Команды «server» и «servers» отличаются только тем, что если FQDN-имя, указанное командой «servers», разрешается в несколько IP-адресов пула NTP-серверов, то будут делаться попытки осуществлять синхронизацию со всеми этими серверами. В случае команды «server» - только с первым IP-адресом, в который разрешилось FQDN.

Список серверов является упорядоченным. Серверы в начале списка имеют больший приоритет. Список можно редактировать с помощью команд с числовыми префиксами и команд «no». Пример:

```
(config—service—ntp)# do show
1 server 1.1.1.1
2 server myntpserver
3 servers 0.ru.pool.ntp.org
(config—service—ntp)# no 2
(config—service—ntp)# do show
1 server 1.1.1.1
2 servers 0.ru.pool.ntp.org
(config—service—ntp)# 1 servers 1.ru.pool.ntp.org
(config—service—ntp)# do show
1 servers 1.ru.pool.ntp.org
2 server 1.1.1.1
3 servers 0.ru.pool.ntp.org
(config—service—ntp)# no all
(config—service—ntp)# do show
```

По умолчанию служба NTP только синхронизирует время от других серверов и сама не является сервером. Чтобы служба NTP стала сервером, необходимо объявить адрес(а), на котором откроется слушающий сокет. Это делается с помощью команды «listen». Допустимы множественные команды «listen».

Например, следующая команда предписывает принимать NTP-запросы на всех интерфейсах:

```
(config—service—ntp)# listen 0.0.0.0
```

Или на некоторых интерфейсах:

```
(config—service—ntp)# listen 10.1.1.1
(config—service—ntp)# listen 10.1.2.1
```

«Слушающие» адреса могут быть удалены командой:

```
(config—service—ntp)# no listen <ip>
```

Для работы с IPv6 адресами используйте команды: listen6 и no listen6.

Для задания часового слоя (stratum) используется команда stratum:

```
(config—service—ntp)# stratum 2
```

Допустимый диапазон значений: от 1 до 15. По умолчанию, если команда не задана, часовой слой установлен в 1. Слой 1 – первичные серверы. Слой 2 – вторичные серверы, синхронизируются с первичными серверами, и т.д.

После настройки службы NTP её необходимо активировать командой:

```
(config—service—ntp)# enable
```

По умолчанию, при выполнении данной команды служба NTP пытается синхронизироваться с NTP-серверами немедленно. Это может вызывать длительные задержки. Если эти задержки недопустимы, то необходимо указать опцию:

```
(config—service—ntp)# sync off
```

В этом случае команда «enable» завершится немедленно, и синхронизация времени будет происходить в фоновом режиме.

Чтобы вернуть поведение по умолчанию, необходима опция:

```
(config—service—ntp)# sync on
```

Кроме того для синхронизации (как немедленной, так и в фоновом режиме) возможно указать предельно-допустимое отклонение от времени на сервере. То есть если время на клиентской машине отличается в большую или меньшую сторону на величину, превышающую предельное отклонение, то синхронизация времени не произойдет:

```
(config—service—ntp)# max—offset <n>
```

Команда устанавливает допустимое отклонение в <n> секунд.

Чтобы вернуть поведение по умолчанию, необходимо выполнить команду:

```
(config—service—ntp)# no max—offset
```

Чтобы остановить службу, нужно выполнить команду:

```
(config—service—ntp)# disable
```

Если служба NTP находится в активированном состоянии, и при этом осуществляется редактирование её настроек, то для того чтобы они вступили в силу, необходимо перезапустить службу:

```
(config—service—ntp)# disable
(config—service—ntp)# enable
```

Чтобы посмотреть журнал службы NTP, нужно выполнить одну из команд привилегированного режима:

```
# show service ntp log
# show service ntp log <n>
# show service ntp log all
# show service ntp log follow
```

Команды выводят, соответственно: 25 последних строк журнала, <n> последних строк журнала, весь журнал, последние строки журнала и последующие при их появлении (режим слежения).

Чтобы остановить службу и удалить все настройки, нужно выполнить команду режима конфигурации:

```
(config)# no service ntp
```

29. Служба DNS

В Dionis-NX реализована возможность настройки службы DNS, обеспечивающая функции по разрешению DNS-запросов.

Для понимания команд настройки необходимо ввести ряд понятий, непосредственно используемых при настройке:

- вид (view): виды позволяют использовать различные настройки сервера при общении с различными наборами узлов; по сути они определяют новый экземпляр сервиса, хотя физически в системе выполняется один процесс сервиса;
- мастер-зона (master): DNS-зона, для которой настраиваемый сервер является мастер-сервером, т.е. является первичным уполномоченным сервером;
- слэйв-зона (slave): DNS-зона, для которой настраиваемый сервер является вторичным уполномоченным сервером; содержимое зоны считывается от одного из серверов, указанных в опции masters данной слэйв-зоны;
- корневая зона (root): описание корневых DNS-серверов; самый новый список корневых серверов можно найти на ftp://ftp.rs.internic.net/domain/named.root;
- форвард-зона (forward): это зона ретрансляции запросов, позволяет перенаправить DNS-запросы по зоне другим серверам;

- зона запрета (blackhole): позволяет фильтровать (запрещать) разрешение указанных “плохих” доменов

Различают два типа DNS-запросов: рекурсивные и итеративные:

- при рекурсивном запросе сервер имен должен найти информацию самостоятельно. То есть при получении рекурсивного запроса сервер имен при отсутствии у него ответа на запрос должен сам обратиться к помощи других серверов имен, например к корневым серверам (данный запрос будет итеративным). Они сами не дадут ответа, но зато направят на другие DNS-серверы. Сервер имен будет проверять все предоставленные ему ссылки, пока не обнаружит необходимую информацию.
- при итеративном запросе сервер имен должен сразу предоставить ответ, не обращаясь к другим DNS-серверам. Если же данный сервер не может предоставить запрошенную информацию, то он возвратит ссылку на другой сервер имен, который, вероятно, может дать ответ на запрошенную информацию.

Основные концепции сервиса DNS - это иерархичность и наследуемость конфигурационных пространств. Иерархия следующая:

- service dns - сервис DNS (уровень A)
  - view <view-name> - виды (уровень B)
    - zone <master|slave|forward|.> - зоны (уровень C)

Самый верхний уровень конфигурации сервиса - это команды уровня service dns.

Часть этих команд есть также на нижележащих уровнях.

Если некоторая опция будет задана на самом верхнем уровне (A), то она определится и для всех нижележащих уровней. Если же значение этой опции будет изменено на каком-либо нижележащем уровне (например, B), то только на этом уровне значение данной опции будет отличаться от всех остальных.

Замечание по формату описания параметров: запись {<NAME>,3} - означает список из максимум трех параметров типа NAME.

Чтобы войти в режим настройки службы DNS, нужно выполнить команду в режиме конфигурации:

```
(config)# service dns
```

29.1 Контроль доступа

acl <NAME> {<IPLIST>,12}

Команда acl задает именованный набор IP-адресов.

Далее созданный ACL можно использовать в командах, где параметр имеет тип IPLIST.

NAME - имя набора адресов

IPLIST - набор IP-адресов, имеет следующий формат:

<any | none | localips | localnets | NAME | [!]A.B.C.D[/MASK] >

- any,none,localips,localnets - имена встроенных наборов:
  - any - любые адреса;
  - none - никакие адреса;
  - localips - IP-адреса, присвоенные интерфейсам системы в момент выполнения команды включения сервиса;
  - localnets - IP-адреса сетей, обслуживаемых интерфейсами системы: все возможный адреса, обслуживаемые интерфейсами системы, например: 192.168.0.1/24 назначен интерфейсу ethernet0, это значит что сервис будет слушать запросы на адресе 192.168.0.1 и ,потенциально,если данные адреса появятся в будущем, - на адресах 192.168.0.2-254;
- NAME - это имя другого ACL, ранее созданного;
- [!] A.B.C.D[/MASK]:
  - если в начальной позиции не указан знак »!», то это IP-адрес узла или сети(если указана маска MASK);
  - если в начальной позиции указан знак »!», то это любые адреса,кроме указанных после знака »!».

Для замены значения уже существующего acl под тем же самым порядковым номером возможны два способа:

- удалить данный acl и добавить этот же acl под нужным или тем же порядковым номером, но с другим значением;

- более простой способ: добавить этот же acl, но с другим значением; при этом можно указать тот же самый порядковый номер; указание другого порядкового номера воспринимается как перемещение acl в списке и не поддерживается, для этого используйте первый способ.

Примеры:

```
(config—service—dns)# acl a1 1.2.3.0/24
(config—service—dns)# acl a2 a1 1.2.3.1
(config—service—dns)# acl a3 a2 5.5.5.5 !10.0.0.0/24
```

allow query {<IPLIST>,3}

Определяет, каким именно узлам разрешено выполнять ДНС-запросы через настраиваемый сервер.

По умолчанию: все.

allow query-on {<IPLIST>,3}

Определяет, на каких внутренних интерфейсах сервера разрешено принимать ДНС-запросы. По умолчанию: все.

allow query-cache {<IPLIST>,3}

Определяет, каким именно узлам разрешено получать ответы на запросы из кэша ДНС настраиваемого сервера.

По умолчанию: все.

allow notify {<IPLIST>,3}

Определяет, каким узлам ,помимо первичных, разрешено уведомлять настраиваемый сервер (если он является для зоны вторичным) об изменениях зоны.

По умолчанию: запрещено всем, кроме мастер-серверов зоны.

allow recursion {<IPLIST>,3}

Указывает, каким узлам разрешено выполнять рекурсивные запросы через данный сервер.

Блокирование рекурсивных запросов для узла не предотвращает получение этим узлом данных, находящихся в кэше ДНС.

По умолчанию: все.

allow transfer {<IPLIST>,3}

Указывает, каким узлам разрешено получать зоны от настраиваемого сервера.

По умолчанию: все.

29.2 Виды

Виды позволяют использовать различные настройки сервера при общении с различными наборами узлов.

Для конфигурации зон необходимо войти в режим вида, даже если функционал видов при настройке не требуется.

Вид - это отдельный экземпляр сервера ДНС со своими настройками и зонами.

На все виды распространяются глобальные опции сервиса.

Чтобы войти в режим настройки вида DNS, нужно выполнить команду в режиме конфигурации службы:

[N] view <VNAME>

Создает вид с именем VNAME и вставляет его в позицию N в списке видов.

Чем меньше N, тем вид приоритетнее, с точки зрения попадания запросов в него.

Входящий DNS-запрос анализируется на предмет соответствия указанным правилам попадания в вид.

Анализ попадания в вид происходит начиная от вида с номером 1 и далее по порядку, пока не будет достигнут вид с максимальным номером.

Если запрос попал в какой-либо вид, то он будет обслуживаться этим видом; далее поиск вида, в который может попасть запрос, прекращается, даже если далее будут виды, в которые данный запрос может попасть.

Пример:

```
(config—service—dns)# view vname
```

Рассмотрим команды вида, позволяющие указать, какие запросы должны в него попадать.

match clients <IPLIST>

Опция указывает, что данный вид будет обслуживать только тех клиентов, IP-адреса которых входят в IPLIST.

match destinations <IPLIST>

Опция указывает, что данный вид будет обслуживать только тех клиентов, которые обращаются к ДНС-серверу по IP-адресам, входящим в IPLIST.

match recursive-only <yes | no>

Если значение yes, то данный вид будет обслуживать только рекурсивные запросы, иначе - все запросы.

По умолчанию: match recursive-only no.

no-empty-zones

Не создает автоматические мастер-зоны 0.in-addr.arpa., 127.in-addr.arpa., 255.in-addr.arpa. и localhost. с параметрами по умолчанию.

По умолчанию эти зоны создаются автоматически.

Примеры

```
(config—dns—vname)# match clients 1.2.3.0/24
(config—dns—vname)# match destinations 192.168.0.1 192.168.0.2
(config—dns—vname)# match recursive—only yes
```

29.3 Зоны

DNS-записи, т.н. ресурсные записи, хранятся в зонах. Служба поддерживает 4 типа DNS зон:

- мастер-зона: зона, для которой настраиваемый сервер является является первичным уполномоченным сервером;
- слэйв-зона: зона, для которой настраиваемый сервер является вторичным уполномоченным сервером;
- корневая зона: описание корневых ДНС-серверов; в это описание можно добавлять любые другие записи, кроме soa-записи;
- зона ретрансляции: позволяет перенаправить DNS-запросы по зоне другим серверам;
- зона запрета: позволяет задать одинаковое разрешение имен для указанного списка доменов (обычно, запрещенных)

Создание ресурсных записей возможно только для мастер-зоны и корневой зоны. Причем для корневой зоны не поддерживается ресурсная запись SOA.

Имена зон должны оканчиваться точкой.

Имена обратных зон должны оканчиваться на **in-addr.arpa.** для IPv4-зоны и на **ip6.arpa.** для IPv6-зоны.

Формирование доменного имени обратной зоны:

- IPv4 зона: необходимо отктеты IP-адреса сети, которая сооветствует зоне, написать в обратном порядке и добавить окончание **in-addr.arpa.**; начальные нули можно опустить;
- IPv6 зона: необходимо представить IPv6-префикс, в который входят узлы зоны, в полном виде, т.е. удалить сжатие за счет нулей (если оно есть) и далее записать полученный префикс в обратном порядке, добаваив в конце окончание **ip6.arpa.**;

Рассмотрим примеры:

- пусть доменные имена зоны example.int. принадлежат сети 192.168.0.0/24; тогда имя обратной зоны будет таким: 168.192.in-addr.arpa.
- пусть доменные имена зоны example.int. принадлежат префиксу 2001:db8:abdc::/64; тогда полный префикс будет 2001:0db8:abdc::, а имя обратной зоны будет 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa

При создании ресурсных записей используются три основных типа данных: IP-адрес, число, имя домена.

Первые два типа данных не нуждаются в пояснении. Особое внимание следует обратить на третий тип данных - имя домена. Можно задать имя домена в двух формах:

- имя оканчивается точкой: это абсолютное имя,полностью задающее домен и зону, в которой он находится;
- имя не оканчивается точкой: это относительное имя домена; такой домен рассматривается как домен зоны,в которой он описан(через ресурсную запись зоны).

Чтобы получить полное имя домена из относительного, нужно справа от относительного имени приписать имя зоны,в которой домен описан (без точки), и завершить полное имя точкой.

Пользуясь приведенным правилом, рассмотрим примеры:

- для зоны «zeta.» ресурсная запись a 1.1.1.1 domain ns задает IP-адрес 1.1.1.1 для домена «ns.zeta.»;
- для корневой зоны “.” ресурсная запись a 2.2.2.2 domain ns задает адрес 2.2.2.2 для домена «ns.».

Рассмторим пример настройки обратной IPv4-зоны:

```
zone master 33.168.192.in—addr.arpa.
 soa master raul.cuba.int. admin admin@raul.cuba.int
 ns raul.cuba.int.
 ptr 160 havana.cuba.int.
 ptr 1 fidel.cuba.int.
```

В примере видно, что зона обслуживает узлы со следующими адресами: 192.168.33.1, 192.168.33.160. У этих узлов есть как прямое имя домена (например, fidel.cuba.int.), так и обратное (1.33.168.192.in-addr.arpa.), которое, для краткости, может быть задано через ресурсную запись **ptr 1 fidel.cuba.int.**, или же, в полной форме, **ptr 1.33.168.192.in-addr.arpa. fidel.cuba.int.**

Рассмторим пример настройки обратной IPv6-зоны:

```
zone master 0.0.0.0.c.d.b.a.8.b.d.0.1.0.0.2.ip6.arpa
 soa master raul.cuba.int. admin admin@raul.cuba.int
 ns raul.cuba.int.
 ptr 4.3.2.1.0.0.0.0.0.0.0.0.0.0.0.0 fidel6.cuba.int.
```

В примере видно, что зона обслуживает узел со следующим адресом: 2001:db8:abdc::1234.

Создание любой зоны возможно только в режиме конфигурации вида. Допустим, создан вид с именем vname.

Рассмотрим далее команды создания зон, а также команды, специфичные именно для зон данного типа.

29.3.1 Мастер зона

zone master <ZNAME>

Создает мастер-зону с именем ZNAME и входит в режим конфигурации зоны.

В мастер-зоне можно задавать любые ресурсные записи. Команды создания ресурсных записей рассмотрены ниже.

ЗАМЕЧАНИЕ: *имя зоны может содержать неразрешенный RFC символ "_", это сделано для возможности задания специальных зон, которые могут понадобиться для интеграции с Windows, поскольку Windows DNS-сервер создает такие служебные зоны (с символом "_" в своем имени) для своих внутренних нужд.*

update [password <PAS> | acl <ACL>]

Включить динамическое обновление A- и PTR- записей для зоны. Параметры команды:

- PAS - пароль, который можно использовать для организации удаленного обновления A- или PTR-записей зоны
- ACL - список контроля доступа, который задает IP-адреса, от которых разрешено получать обновления любых записей зоны (не только A и PTR)

Более подробно о настройке динамического обновления см.**Динамическое обновление зон** .

29.3.2 Слэйв зона

zone slave <ZNAME>

Создает слэйв-зону (подчиненную зону) с именем ZNAME и входит в режим конфигурации зоны.

Ресурсные данные для зоны будут автоматически получаться с мастер-серверов, когда зонные данные изменяются (режим нотификации) и когда истекает TTL записи SOA-зоны.

ЗАМЕЧАНИЕ: *имя зоны может содержать неразрешенный RFC символ "_", это сделано для возможности задания специальных зон, которые могут понадобиться для интеграции с Windows, поскольку Windows DNS-сервер создает такие служебные зоны (с символом "*"в своем имени) для своих внутренних нужд._

masters {<IP[:port]>,5}

Позволяет задать максимум 5 серверов, являющихся первичными для данной зоны.

Зонную информацию служба будет получать от этих мастер-серверов.

Пример:

```
(config—dns—vname—slv—zeta.)# masters 1.2.3.4 2.3.4.5
```

29.3.3 Форвард зона

zone forward <ZNAME>

Создает зону ретрансляции с именем ZNAME и входит в режим конфигурации зоны.

Эта зона, все запросы по которой сразу же обслуживаются указанными в данной зоне ретрансляторами, т.е. другими серверами имен.

Единственная команда зоны - это forwarders. Она уже рассмотрена в разделе Другие настройки.

Данная команда определяет список ретрансляторов, т.е. серверов, которым будут перенаправляться запросы DNS.

Перенаправленный ретранслятору запрос является рекурсивным, т.е. ожидается, что ретранслятор вернет окончательный ответ.

По умолчанию в службе DNS разрешена рекурсия, т.е запросы службы к другим серверам имен являются итеративными, т.е. служба DNS сама будет производить поиск окончательного ответа.

Данная команда может быть выполнена в любом типе зоны, кроме корневой, а также на глобальном уровне и уровне видов.

Алгоритм работы службы различается в зависимости от того, на какой тип зоны распространяется данная команда:

- если запрос попадает в авторитетную зону (мастер- или слэйв-) настроенную на ретрансляцию:
  - сначала ответ ищется в кэше или данных зоны;
  - если ответ не найден - посылается рекурсивный запрос ретранслятору;
  - если ответ от ретранслятора не пришёл - сервером начинается самостоятельный итеративный поиск ответа;
- если запрос попадает в зону ретрансляции, то он сразу же обслуживается указанными в данной зоне ретрансляторами.

Список ретрансляторов может быть пустым. Это полезно,если набор ретрансляторов установлен для всех зон глобально, но для одной из зон не нужно использовать ретрансляторы.

ЗАМЕЧАНИЕ: *имя зоны может содержать неразрешенный RFC символ "* это сделано для возможности задания специальных зон, которые могут понадобиться для интеграции с Windows, поскольку Windows DNS-сервер создает такие служебные зоны (с символом *""в своем имени) для своих внутренних нужд.*

Пример:

```
(config—dns—vname—fwd—zeta.)# forwarders 1.2.3.4 2.3.4.5 only
```

29.3.4 Корневая зона

zone .

Позволяет задать корневую зону.

auto [[[daily | weekly | monthly] [SRV] [TIMEOUT]] | static]

Команда автоматического обновления корневой зоны.

Параметры:

- daily | weekly | monthly - частота (ежедневно,еженедельно или ежемесячно) загрузки зонных данных для корневой зоны из внешней сети по URL: ftp://ftp.rs.internic.net/domain/db.cache;
- SRV - имя домена ftp-сервера или его IP-адрес; путь для загрузки зонных данных при использовании этого параметра будет выглядеть следующим образом: ftp://SRV/domain/db.cache;
- TIMEOUT - максимальное время соединения с SRV, по умолчанию 5 сек;
- static - используется корневая зона по-умолчанию, датированная июнем 2011 года, взятая с вышеуказанного URL. Периодического обновления в данном случае не происходит.

Зонный файл для корневой зоны может быть получен из внешней сети по URL: ftp://ftp.rs.internic.net/domain/db.cache и далее обновлен с указанной периодичностью: ежедневно, еженедельно (по умолчанию) или ежемесячно.

Команды создания RR-записей игнорируются в случае наличия команды auto, кроме auto static. В последнем случае к данным корневой зоны по умолчанию добавляются любые дополнительные ресурсные записи, кроме SOA-записи.

Параметры команды по умолчанию: auto weekly ftp.rs.internic.net

29.3.5 Зона запрета

zone blackhole

Позволяет задать зону запрещенных доменов. Домены, которые добавлены в данную зону, будут разрешаться в некоторый заранее опредленный IP-адрес, либо в адрес 127.0.0.1. В результе чего, последующие Интернет-запросы на данные домены будут уходить вникуда, например на localhost (отсюда букв. название зоны “черная дыра”). Это дает еще один способ фильтрации “плохих” или запрещенных доменов.

blacklist <PATH>

Задает текстовый список запрещенных доменных имен. Доменные имена могут быть любые: краткие или полные.

Для каждого имени домена будет сформировано имя зоны путем добавления точки (“.”) к имени домена. Содержимое зоны будет создано автоматически исходя из параметров по-умолчанию:

- имя мастер-сервера зоны: localhost.
- IP-адрес мастер-сервера зоны: 127.0.0.1
- A-запись для доменов зоны и для самой зоны: 127.0.0.1
- AAAA-запись для доменов зоны и для самой зоны: ::1

Для задания других настроек зоны можно использовать следующую команду:

blackzone [master <DOMAIN> <IP>] [a <IP_ADDR>] [aaa <IP6_ADDR>]

Параметры команды:

- DOMAIN и IP - доменное имя мастер-сервера зоны и его IP-адрес;

- IP_ADDR и IP6_ADDR - адреса для A- и AAAA-записей соответственно, в которые будут разрешаться поддомены зоны и сама зона.

29.3.6 Команды создания ресурсных записей

Команды создания ресурсных записей имеют смысл только для мастер-зоны и корневой зоны (в режиме auto static).

ttl <N>

Команда указывает время жизни по умолчанию для всех записей зоны.

По умолчанию: 86400

soa [master <MNAME> admin <EMAIL> ttl <TTL> refresh <NRF> retry <NRT> expire <EXP> negttl <NEGTTL>]

Команда создает заголовочную SOA-запись.

Необязательные параметры:

- MNAME - имя мастер-сервера зоны; по умолчанию - имя зоны;
- EMAIL - почтовый адрес администратора зоны; по умолчанию - root@<имя зоны>;
- TTL - время жизни записи; если не указано, используется значение ttl для зоны;
- NRF - период обновления зоны вторичным сервером; по умолчанию - 21600 сек;
- NRT - период повторной попытки обновления зоны вторичным сервером; по умолчанию - 1800 сек;
- EXP - интервал устаревания данных зоны для вторичного сервера; по его истечении данные, содержащиеся в зоне, не будут использоваться для ответов на запросы; по умолчанию - 1209600 сек;
- NEGTTL - время жизни отрицательных (неуспешных) ответов в кэше; по умолчанию - 1000 сек.

a <IP> [<DOMAIN>] [ttl <N>]

Команда создает адресную A-запись.

Обязательные параметры:

- IP-адрес.

Необязательные параметры:

- имя домена NAME; если не указан,используется имя текущей зоны;
- время жизни записи N; если не указано, используется значение ttl для зоны.

aaaa <IP6> [<DOMAIN>] [ttl <N>]

Команда создает адресную AAAA-запись.

Обязательные параметры:

- IPv6-адрес.

Необязательные параметры:

- имя домена NAME; если не указан,используется имя текущей зоны;
- время жизни записи N; если не указано, используется значение ttl для зоны.

cname <ANAME> <CNAME> [ttl <N>]

Команда создает алиасную CNAME-запись.

Обязательные параметры:

- имя алиаса ANAME;
- каноническое имя CNAME для алиаса ANAME.

Необязательные параметры:

- время жизни записи N; если не указано, используется значение ttl для зоны;

ds <DOM> <TAG> <ALG> <DIGTYPE> <DIGEST>

Команда создает DS-запись.

Это ресурсная запись, генерируемая из открытого KSK-ключа зоны и публикуемая в родительской зоне; это нужно для построения цепочки доверия, когда родительская зона как бы поручается за все свои подчиненные зоны.

Обязательные параметры:

- DOM - полное доменное имя зоны;
- TAG - идентификатор KSK-ключа;
- ALG - криптографический алгоритм KSK ключа;
- DIGTYPE - криптографический алгоритм хэша;
- DIGEST - строка хэша

Значения параметров команды можно узнать по команде **service dns sec ds** .

Примечание: создание цепочки доверия в основном используется для прямых зон, ввиду того, что для обратных зон не все провайдеры поддерживают DNSSEC, в результате создать цепочку доверия от обратной зоны до корня in-addr.arpa проблематично.

ds-from-file <FILE>

Создает DS-запись из файла, ранее созданного командой **service dns sec ds** (обычно, на другой Dionis-NX системе, которая обслуживает подчиненную зону).

mx <NAME> <prio> [<NAME>] [ttl <N>]

Команда создает почтовую MX-запись.

Обязательные параметры:

- имя почтового ретранслятора NAME;
- приоритет почтового ретранслятора.
- имя домена NAME;

Необязательные параметры:

- время жизни записи N; если не указано, используется значение ttl для зоны.

ns <NSNAME> [<DOMAIN>] [ttl <N>]

Команда создает NS-запись.

Обязательные параметры:

- NSNAME - имя сервера имен.

Необязательные параметры:

- имя домена NAME; если не указано, используется имя текущей зоны;
- время жизни записи N; если не указано, используется значение ttl для зоны.

txt <STRING> [<DOMAIN>] [ttl <N>]

Команда создает TXT-запись.

Обязательные параметры:

- STRING - текстовая запись для описания домена.

Необязательные параметры:

- имя домена DOMAIN; если не указано, используется имя текущей зоны;
- время жизни записи N; если не указано, используется значение ttl для зоны.

ptr <ARPANAME> <FULLNAME> [ttl <N>]

Команда создает обратную адресную PTR-запись.

Обязательные параметры:

- ARPANAME - поддомен текущей обратной зоны;
- FULLNAME - каноническое(полное) имя домена для ARPANAME.

Необязательные параметры:

- время жизни записи N; если не указано, используется значение ttl для зоны.

29.4 Другие настройки

Данные настройки задаются на разных уровнях службы. Для каждой команды область, где может быть вызвана команда, указана в строке Область определения.

Для указания версии IP протокола, с которым службе следует работать при выполнении рекурсивных запросов, выполните:

mode <ipv6|ipv46|ipv4>

По умолчанию: mode ipv4 (поддержка только IPv4).

forwarders [{<IP[:PORT]>,3}] [first | only]

Определяет список ретрансляторов, т.е. серверов, которым мы переназначаем запросы ДНС.

Переназначенный ретранслятору запрос является рекурсивным, т.е. ожидается, что ретранслятор вернет окончательный ответ.

По умолчанию,если это не изменено настройками сервиса,запросы серверам, не являющимся ретрансляторами, будут итеративными, т.е. такой сервер ДНС сам будет производить поиск окончательного ответа.

Данная опция может быть в любом типе зоны.

Если запрос попадает в авторитетную зону (мастер или слэйв) настроенную на ретрансляцию:

- сначала ответ ищется в кэше или данных зоны;
- если ответ не найден - посылается рекурсивный запрос ретранслятору;
- если ответ от ретранслятора не пришёл - сервером начинается самостоятельный итеративный поиск ответа.

Существует специальный тип зоны, форвард-зона или зона ретрансляции, все запросы по которой сразу же обслуживаются указанными в данной зоне ретрансляторами.

Список ретрансляторов может быть пустым. Это полезно,если набор ретрансляторов установлен для всех зон глобально, но для одной из зон использовать ретрансляторы не нужно.

Параметры:

- IP:port - адрес ретранслятора и порт(необязательно), на который следует пересылать запрос;
- first: если ответа в данных зоны нет, следует пересылать запрос ретрансляторам и, если ответа нет, попытаться самостоятельно разрешить запрос (по умолчанию);
- only: если ответа в данных зоны нет, то для разрешения запроса следует использовать только ретрансляторы.

Область определения: служба,вид,зона ретрансляции.

notify <yes | no | master-only | explicit> [also-notify {<IP:PORT>,3}]

Определяет, нужно ли слать нотификацию (сообщение DNS NOTIFY) вторичным серверам зоны, для которой изменился идентификатор (поменялся ее SOA ID).

Вторичные сервера зоны выбираются из NS-записей зоны.

Дополнительно, параметром also-notify можно указать адреса и порты дополнительных серверов, которым следует слать нотификации.

В случае указания опции explicit, нотификации не шлются никому, кроме списка серверов, указанных параметром also-notify.

Область определения: служба,вид,мастер-зона.

По умолчанию: notify yes.

query-source [IP] [PORT_START [PORT_END]]

Определяет адрес и порт сервера для посылки запросов другим DNS-серверам.

Эта опция используется, если DNS-сервер должен работать с определённым локальным сетевым интерфейсом для отправки запросов, в случае, например, если один из основных DNS-серверов опознает лишь один из его многочисленных адресов.

Указанный IP-адрес будет использован и для TCP-,и для UDP-запросов.

Указанный порт(порты) будет использован только для UDP-запросов, а для TCP будет выбран случайный непривилегированный порт (>1024).

Предупреждение: небезопасно назначать фиксированный порт, т.к. злоумышленник может угадать 16-битный DNS Transaction ID и замусорить кэш сервера своими неадекватными ответами. В случае же случайного порта - ему необходимо угадать два 16-битных числа (порт и id транзакции), что гораздо сложнее.

Область определения: служба.

По умолчанию: query-source 0.0.0.0 и случайный непривилегированный порт (номер порта больше 1000)

Для работы с IPv6 адресами используйте команды: query-source6 и no query-source6.

listen [PORT] [IPLIST]

Установка номера порта и IP-адресов, на которых будет слушать ДНС-сервер.

Область определения: служба.

По умолчанию: listen 53 localips.

Для работы с IPv6 адресами используйте команды: listen6 и no listen6.

recursion <yes | no >

Нужно ли обслуживать рекурсивные запросы?

Область определения: служба,вид.

По умолчанию: recursion yes.

filter-aaaa

Удалять все AAAA-записи из ответа, предназначенного для IPv4-клиента (запрос в DNS-службу пришел с IPv4-адреса).

check-names <ignore | warn | fail >

Команда задает, как именно реагировать на неверные с точки зрения формата имена зоны и доменные имена. Возможные варианты:

- ignore - игнорировать неверные имена
- warn - предупреждать о неверных именах (по-умолчанию для слэйв-зоны и для мастер-зоны, если это специальная Windows мастер-зона (содержит "_" в своем имени))
- fail - не разрешать неверные имена (по-умолчанию для мастер-зоны)

Область определения: прямая зона, слэйв-зона.

29.5 DNSSEC

Служба поддерживает DNSSEC - расширение протокола DNS для обеспечения безопасности. Это позволяет обеспечить подлинность и целостность DNS-данных, что в свою очередь позволяет защититься от кибер-атак типа загрязнения кэша DNS-серверов, перенаправления или подмены DNS-запросов.

DNSSEC использует криптографию с открытым ключом (PKI): добавляет цифровую подпись в данные DNS-зоны таким образом, что DNS-ответы могут быть проверены на целостность (сообщение не изменено во время пересылки) и подлинность (информация пришла от того, от кого ее запрашивали, т.е. от надежного источника).

DNSSEC не обеспечивает защищенный туннель передачи данных, не шифрует DNS-данные.

Рассмотрим три главных компонента инфраструктуры DNSSEC:

- валидирующий резолвер (ВР) - рекурсивный DNS-сервер с возможностью проверки подлинности ответов при рекурсивном поиске доменного имени (посредством DNSSEC);
- авторитетный DNS-сервер - сервер, подписывающий данные обслуживаемых им зон; это включает в себя создание специальных ресурсных записей для подписи и публикации их на родительских DNS-серверах, если это необходимо; такой сервер при ответе на DNS-запрос будет возвращать помимо обычных ресурсных записей, также цифровые подписи и DNSSEC-ключи в виде ресурсных записей;
- клиентское приложение, осуществляющее DNS-запрос

DNSSEC вводит несколько новых типов ресурсных записей:

- RRSIG - подпись ресурсной записи, в службе существует одна подпись для каждой ресурсной записи зоны; ВР используют RSIG-записи для проверки получаемых ответов;

- DNSKEY - публикуемые открытые ключи, закрытые ключи не публикуются и хранятся на авторитетном DNS-сервере; существует два типа таких ключей: ZSK-ключ для подписи ресурсных записей зоны и KSK-ключ для подписи самих ключей (как ZSK, так и KSK);

- NSEC - от англ. Next Secure обеспечивает аутентифицированный отказ существования для набора ресурсных записей DNS; запись содержит цепочку авторизованных и упорядоченных имен зоны, а также их типов, это позволяет ВР аутентифицировать отрицательный отклик (NXDOMAIN) для несуществующего имени или типа; за счет упорядоченности имен цепочка NSEC определяет пустые промежутки между именами, что позволяет дать быстрый ответ NXDOMAIN для несуществующего имени;

- NSEC3 - запись NSEC позволяла получить записи зоны путем зонной нумерации; эта уязвимость преодолена в NSEC3 записи за счет хеширования имен зоны в цепочке NSEC3;

- DS - ресурсная запись, генерируемая из открытого KSK-ключа зоны и публикуемая в родительской зоне; это нужно для построения цепочки доверия, когда родительская зона как бы поручается за все свои подчиненные зоны;

Поскольку DNSSEC основывается на PKI, то по отношению к ключам подписи записей зоны верны следующие особенности PKI:

- закрытая часть ключа используется для подписи DNS-записей зоны

- открытая часть ключа используется для верификации DNS-записей, полученных в ответе DNS-сервера.

Таким образом, обобщим: ресурсные записи зоны, кроме DNSKEY записей, подписываются закрытой частью ZSK-ключа, открытые части ключей ZSK и KSK публикуются в зонном файле в виде записей DNSKEY и подписываются закрытой частью KSK-ключа. Кроме того, из открытой части KSK-ключа формируется DS-запись, которую можно передать DNS-серверу, обслуживающему родительскую зону, для формирования цепочки доверия. Цепочка доверия идет от корневой зоны вниз к подчиненным зонам.

Рассмотрим подробнее ключи KSK и ZSK. У ключей KSK и ZSK существует такая характеристика, как временная метка. Виды временных меток:

- create: время создания (генерации) ключа;
- publish: время публикации ключа; после этого времени ключ будет включен в зону, но не будет использоваться для подписи; это уведомляет ВР о готовности использовать новый ключ; по умолчанию это время равно create-времени;
- activate: время активации ключа; после этого времени ключ будет оставаться в зоне и будет использоваться для подписи; по умолчанию это время равно create-времени;

- revoke: время отзыва ключа; после этого времени в ключе будет установлен флаг отзыва; ключ будет оставаться в зоне и будет использоваться для подписи; это уведомляет ВР о скором удалении ключа;
- inactive: время деактивации ключа; после этого времени ключ будет оставаться в зоне, но не будет использоваться для подписи; это есть время “срока годности” ключа;

- delete: время удаления ключа; после этого времени ключ не будет включен в зону, но будет оставаться в виде файла в файловой системе и может быть удален.

Таким образом, подытожим:

- временные метки, находясь внутри которых ключ **не находится** в зоне и **не подписывает** ее данные: create, delete;

- временные метки, находясь внутри которых ключ **находится** в зоне, но **не подписывает** ее данные: publish, inactive;

- временные метки, находясь внутри которых ключ **находится** в зоне и **подписывает** ее данные: activate, revoke.

В службе dns часть операций с DNSSEC осуществляются автоматически, а часть в ручном режиме. Ручные операции:

- создание ключей ZSK, KSK
- установка временных меток ключей ZSK, KSK
- получение DS-записи из ключа KSK и регистрация ее в родительской зоне
- ротация ключей ZSK, KSK

Автоматические операции:

- подписывание зоны
- ротация ключей ZSK, KSK

Исходя из вышеописанных особенностей для работы с DNSSEC администратору необходимо уметь осуществлять следующие основные операции:

- создавать ключи KSK и ZSK
- подписывать зоны
- передавать информацию о ключах KSK регистратору домена, т.е. как минимум уметь создавать DS-запись из KSK-ключа
- осуществлять периодическую ротацию ключей KSK и ZSK

Для обеспечения вышеописанных операций в службе dns существует набор команд, который будет рассмотрен в следующих подразделах.

29.5.1 Включение режима DNSSEC

Включить или выключить поддержку обработки DNSSEC-запросов для зоны или вида:

dnssec <enable | disable>

В результате служба сможет отвечать на DNSSEC-запросы DNSSEC-клиентов.

По умолчанию: enable.

Включить или выключить возможность проверки DNSSEC-ответов других DNS-серверов:

dnssec-validation <on | off>

В случае включения, служба будет использовать встроенный ключ (якорь доверия) корневой зоны и его автоматическое обновление согласно RFC-5011.

Для проверки правильности валидации DNSSEC службой выполните команду, например для ОС на базе Linux:

```
dig @DNS_SERVICE_IP servfail.nl
```

Где DNS_SERVICE_IP - IP-адрес службы DNS. В результате работы команды должен вернуться статус SERVFAIL. Если вернулся статус NOERROR, значит валидация работает не корректно.

По умолчанию: off

29.5.2 Работа с ключами

Ключи в системе идентифицируются по идентификатору KEYID, формат которого следующий: <ALG_NAME>-<ID>, где ALG_NAME - крипто-алгоритм ключа, например RSASHA1, ID - уникальный 16-битный идентификатор ключа, создаваемый службой в процессе генерации ключа.

Создание ключа ZSK или KSK:

service dns sec key gen <VIEW> <ZONE> <ksk|zsk> [ALGO | nsec-algo <ALGO_NSEC>] [TTL]

Обычно данную команду следует использовать при первичной инициализации DNSSEC.

Службой установлено максимальное число ключей каждого типа для каждой зоны: 5.

Параметры:

- VIEW, ZONE - имя вида и имя зоны, для которой создается ключ
- ksk, zsk - тип ключа: KSK или ZSK
- ALGO - криптографический алгоритм ключа, по умолчанию RSASHA1
- ALGO_NSEC - криптографический алгоритм ключа с поддержкой NSEC3, по умолчанию NSEC3RSASHA1

- TTL - время жизни записи DNSKEY, формируемой из ключа службой при создании зонного файла

Данная команда только создает ключ, подписания зоны не происходит. После создания ключа он попадает в файловую систему в виде файла - в репозиторий ключей. Временные метки по умолчанию установлены следующие: create, publish, acivate, они одинаковые и равны времени генерации ключа. Таким образом, ключ будет использован для подписи сразу же, при включении службы. Однако, это можно изменить,если необходимо, командой service dns sec key time.

Установка временной метки ключа:

service dns sec key time <VIEW> <ZONE> <ksk|zsk> <KEYID> [TIME_TYPE <TIME>]{0,5}

Параметры:

- VIEW, ZONE - имя вида и имя зоны
- ksk, zsk - тип ключа: KSK или ZSK
- KEYID - идентификатор ключа
- TIME_TYPE - одна из временных меток (create, publish, acivate, revoke, inactive, delete)
- TIME - указание времени действия в одном из следующих форматах:
  - YYYYMMDD - где YYYY - год, MM - месяц, DD - день
  - YYYYMMDDHHMMSS - как и выше, но добавляется HH - часы, MM - минуты, SS - секунды
  - <+->N[y|mo|w|d|h|mi] - приставка "+" или "-" задает смещение по времени в будущее или прошлое соответственно, N - число единиц времени, суффикс задает тип единицы времени: y - год, mo - месяц, w - неделя, d - день, h - час, mi - минута; если суффикс не задан, то секунда.

Если временная метка не задана, будут выведены все временные метки ключа.

Создание ключа-преемника ZSK или KSK на основе имеющегося ключа ZSK или KSK:

service dns sec key <VIEW> <ZONE> <ksk|zsk> successor <KEYID>

Данную команду обычно следует использовать при процедуре ротации ключей (см. ниже). Она создает новый ключ с такими же характеристиками, как и уже имеющийся ключ с идентификатором KEYID.

Параметры:

- VIEW, ZONE - имя вида и имя зоны
- ksk, zsk - тип ключа: KSK или ZSK
- KEYID - идентификатор ключа-родителя

Алгоритм, размер и тип ключа сохраняются. Метка activate ключа-преемника будет установлена в метку inactive ключа-родителя (ключ KEYID), метка publish ключа-преемника будет установлена в метку activate ключа-преемника за минусом 30 дней (т.е. от публикации до активации по умолчанию 1 месяц).

Генерация DS-записи из подписанной зоны или из KSK-ключа.

service dns sec ds <VIEW> <ZONE> [KEYID] [ALGO] [FILE]

Параметры:

- VIEW, ZONE - имя вида и имя зоны
- KEYID - идентификатор KSK-ключа; если он не задан - DS-запись будет подсчитана на основе зонного файла, если он задан - DS-запись будет подсчитана на основе этого ключа
- ALGO - криптографический алгоритм ключа, по умолчанию SHA1
- FILE - файл, куда может быть записана DS-запись

Пример выдачи данной команды:

```
Info: int. IN DS 32143 5 1 C7B3FC69A0C28E66D67E4645EB52098D438CDA22
Info: Crypto—algorithm type: RSASHA1 (5)
Info: DIgest type: SHA1 (1)
```

Отметим следующие поля записи DS:

- 32143 - это ID ключа KSK
- 5 - номер крипто-алгоритма ключа RSASHA1
- 1 - номер крипто-алгоритма отпечатка ключа SHA1

Как можно видеть, пояснения к этим полям даны во 2й и 3й строке вывода команды.

Вывод информации о подписанной зоне:

service dns sec verify <VIEW> <ZONE>

```
Loading zone ’int.’
Verifying the zone using the following algorithms: RSASHA1.
Zone fully signed:
Algorithm: RSASHA1: KSKs: 1 active, 0 stand—by, 0 revoked
                    ZSKs: 1 active, 0 stand—by, 0 revoked
```

Как можно видеть, в выводе описано следующее: крипто-алогритм ключей подписи RSHSHA1, количество и состояние ключей KSK и ZSK: активен, приостановлен, отозван.

Удаление ключа:

service dns sec key remove <VIEW> <ZONE> <ksk|zsk> <KEYID>

Параметры:

- VIEW, ZONE - имя вида и имя зоны
- ksk, zsk - тип ключа: KSK или ZSK
- KEYID - идентификатор ключа

НЕ СЛЕДУЕТ производить удаление ключа, который в настоящее время используется зоной, т.е. прописан в команде sign или new-zsk.

Вывод информации об имеющихся ключах в репозитории ключей:

show service dns sec key <VIEW> <ZONE> [TIME_TYPE] [brief]

Данная команда выводит информацию о временных метках ключей KSK и ZSK, по умолчанию ключи отсортированы в порядке возрастания create-метки - т.е. по времени создания ключа.

Параметры:

- VIEW, ZONE - имя вида и имя зоны
- TIME_TYPE - временная метка, по которой производить сортировку вывода, если не задана,то create-метка
- brief - краткая таблица, выводит не все временные метки, а только activate,inactive и deleted.

29.5.3 Подписывание зоны

Для подписывания зоны введите в master- или slave-зоне команду:

sign <KSK> <ZSK> [nsec3] [rsig-expire TTL] [key-check INT]

Параметры:

- KSK, ZSK - задает соответственно ключ KSK и ZSK для подписи
- TTL - время жизни записи RSIG, которая есть подпись для каждой из ресурсных записей зоны , по умолчанию раз в 30 дней
- INT - интервал сканирования репозитория ключей на предмет наличия новых ключей или изменения временных меток ключей, по умолчанию раз в 60 минут
- nsec3 - указывает создать NSEC3-записи в зонном файле

При задании данной команды и включении службы dns происходит автоматическое подписывание зоны, а также все дальнейшее DNSSEC-операции с данной зоной происходят автоматически: создание подписей для ресурсных записей зоны, удаление из зоны ключей и добавление новых (см. Ротация ключей)

Добавить второй (новый) ключ ZSK:

new-zsk <ZSK>

Параметры:

- ZSK - задает ключ ZSK, обычно это ключ-преемник для существующего ZSK-ключа из команды sign.

Эту команду можно использовать при ротации ZSK-ключей по методу двойной подписи.

29.5.4 Работа с доверенными якорями (ключами) корневой зоны.

show service dns sec root-keys [dump]

Данная команда выводит информацию о текущих используемых службой якорях доверия корневой зоны (по сути это KSK-ключи корневой зоны). Эти ключи автоматически поддерживаются службой и начинают цепочку доверия при проверке DS записей удаленных DNS-серверов.

service dns sec root-keys update

Данная команда проверяет якорь доверия корневой зоны на наличие более свежих ключей и обновляет их при необходимости. Это же делается при включении службы, если задана команда 'dnssec-validation on'.

29.5.5 Ротация ключа ZSK.

29.5.5.1 Метод предварительной публикации.

Предположим, что в начале использования ключа, мы задали время деактивации и удаления существующего ключа - через 5 и через 6 месяцев соответственно:

```
service dns sec key time default int. zsk RSASHA1—47355 inactive +5mo delete +6mo
```

Сразу же или за месяц до деактивации ключа создадим ключ-преемник:

```
service dns sec key gen default int. zsk successor RSASHA1—47355
```

Ключ преемник будет активирован через 5 месяцев (когда старый ключ будет деактивирован).

Соответственно после момента ротации ключей, т.е. после активации нового ключа, можно будет заменить команду sign указав новый ZSK-ключ.

29.5.5.2 Метод двойной подписи

Предположим, что в начале использования ключа, мы задали время деактивации и удаления существующего ключа - через 5 и через 6 месяцев соответственно:

```
service dns sec key time default int. zsk RSASHA1—47355 inactive +5mo delete +6mo
```

Сразу же или за месяц до деактивации ключа создадим ключ-преемник:

```
service dns sec key gen default int. zsk successor RSASHA1—47355
```

Ключ преемник будет активирован через 5 месяцев (когда старый ключ будет деактивирован).

Пропишем ключ-преемник, в результате чего размер зонного файла будет больше чем в предыдущем варианте:

```
new—zsk RSASHA1—12345
```

Соответственно после момента ротации ключей, т.е. после активации нового ключа, можно будет удалить команду new-zsk и заменить команду sign указав новый ZSK-ключ (RSASHA1-12345).

После ротации ZSK-ключей можно удалить старый ключ командой

```
service dns sec key remove
```

29.6 Динамическое обновление зон

Для мастер-зоны можно использовать динамическое обновление. Динамические обновления позволяют локальной или удаленной службе DHCP обновлять A- и PTR-записи в мастер-зоне. Динамическое обновление зоны может быть двух типов:

- локальное: осуществляется в связке со службой DHCP, работающей на той же машине, что и настраиваемая DNS-служба;
- удаленное: осуществляется любым удаленной DHCP-службой Dionis-NX, отдельным DHCP-сервером или другим удаленным ПО, которое может осуществлять динамические DNS-обновления.

В качестве удаленного DHCP-сервера наиболее совместимым с Dionis-NX службами является сервер ISC DHCP.

В обновляемых зонах могут быть различные статические записи.

Если все настроено правильно, то при выдаче службой DHCP IP-адреса из подсети, которой соответствует настроенная на обновление обратная зона в службе DNS, и доменное имя данной подсети соответствует настроенной прямой зоне, будет происходить автоматическое обновление A- и PTR- ресурсных записей.

29.6.1 Локальное динамическое обновление

Для настройки локального автоматического обновления необходимо:

- сначала настроить DNS-службу:
  - настроить прямую и обратную зоны, подлежащие авто-обновлению;

- если есть опция allow query для зоны, подлежащей обновлению, среди аргументов этой команды должен быть localhost;
- предусмотреть,чтобы динамические обновления попали в нужный вид (тот, в котором содержится обновляемая зона):
  - в виде, в котором находится обновляемая зона, среди аргументов команды match clients должен быть localhost;
  - в видах, которые расположены выше вида, в котором находится обновляемая зона, среди аргументов команды match clients не должно быть localhost;
- добавить в прямую и обратную обновляемые зоны команду update;

- затем настроить DHCP-службу:

- настроить интервал раздаваемых адресов (range) таким образом, чтобы он включал в себя подсеть обновляемой обратной зоны, либо был равен этой подсети.
- указать domain-name для подсети, либо глобально;
- запустить или перезапустить DHCP-службу.

Пример:

```
#отрывок DNS—конфигурации:
 view v1
  match clients localhost 192.168.33.0/24
  zone master 33.168.192.in—addr.arpa.
   update
   soa master raul.cuba.int.
   ns raul.cuba.int.
   ptr 254 raul.cuba.int.

  zone master cuba.int.
   update
   soa master raul
   a 192.168.33.254 domain raul
   ns raul

 view default
  ...

#отрывок DHCP—конфигурации:
 domain—name cuba.int
 subnet 192.168.33.0/24
  range 192.168.33.10 192.168.33.200
```

29.6.2 Динамическое обновление удаленной DHCP-службой Dionis-NX (аутентификация по общему ключу)

Для настройки удаленного автоматического обновления DNS удаленной DHCP-службой Dionis-NX необходимо:

- сначала настроить локальную DNS-службу:
  - настроить прямую и обратную зоны, подлежащие авто-обновлению;
  - если есть опция allow query для зоны, подлежащей обновлению, то в список IP-адресов этой команды должен входить адрес удаленной DHCP-службы;
  - предусмотреть,чтобы динамические обновления попали в нужный вид (тот, в котором содержится обновляемая зона):
    * в виде, в котором находится обновляемая зона, в списке IP-адресов команды **match clients** должен быть адрес удаленной DHCP-службы;
    * в видах, которые расположены выше вида, в котором находится обновляемая зона, в списке IP-адресов команды **match clients** не должен быть адрес удаленной DHCP-службы;
  - добавить в обновляемую прямую и обратную зоны команду **update password PAS**, где PAS - некоторая строка-пароль, которые могут быть разными для этих зон.
- затем настроить удаленную DHCP-службу:
  - настроить интервал раздаваемых адресов (range)таким образом, чтобы он включал в себя подсеть обновляемой обратной зоны, либо был равен этой подсети;
  - указать domain-name для подсети, либо глобально;
  - добавить команду **update** <DNSIP> <FZONE> <PAS>, где DNSIP - IP-адрес DNS-сервера, FZONE - имя прямой зоны, PAS - тот же пароль,что указан в опциях команды **update** прямой зоны в DNS-службе;
  - добавить команду **update** <DNSIP> <RZONE> <PAS>, где DNSIP - IP-адрес DNS-сервера, RZONE - имя обратной зоны, PAS - тот же пароль,что указан в опциях команды update обратной зоны в DNS-службе;
  - запустить или перезапустить DHCP-службу.

Пример:

```
#отрывок DNS—конфигурации:
 view v1
  match clients localhost 192.168.33.0/24
  zone master 33.168.192.in—addr.arpa.
   update password pas
   soa master raul.cuba.int.
   ns raul.cuba.int.
   ptr 254 raul.cuba.int.

  zone master cuba.int.
   update password pas
```

```
   soa master raul
   a 192.168.33.254 domain raul
   ns raul

 view default
  ...

#отрывок DHCP—конфигурации удаленной DHCP—службы:
 domain—name cuba.int
 update 192.168.33.254 cuba.int. pas
 update 192.168.33.254 33.168.192.in—addr.arpa. pas
 subnet 192.168.33.0/24
  range 192.168.33.10 192.168.33.200
```

29.6.3 Динамическое обновление удаленным DHCP-сервером (аутентификация по общему ключу)

Для настройки удаленного автоматического обновления DNS удаленным DHCP-сервером, либо другим ПО, которое может осуществлять удаленные DNS-обновления необходимо:

- сначала настроить локальную DNS-службу: осуществляется аналогично настройке удаленного автоматического обновления DNS удаленной DHCP-службой Dionis-NX (см. предыдущий раздел);
- после выполнения команды update PAS на экран будет выведена информация об общем ключе и его параметрах, например: key(name: KEYNAME,secret: SECRET);
- использовать эту информацию для настройки ISC DHCP-сервера; конфигурация общего ключа в DHCP-сервере будет в следующей форме:

Конфигурация общего ключа:

```
key ”KEYNAME” {
algorithm hmac—md5;
secret ”SECRET”;
};
```

Пример:

```
#1. отрывок DNS—конфигурации:
 view v1
  match clients localhost 192.168.33.0/24
  zone master 33.168.192.in—addr.arpa.
   update password pas
   soa master raul.cuba.int.
   ns raul.cuba.int.
   ptr 254 raul.cuba.int.
```

```
  zone master cuba.int.
   update password pas
   soa master raul
   a 192.168.33.254 domain raul
   ns raul

 view default
  ...

#2. После выполнения команды update pas на экран будет выведено:
  * для прямой зоны:    To update A RRs by remote dhcp servers use key   (name:pas,key:cGFz)
  * для обратной зоны: To update PTR RRs by remote dhcp servers use key (name:pas,key:cGFz)

#3. В конфигурацию удаленного ISC DHCP сервера добавьте следующие строки:


ddns—update—style interim;

key pas {
  algorithm hmac—md5;
  secret ”cGFz”;
}

zone cuba.int. {
  primary 192.168.33.254;
  key pas;
}
zone 33.168.192.in—addr.arpa. {
  primary 192.168.33.254;
  key pas;
}
```

29.6.4 Динамическое обновление удаленным DHCP-сервером (аутентификация по IP-адресу)

Можно настроить принятие динамический обовлений от любого DHCP-сервера в зависимости от его IP-адреса. Это менее безопасный вариант (IP-адрес можно подменить), чем вариант обновления по общему ключу, однако более простой в настройке и единственный для принятия обновлений от Windows DHCP-сервера.

Для настройки обоновляемой зоны с аутентификацией по IP-адресу необходимо добавить в нее команду:

```
update acl ACL1 [ ACL2  ACL3] ...
```

Где ACL1, ACL2, ACL3 задают до 3х штук ранее определенных ACL (набор IP-адресов и/или сетей) .

Обновления с IP-адресов, которые входят в указанные ACL, будут приниматься службой DNS.

Причем нет возможности указать, какие именно записи разрешено обновлять указанным ACL - могут быть обновлены записи любого типа в зоне.

29.7 Ограничения ресурсов службы

Значения указываются в байтах.

limit journal-size <N>

Устанавливает максимальный размер отдельного файла журнала.

Файл журнала автоматически создается для динамически обновляемой зоны (см. команду update в конфигурировании мастер-зоны) и содержит информацию об обновлении зоны в бинарном формате.

Информация из журнала автоматически переносится в зонный файл,а журнал удаляется в следующих случаях:

- автоматически примерно каждые 15 минут;
- по команде перезагрузки зон: «service dns relod zones <all | dynamic>»;
- по команде запуска сервиса: enable.

При достижении указанного максимального размера файла самые старые транзакции журнала удаляются автоматически.

Умолчание: неограниченно.

limit recursive-clients <N>

Максимальное количество одновременно выполняемых рекурсивных запросов, поступивших от клиентов.

Для справки: каждый рекурсивный запрос потребляет примерно 20Кб памяти.

Умолчание: 1000.

limit tcp-clients <N>

Максимальное количество TCP-соединений, поддерживаемых сервером в каждый момент времени.

Умолчание: 100.

limit cache-size <N>

Максимальный объем памяти, отводимой под кэш DNS-сервера для каждого вида,в байтах.

Когда объем данных кэша достигает этого предела, DNS-сервер принудительно удаляет записи из кэша.

Умолчание: неограниченно.

limit tcp-listen-queue <N>

Устанавливает число (не меньше 3) TCP-соединений в ядре системы, ожидающих передачи данных.

Умолчание: 3.

limit max-cache-ttl<N>

Устанавливает TTL хранения ответов в кэше.

Умолчание: 7 дней.

limit max-ncache-ttl<N>

Устанавливает TTL хранения отрицательных ответов в кэше.

Умолчание: 3 часа.

limit edns-size <MAX_ADV> [MAX_REP]

Задает лимиты размеров сообщений EDNS. MAX_ADV - максимальный согласуемый размер буфера для приема UDP EDNS-сообщения. MAX_REP - максимальный размер UDP EDNS-сообщения ответа. По умолчанию выбраны значения 4096 для обоих лимитов.

29.8 Журналы

log <TYPE> <LEVEL>

Определяет тип и уровень журналирования.

Параметр TYPE задает тип информации для журналирования:

- other : остальная информация, не входящая в следующие типы;
- config : информация, связанная с внутренним конфигурационным файлом сервера;
- queries : информация, связанная с DNS-запросами серверу;
- transfer : информация, связанная с передачей зоны;
- update : информация, связанная с обновлением зон.

Параметр LEVEL задает уровень подробности журналируемой информации:

- none : ведение журнала отключено;
- critical : вести журнал только критических ошибок;
- error : вести журнал обычных ошибок и более серьезных;
- warning : вести журнал предупреждений и более серьезных событий;
- notice : вести журнал замечаний и более серьезных событий;
- info : вести журнал информационных сообщений и более серьезных событий;
- debug <N> : вести журнал отладочных сообщений и более серьезных событий; N - уровень отладки от 1 до 3х.

log all <LEVEL>

Команда аналогична **log <TYPE>** за исключением того, что менее приоритетна и задает уровень подробности журналирования для всех типов информации.

В результате, можно переопределить уровень подробности журналирования для отдельных типов информации командами **log <TYPE>**.

29.9 Диагностика

Рассмотрим команду режима enable для диагностики работы службы DNS.

nslookup <DOMAIN> [TYPE] [rr <RRTYPE>] server <SERVER>

Команда осуществляет DNS-запрос.

Параметры:

- DOMAIN : IP-адрес, либо абсолютное доменное имя хоста, т.к. команда не использует ip resolver search/domain-name;
- TYPE : рекурсивный (recursive) или нерекурсивный (non-recursive) запрос; по умолчанию - рекурсивный;
- RRTYPE : тип ресурсной записи (mx,ptr,cname,a,soa,ns); по умолчанию - все типы ресурсных записей;
- SERVER : доменное имя или IP адрес сервера, на который посылать запрос; по умолчанию - серверы, указанные в ip resolver nameserver;
- PORT : порт, на который посылать запрос; по умолчанию - 53.

Формат вывода команды (рассмотрим пример):

```
;; —>>HEADER<<— opcode: QUERY, status: NOERROR, id: 25210
;; flags: qr rd ra; QUERY: 1, ANSWER: 8, AUTHORITY: 2, ADDITIONAL: 1

;; QUESTION SECTION:
;ya.ru.              IN  A

;; ANSWER SECTION:
ya.ru.          3259    IN  A   87.250.250.203
...
...

;; AUTHORITY SECTION:
ya.ru.          3233    IN  NS  ns5.yandex.ru.
...
...

;; ADDITIONAL SECTION:
ns1.yandex.ru.      124 IN  AAAA    2a02:6b8::1
```

```
...
...

;; Query time: 1 msec
;; SERVER: 192.168.33.254#53(192.168.33.254)
;; WHEN: Fri Dec  2 15:59:48 2011
;; MSG SIZE  rcvd: 222
```

Рассмотрим более подробно,что означают эти данные в выводе:

- **opcode** : тип операции (QUERY - запрос);
- **status** : статус операции:
  - NO ERROR - нет ошибок;
  - SERVER FAILURE - ошибка сервера;
  - NXDOMAIN - ошибка в имени, нет такого имени;
  - NOT IMPLEMENTED - реализация отсутствует;
  - REFUSED - отказ;
- **id** : 16-битный номер ДНС-транзакции,нужный для связи запросов и ответов;
- **flags** : дополнительные сведения об ответе, могут быть следующими:
  - qr - сообщение является ответом, а не запросом; присутствует всегда;
  - aa - авторитетный ответ;
  - rd - послано требование обслужить наш запрос рекурсивно;
  - ra - требование рекурсии удовлетворено, т.к. рекурсивные запросы разрешены на сервере имен.
- **Число записей**:
  - QUERY : число записей в разделе запроса;
  - ANSWER : число записей в разделе ответа;
  - AUTHORITY : число записей в разделе авторитета;
  - ADDITIONAL : число записей в дополнительном разделе;
- **Записи**:
  - QUESTION SECTION : раздел запроса, содержит записи, которые необходимо получить от сервера;
  - ANSWER SECTION : раздел ответа, содержит записи, полученные в ответе;
  - AUTHORITY SECTION : раздел авторитета, содержит имена серверов, авторитетных для запрашиваемого домена;
  - ADDITIONAL SECTION : дополнительный раздел, содержит дополнительные записи, по смыслу связанные с ответной информацией; например A-записи для возвращаемых серверов имен;
- **Дополнительная информация о запросе**:
  - Query time : через сколько времени после отправки запроса удалённый сервер имен вернул ответ;

- SERVER : адрес и порт сервера имен, через который отправлен запрос;
- WHEN : дата и время получения ответа;
- MSG SIZE : размер сообщения запроса(sent) и/или ответа(rcvd) в байтах.

29.10 Работа со службой

В этом подразделе рассматриваются команды режима enable для работы со службой DNS.

29.10.1 Команды управления сервисом

service dns reload

Эта команда может быть полезна если конфигурация службы не поменялась, однако вы хотите чтобы служба заново считала зонные файлы, например потому, что вы поменяли их командой режима enable (например, командой service dns remove dynamic-rrs).

service dns restart

Полный перезапуск службы.

29.10.2 Команды просмотра данных

show service dns cache <only | zones | all> [VIEW] [domain <NAME>]

Показать данные кэша ДНС для вида VIEW по доменному имени NAME.

Если указан параметр NAME - будут показаны только те записи, в которых присутствует домен NAME, иначе - все записи.

Кэш показывается в формате зонных файлов.

Если вид не задан - показываются данные для всех видов.

- only: показать данные кэша;
- zones: показать данные зон, для которых сервер является авторитетным;
- all: показать данные кэша и зон.

show service dns dynamic-rrs [VIEW]

Показывает динамические записи (прямой и обратной зон) вида VIEW.

show service dns log <all | queries | config | transfer | update | other> [all | number <N> | archive <N>] [archive <N>] [follow]

Показывает записи журналов. Типы журналов:

- queries : журналы ДНС-запросов;
- config : журналы работы с внутренним конфигурационным файлом;
- transfer : журналы передачи зон;
- update : журналы обновления;
- other : журналы другой информации,не вошедший в предыдущие типы;
- all : все журналы.

Число записей:

- all : все записи;
- number N : N записей;
- archive N : записи архива журналов под номером N.

Порядок отображения:

- follow: показывать записи журналов по мере их поступления.

show service dns statistic

Показывает статистику ДНС (число запросов, обновлений, данные по сокетам, соединениям и т.д.).

show service dns status

Показывает текущий статус сервиса ДНС (число потоков, число зон, число клиентов в текущий момент и т.д.).

show service dns zones <all | static | dynamic> [VIEW [ZONE]]

Показывает зонные данные статических или динамических зон.

В качестве параметров можно указать вид и зону.

29.10.3 Команды удаления данных

service dns remove cache [view VIEW] [name NAME [all]]

Удаляет данные для имени NAME из кэша ДНС. Если указан параметр all - будут удалены все записи, в которых onwer имеет вид *.NAME, т.е. все поддомены домена NAME.

Если имя NAME не задано - удаляет весь кэш.

Если задан вид - удаляет только из этого вида.

service dns remove dynamic-rrs <VIEW> <FZONE> <RZONE> <NAME> <IP> [force]

Удаляет динамическую запись из прямой зоны FZONE и обратной зоны RZONE вида VIEW.

Запись определяется доменным именем NAME (полным или относительным) и IP-адресом IP.

Если зона статическая - запись удалена не будет, только если не будет указан параметр force. В этом случае запись, если она будет найдена, будет удалена.

30. Служба DHCP

Dionis-NX имеет службу DHCP, реализующую серверную часть протокола DHCP (Dynamic Host Configuration Protocol — протокол динамической конфигурации узла).

Протокол DHCP - это сетевой протокол прикладного уровня модели OSI, позволяющий компьютерам автоматически получать IP-адрес и другие параметры, необходимые для работы в сети TCP/IP. Данный протокол работает по модели «клиент-сервер». Для автоматической конфигурации клиент на этапе конфигурации сетевого устройства обращается к серверу DHCP и получает от него нужные параметры.

Основная концепция службы DHCP напоминает концепцию службы DNS - это иерархичность, вложенность и наследуемость конфигурационных пространств.

Иерархия следующая:

- service dhcp - служба DHCP (уровень А)
- subnet/host - подсеть/хост (уровень Б)

Самый верхний уровень конфигурации службы - это команды уровня service dhcp.

Многие из этих команд есть также на нижележащем уровне Б - в описании подсети или хоста.

Если на верхнем уровне (А) задана некоторая опция, то она определяется и для нижележащего уровня (Б). Далее можно изменить на значение данной опции на уровне Б и оно уже будет отличаться от все остальных.

Замечание по формату описания параметров: запись {<NAME>,3} - означает список максимум из трех параметров типа NAME

Чтобы настроить службу DHCP, следует войти в режим ее конфигурации:

service dhcp

Команда осуществляет вход в конфигурацию DHCP-службы.

30.1 Общие настройки службы

listen <ethernet | bond> <N>

Слушать запросы DHCP на Ethernet/Bond интерфейсе N.

Для задания нескольких интерфейсов нужно использовать данную команду для каждого интерфейса.

По умолчанию: слушать на всех широковещательных Ethernet-интерфейсах.

local-port <N>

Порт, на котором слушать DHCP-запросы.

По умолчанию: 67

local-address <IP>

IP-адрес, на котором слушать нешироковещательные DHCP-запросы на порт 67 (если иное не указано командой local-port).

Широковещательные запросы приниматься не будут.

Эта команда может быть полезна,если в связке со службой DHCP используется (на другом узле) служба DHCP-RELAY.

max-lease-time <N>

Максимальный срок аренды адреса (сек.).

Клиент в запросе может прислать желаемое максимальное значение срока аренды адреса.

Опцию можно указывать глобально, в настройках хоста и подсети.

При указании данной опции, клиент не может получить аренду адреса на большее время, чем указано в данной опции.

По умолчанию: 86400

min-lease-time <N>

Минимальный срок аренды адреса (сек.).

Клиент в запросе может прислать желаемое минимальное значение срока аренды адреса.

Опцию можно указывать глобально, в настройках хоста и подсети.

При указании данной опции, клиент не может получить аренду адреса на меньшее время, чем указано в данной опции.

По умолчанию: 300

default-lease-time <N>

срок аренды адреса по умолчанию.

Опцию можно указывать глобально, в настройках хоста и подсети.

Этот срок аренды адреса назначается клиенту, если он не прислал в запросе желаемое максимальное значение.

По умолчанию: 43200

respond-delay <N>

Задает число секунд (от 0 до 255) , в течение которых служба будет ждать,прежде чем ответить на запрос клиента.

Эта опция позволяет сделать службу DHCP дублирующей/запасной для основного DHCP-сервера на другом компьютере. Когда основной DHCP-сервер не отозвался, дублирующая служба DHCP будет отзываться после определенного количества запросов клиента, указанного параметром N как продолжительность ожидания перед ответом клиенту. Она может понадобиться для организации дублирующего DHCP-сервера на основе данной службы: если основной сервер DHCP не ответил в течении указанного числа секунд, то клиенту ответит дублирующая служба DHCP Dionis-NX.

По умолчанию: 0

send-hostname <on|off>

Это команда нужна, если в сети есть BOOTP-клиенты (бездисковые рабочие станции), которым необходимо помимо фиксированного адреса присваивать имя хоста.

Включить (on) или отключить (off).

Если опция включена, то для каждого объявления хоста, находящегося в зоне действия опции, имя , использованное в объявлении host (см. ниже подраздел “Настройка статического назначения”) передается клиенту в качестве его имени.

Опцию можно указывать глобально и в настройках хоста.

По умолчанию: отключено.

ddns-ttl <N>

Устанавливает умалчиваемое значение для TTL-динамических записей. Может понадобиться, если будут использоваться динамические DNS-обновления.

По умолчанию: определяется клиентом.

30.2 Настройка статического назначения

Обычно фиксированные адреса назначаются важным хостам сети,например серверам.

Рабочим станциям обычно назначают адрес динамически(см. подраздел “Настройка динамического назначения”).

host <NAME>

Вход в режим статического назначения IP-адреса для клиентов.

NAME - имя, идентифицирующее хост; используется только при включённой опции send-hostname: в этом случае оно передается клиенту в качестве его имени.

Переопределяется опцией host-name.

Имя хоста несущественно,если не включена(см.выше) опция send-hostname.

Следует также настроить признаки, по которым служба будет определять, принадлежит ли входящий DHCP-запрос данной host-конфигурации.

mac <MAC>

MAC - MAC-адрес клиента.

Наиболее часто используется привязка хост-декларации к MAC-адресу клиента.

client-id <NAME>

Помимо привязки к MAC-адресу, можно выполнить привязку хост-декларации к идентификатору клиента, который может передаваться в запросе (это т.н. dhcp-client-identifier DHCP-запроса).

NAME - идентификатор клиента.

В результате для поиска host-декларации, соответствующей клиенту,приславшему запрос, происходит следующее:

- сначала ищется client-id, совпадающий с dhcp-client-identifier, который прислан клиентом;
- если нужный client-id не находится, то ищется совпадение MAC-адреса в команде mac MAC-адресу клиента.

ip <IP>

Задает статический IP-адрес клиента.

30.3 Настройка динамического назначения

Для настройки динамического назначения адресов необходимо описать все сети,обслуживаемые интерфейсами системы, которые будут обслуживать DHCP-запросы.

У этих интерфейсов должны быть назначены IP-адреса с нужным масками сети.

subnet <IP/MASK>

Вход в режим config-service-dhcp-subnet-IP/MASK: динамическое назначение IP-адресов для клиентов.

range <IP_START> [IP_END]

Задает диапазон в сети SUBNET, адреса в пределах которого могут быть наначены клиентам.

Все IP-адреса в диапазоне должны принадлежать той подсети, к описанию которой относится секция range.

Если IP_END не указан, то диапазон состоит из одного адреса.

Пример:

```
(config—service—dhcp—subnet—192.168.1.0/24)# range 192.168.1.10 192.168.1.210
```

Этой командой мы предписываем службе назначать клиентам IP-адреса из указанного диапазона (всего 201 адрес).

30.4 Сетевые DHCP-опции

Помимо назначения клиентам IP-адресов, служба может передавать им другую конфигурационную сетевую информацию.

Эта информация называется сетевые DHCP-опции. Они могут быть указаны на любом из уровней службы: как глобально, так и в хост- или сетевой декларациях.

broadcast-address <IP>

Адрес для широковещательных запросов.

domain-name <NAME>

Имя домена для разрешения имен через DNS.

domain-search <NAME>

Имя домена для разрешения имен через DNS.

Можно указать несколько доменов.

gateway <IP>

Адрес шлюза. Можно задавать несколько шлюзов.

subnet-mask <IP>

Маска подсети.

Если не указана, значение маски берется из описания subnet, в которую попадает запрос.

[N] name-server <IP>

IP-адрес сервера имен с приоритетом N.

[N] ntp-server <IP>

IP-адрес сервера времени с приоритетом N.

[N] wins-server <IP>

IP-адрес WINS (NetBios) сервера с приоритетом N.

[N] smtp-server <IP>

IP-адрес сервера SMTP сервера с приоритетом N.

[N] route <NET> <IP>

Статический маршрут в сеть NET через шлюз с адресом IP.

30.5 Пользовательские DHCP-опции

Существует возможность создать пользовательские DHCP-опции, которые будут передаваться указанным клиентам так же, как и стандартные DHCP-опции.

Перед использованием пользовательской опции ее необходимо определить.

user-option-def <NAME> <CODE> <TYPE>

Данная команда определяет новую опцию с именем NAME, имеющую код CODE (от 128 до 254) и тип TYPE.

Код опции CODE принимает значения из интервала 128-254, т.к. все коды меньше 128 зарезервированы под стандартные DHCP-опции. На самом деле интервал от 128 до 224 так же зарезервирован под стандартные DHCP-опции, однако это произошло гораздо позже опубликования стандарта DHCP-протокола в соответствующем документе RFC и не все клиенты могу поддерживать данный стандарт и вполне могут использовать интервал 128-224 как пользовательский интервал опций. Поэтому, если это возможно, рекомендуется использовать интервал кодов 224-254. Интервал 128-224 оставлен для совместимости.

Тип TYPE определяет тип значений опции и может быть следующим:

- bool - задает булевый тип значения опции: on,off
- string - задает строковый тип значения опции: любая текстовая строка
- bytes - задает бинарный тип значения опции: последовательность байт длиной до 128, байты разделены символом »:».
- uint32 - задает целочисленный тип значения опции: любое 32-битное число
- ip - задает тип значения опции в виде IP-адреса.

user-option <NAME> <VAL>

Задает значение VAL ранее определенной опции NAME. Определение опции делается командой user-option-def.

30.6 Связь с DNS (динамическое обновление)

Служба может работать в связке с DNS службой по части динамического обновления DNS-записей.

Данная возможность описана в главе “Служба DNS” в пункте “Динамическое обновление зон”.

30.7 Работа со службой

30.7.1 Команды просмотра данных

show service dhcp log [all | N] [archive N] [follow]

Показывает журнал сервиса.

Параметры:

- N - число записей
- all - показать все записи
- follow - просмотр журналов по мере появления
- archive - просмотр старых журналов

По умолчанию: show service dhcp log 25

show service dhcp status

Показывает текущий статус службы DHCP (корректность внутренней конфигурации, корректность базы данных зон, состояние сервиса).

show service dhcp lease [all | active | free | IP]

Показывает данные по выданным адресам: всем, действующим, свободным или по указанному.

Формат выходных данных:

```
HOST:<HOST> (<STATUS>)
IP:<IP> MAC:<MAC> <DATE_S>—<DATE_E>
```

Значения полей:

HOST - имя хоста клиента,присланное им в DHCP-запросе(может отсутствовать, т.к. протокол DHCP не требует передавать его)

STATUS - статус адресной информации (free - свободная, active - занятая)

IP - арендованный IP-адрес

MAC - MAC-адрес клиента

DATE_S - время начала срока аренды адреса (формат YYYY/MM/DD/HH:MM:SS)

DATE_E - время конца срока аренды адреса (формат YYYY/MM/DD/HH:MM:SS)

По умолчанию: show service dhcp lease all.

30.7.2 Команды удаления данных

service dhcp remove lease [IP | DOMAIN]

Прекратить аренду адреса по указанному IP-адресу узла или его имени DOMAIN.

Если ничего не указано - прекращается аренда всех адресов.

30.8 Примеры

Рассмотрим пример:

- пусть в DHCP описано несколько сетей и хостов, и среди них имеется сеть 1.2.3.0/24 и хост zeta.
- если необходимо для хоста с MAC-адресом 22:22:22:33:33:33 назначить север имен 2.2.2.2, для подсети 1.2.3.0/24 назначить сервер имен 3.3.3.3, а для всех остальных - сервер имен 5.5.5.5, следует выполнить следующие команды:

```
(config—service—dhcp)# name—server 5.5.5.5
(config—service—dhcp)# host zeta
(config—service—dhcp—host—zeta)# mac 22:22:22:33:33:33
(config—service—dhcp—host—zeta)# name—server 2.2.2.2
(config—service—dhcp—host—zeta)# subnet 1.2.3.0/24
(config—service—dhcp—subnet—1.2.3.0/24)# name—server 3.3.3.3
```

31. Служба DHCP6

Dionis-NX имеет службу DHCP6, реализующую серверную часть протокола DHCPv6 (Dynamic Host Configuration Protocol v6 — протокол динамической конфигурации узла версии 6).

Для базовой информации о службе DHCP6 и протоколе см п. 30.

Помимо общей информации о DHCP, протокол DHCPv6 использует понятия префикс сети и длина префикса сети. Префикс сети - это часть IPv6-адреса, которая задает адрес сети. Дина префикса сети - обозначает, какая часть IPv6 адреса является адресом сети, причем отсчет сетевого адреса начинается слева направо. Длина префикса может быть от 0 до 128 (максимальная длина в битах IPv6 адреса).

Например FF00::1/16 - задает префикс(сеть) FF00 и 112-битное пространство адресов внутри него.

Чтобы настроить службу DHCP, следует войти в режим ее конфигурации:

service dhcp6

Команда осуществляет вход в конфигурацию DHCP-службы.

31.1 Общие настройки службы

listen <ethernet | bond> <N>

см п. (30).

max-lease-time <N>

см п. 30.

min-lease-time <N>

см п. 30.

default-lease-time <N>

см п. 30.

respond-delay <N>

см п. 30.

send-hostname <on|off>

см п. 30.

ddns-ttl <N>

см п. 30.

31.2 Настройка статического назначения

Обычно фиксированные адреса назначаются важным узлам сети,например серверам.

Рабочим станциям обычно назначают адрес динамически(см. подраздел “Настройка динамического назначения”).

host <NAME>

Вход в режим статического назначения IP-адреса для клиентов.

см п. 30.

host-id <DUID>

Задает привязку хост-декларации к идентификатору клиента.

Клиент использует уникальный идентификатор DHCP (DUID), чтобы получить IP-адрес и другие настройки от сервера DHCPv6. Фактическая длина DUID зависит от его типа. Первые 16 битов DUID содержат тип DUID, которых всего три. Значение оставшихся битов зависит от типа. Три типа, идентифицированные в RFC 3315:

- адрес уровня ссылки плюс время
- присвоенный поставщиками уникальный идентификатор
- адрес уровня ссылки

ip <IP>

Задает статический IPv6-адрес клиента.

Если клиент узел Dionis-NX, то данный адрес получит на клиенте длину префикса 64.

31.3 Настройка динамического назначения

Для настройки динамического назначения адресов необходимо описать сеть, внутри которой задает интервал адресов, подлежащих раздаче клиентам.

subnet <IP/MASK>

Вход в режим динамического назначение IP-адресов для клиентов.

range < <IP_START> <IP_END> | <SUBNET> [temporary] >

Задает диапазон адресов для назначения клиентам.

Существует два варианта задания диапазона:

- начальный и конечный адрес
- подсеть адресов; если задан параметр temporary - подсеть может использоваться для раздачи временных адресов (Temporary Addresses, опция IA_TA)

Если клиент узел Dionis-NX, то данный адрес получит на клиенте длину префикса 64.

31.4 Делегирование префиксов

В DHCPv6 существует понятие Делегирующий роутер, раздающий префиксы и Запрашивающий роутер, который их получает и затем их использует для назначения внутри них адресов для своих клиентов.

В секции host можно настроить делегирование префиксов следующей командой:

prefix <PREFIX>

В данной команде PREFIX задает префикс в формате IPV6/MASK.

В секциях subnet можно настроить делегирование префиксов следующей командой:

prefix <IP_START> <IP_END> <PLEN>

В данной команде задается начальный и конечный адрес внутри сети subnet, а также длина префикса (PLEN) для данного интервала адресов.

31.5 Сетевые DHCP-опции

Помимо назначения клиентам IP-адресов, служба может передавать им другую конфигурационную сетевую информацию.

Эта информация называется сетевые DHCP-опции. Они могут быть указаны на любом из уровней службы: как глобально, так и в хост- или сетевой декларациях.

ia-prefix <LEN>

Задает длину префикса в процессе делеигрования префиксов службой. Данная опция передается в DHCP6-подопции IAPREFIX опции IA_PD.

Обычно используется службой автоматически и не требует задания администратором.

fqdn <NAME>

Задает FQDN клиента в процессе динамического DNS-обновления зон.

Обычно используется службой автоматически и не требует задания администратором.

info-refresh-time <VAL>

Сообщает клиенту, если он этого затребует, как долго ждать следующего обновления информации через Information-request сообщения.

domain-search <NAME>

Имя домена для разрешения имен через DNS.

Можно указать несколько доменов.

[N] name-server <IP>

IP-адрес сервера имен с приоритетом N.

[N] ntp-server <IP>

IP-адрес сервера времени с приоритетом N.

unicast

Разрешает обработку Unicast-сообщений от DHCP-клиента.

Некоторые DHCP-клиенты (например, клиент Dionis-NX) шлют некоторые DHCP-сообщения (например, RELEASE) методом Unicast.

Данная опция включена по-умолчанию для службы.

Однако если включить ее для какой-то из секций службы (например, subnet), то действие опции будет ограничено этой секцией. Таким образом, можно избиратльно включать разрешение Unicast-сообщений только для конкеретных секций службы DHCP. Для других секций в этом случае будет запрещено принятие и обработка Unicast-сообщений клиентов.

31.6 Пользовательские DHCP-опции

Существует возможность создать пользовательские DHCP-опции, которые будут передаваться указанным клиентам так же, как и стандартные DHCP-опции.

Подробности см в п. 30.

31.7 Связь с DNS (динамическое обновление)

Служба может работать в связке с DNS службой по части динамического обновления DNS-записей.

Данная возможность описана в главе “Служба DNS” в пункте “Динамическое обновление зон”.

31.8 Работа со службой

Команды аналогичны службе DHCP, за исключением того, что в имени команды нужно указать имя службы dhcp6, например: show service dhcp6 log.

Подробности см п. (30).

31.9 Примеры

Рассмотрим пример настройки DHCP6:

```
listen ethernet 0
subnet fd55:1::/64
 range fd55:1::10 fd55:1::100
subnet fd77:111::/64
```

```
 1 name—server fd77:111::1
 prefix fd77:111:: fd77:111:: 64
 range fd77:111::/64
enable
```

32. Служба DCHPRELAY

Dionis-NX имеет службу DCHPRELAY, которая является ретранслятором DHCP- сообщений на указанные сервера.

Для запуска службы необходимо:

- указать хотя бы один IP-адрес сервера, на который будут перенаправляться DHCP-запросы, командой server;
- указать интерфейс для взаимодействия с DHCP-серверами командой listen-server.

Для настройки службы следует войти в ее конфигурацию:

service dhcprelay

Осуществляет вход в режим конфигурации службы DHCPRELAY.

32.1 Основные настройки

listen-server <IP>

listen <IFACE>

Задать интерфейс, на котором следует ожидать запросы/ответы DHCP и с которого будут перенаправляться DHCP-запросы на сервера DHCP.

По умолчанию: все широковещательные интерфейсы.

32.2 Дополнительные настройки

send-relay-options

Служба будет добавлять к DHCP-запросу свой идентификатор, который состоит из Circuit ID (имя аппаратного порта получения запроса) и Remote ID (MAC-адрес интерфейса получения запроса); это т.н. DHCP Опция 82 — опция протокола DHCP, использующаяся для того чтобы проинформировать DHCP-сервер о том, от какого DHCP-ретранслятора и через какой его порт был получен запрос; применяется при решении задачи привязки IP-адреса к порту коммутатора и для защиты от атак с использованием протокола DHCP.

drop-alien-replies

Отбрасывать ответы DHCP-серверов, если они содержат чужую Опцию 82 (идентификаторы в ней не соответствует идентификаторам службы)

port <PORT>

Порт, на котором слушать DHCP-запросы

По умолчанию: 67.

max-hops <NUM>

Максимальное число узлов, через которые может пройти DHCP-пакет, прежде чем будет отброшен службой.

По умолчанию: 10.

max-packet-size <SZ>

Максимальный размер DHCP-пакета (вместе с Опцией 82).

По умолчанию: 576 байт.

mode <append|replace|forward|discard>

Указывает, что делать с входящими DHCP-пакетами, которые уже имеют внутри себя Опцию 82, т.е. пришли от других релэй-агентов.

Возможные варианты:

- append : добавить свои идентификаторы к Опции 82
- replace : заменить своими идентификаторами уже имеющиеся в Опции 82 (по умолчанию)
- forward : ничего не менять
- discard : отбросить

32.3 Пример

Приведем настройку для следующей сети:

КЛИЕНТ(ethernet0)<—>DhcpRelay(ethernet1(192.168.0.1)-ethernet2(10.0.0.1))<—>DhcpServer(ethernet3(10.0.0.2))

Интерфейсы ethernet0,ethernet1 - обслуживают клиентскую сеть 192.168.0.0/24.

Интерфейсы ethernet2,ethernet3 - обслуживают сеть сервера 10.0.0.0/24.

Минимальная конфигурация Dhcp Relay:

```
(config—service—dhcprelay)# listen ethernet 1
(config—service—dhcprelay)# listen—server ethernet 2
(config—service—dhcprelay)# server 10.0.0.2
(config—service—dhcprelay)# enable
```

Минимальная конфигурация Dhcp Server:

```
(config—service—dhcp)# local—address 10.0.0.2
(config—service—dhcp—subnet—192.168.0.0/24)# range 192.168.0.10 192.168.0.100
(config—service—dhcp)# enable
```

33. Служба DCHPRELAY6

Dionis-NX имеет службу dhcprelay6, которая является ретранслятором DHCPv6- сообщений на указанные сервера.

Для запуска службы необходимо:

- указать хотя бы один интерфейс командой listen-from;
- указать хотя бы один интерфейс командой listen-to;

Для настройки службы следует войти в ее конфигурацию:

service dhcprelay6

Осуществляет вход в режим конфигурации службы dhcprelay6.

33.1 Основные настройки

listen-from <IFACE> [IP]

Задать интерфейс и его адрес, на котором служба будет ожидать запросы клиентов или других ретрансляторов DHCPv6. Если IP не задан, будет использоваться первый найденный не Link-local адрес интерфейса.

listen-to <IFACE> [IP]

Задать интерфейс и его адрес, на который служба будет перенаправлять запросы клиентов или других ретрансляторов DHCPv6. Если IP не задан, будет использоваться адрес FF02::1:2 (All_DHCP_Relay_Agents_and_Servers).

33.2 Дополнительные настройки

send-iface-id

Слать Interface-ID опцию, даже если указан только один интерфейс командой listen-from. В случае, если таких интерфейсов несколько, данная опция шлется в любом случае.

max-hops <N>

Максимальное число узлов, через которые может пройти DHCPv6-пакет, прежде чем будет отброшен службой.

Умолчание: 10.

port <PORT>

Порт, на котором слушать DHCPv6-запросы

По умолчанию: 547.

33.3 Пример

Приведем настройку для следующей сети:

КЛИЕНТ(ethernet0)<—>DhcpRelay(ethernet1-ethernet2)<—>DhcpServer(ethernet3)

Интерфейсы ethernet0,ethernet1 - обслуживают клиентскую сеть.

Интерфейсы ethernet2,ethernet3 - обслуживают сеть сервера.

Минимальная конфигурация Dhcp Relay:

```
(config—service—dhcprelay)# listen—from ethernet 1
(config—service—dhcprelay)# listen—to ethernet 2
(config—service—dhcprelay)# enable
```

34. Служба PROXY

Dionis-NX имеет службу PROXY, которая является прокси-сервером для протоколов FTP и HTTP.

Прокси-сервер - это служба, позволяющая клиентам выполнять косвенные запросы к другим сетевым службам.

Далее под службой будем понимать описываемую прокси-службу Dionis-NX, под УС будем понимать удаленный сервер, ресурс которого желает получить клиент через прокси-службу.

Рассмотрим алгоритм обработки HTTP-запроса клиента через PROXY-службу при задействовании всех ее возможностей:

1. клиент пытается получить ресурс, расположенный, например, на удалённом HTTP-сервере (web-страница, картинки, аудио-, видео-файлы и др.);
2. клиент подключается к службе или запрос клиента перенаправляется маршрутизатором на службу;
3. служба аутентифицирует пользователя, анализирует запрос клиента и проверяет,разрешен ли данный запрос для данного клиента;
4. если запрос разрешён, могут быть сделаны модификации заголовков HTTP-запроса клиента;
5. далее служба ищет запрашиваемый ресурс в своем кэше;
6. если ресурс найден в кэше, делается проверка на свежесть ресурса;
7. если ресурс свеж, он возвращается клиенту;
8. если ресурс стар или не найден в кэше, то служба пытается получить его с удаленного HTTP-сервера, возможно на ограниченной скорости;
9. если ресурс получен с удаленного сервера, то служба проверяет, разрешен ли данный ресурс для возврата клиенту;
10. если ресурс разрешен, то служба проверяет, нужно ли его кэшировать;
11. перед возвратом ресурса его HTTP-заголовок также может быть изменен;
12. наконец ресурс возвращается клиенту.

В дальнейшем в разделах данной главы будет указано, к какому пункту данного алгоритма относится раздел.

Таким образом, основные цели службы:

- кэширование данных: может держать копию часто запрашиваемых Интернет-ресурсов в своем кэше и выдавать по запросу, снижая тем самым нагрузку на Интернет-канал и ускоряя получение клиентом запрошенной информации;
- защита локальной сети от внешнего доступа: локальные узлы взаимодействуют с внешними ресурсами только через службу, а внешние узлы не могут обращаться к локальным;
- ограничение доступа из локальной сети к внешней:
  - на удаленные ресурсы определённого типа;
  - для определённых клиентов;
  - на объем трафика извне в сторону определённых клиентов и/или сетей.

34.1 Общие понятия

34.1.1 Режимы работы службы

Существуют два режима работы службы:

- обычный режим: необходимы настройки прокси-сервера в браузере клиента; поддерживает проксирование/кэширование HTTP,FTP протоколов
- прозрачный режим: настраивать браузеры клиентов не нужно; необходима настройка NAT-правил; поддерживает проксирование/кэширование только HTTP протокола; успешность выполнения запроса полностью зависит от наличия заголовка Host в HTTP-запросе клиента.

Более подробно настройка службы в обоих режимах описана далее в подразделе "Общие настройки службы".

34.1.2 Правила и списки доступа

Основные понятия службы - списки контроля доступа(ACL) и правила доступа. ACL - это именованный набор элементов определённого типа.

Формат ACL следующий: **acl <NAME> <TYPE> <PARAMS>**

Рассмотрим поля команды:

- NAME - это имя ACL;
- TYPE - это тип ACL; определяет тип элементов данного списка ACL; может принимать множество значений, рассмотренных ниже;
- PARAMS - это элементы, входящие в данный список.

Рассмотрм возможные типы ACL:

- src - исходный IP или сеть;
- dst - IP или сеть назначения;
- localip - локальный IP-адрес, к которому подключился клиент;
- srcdomain - доменное имя клиента;
- dstdomain - доменное имя сервера назначения;
- krb-ldap-group - группа Kerberos LDAP;
- srcdom-regex - регулярное выражение в домене источника;
- dstdom-regex - регулярное выражение в домене назначения;
- time - спецификация времени;
- port - порт назначения;
- localport - локальный сервисный порт, к которому подключился клиент;

- proto - протокол;
- method - метод HTTP;
- user-agent - User-Agent запроса или ответа;
- proxy-auth - указанное аутентифицированное имя пользователя;
- proxy-auth-all - все аутентифицированные имена пользователей;
- proxy-auth-regex - проверка имени пользователя по шаблону для определения доступности запроса;
- maxconn - максимальное число прямых TCP-соединений (X-Forwarded-For не учитывается);
- max-user-ip - максимальное число попыток входа с разных IP для одного пользователя;
- req-mime - MIME-тип запроса, сгенерированного клиентом;
- rep-mime - MIME-тип ответа;
- mac - Ethernet (MAC) адрес для клиентов одной подсети;
- uri - регулярное выражение в URI;
- urn - регулярное выражение в URN;
- req-header - регулярное выражение в значении заголовка HTTP запроса;
- rep-header - регулярное выражение в значении заголовка HTTP ответа;
- ssl-error - ошибка проверки SSL сертификата;
- ssl-fingerprint - идентификатор сертификата SSL сервера;
- ssl-srv - имя сервера, полученное из процесса установления соединения TLS/SSL;
- ssl-srv-regex - регулярное выражение имени сервера, полученного из процесса установления соединения TLS/SSL;
- ssl-bump - текущий шаг при оценке ssl_bump.

Под источником и назначением обычно в службе прокси понимается клиент и сервер.

На примерах рассмотрим, как создавать ACL:

```
(config—service—proxy)# acl a1 src 10.0.0.0/24 10.0.1.0/24
(config—service—proxy)# acl a1 src 10.0.2.0/24
(config—service—proxy)# acl a2 src 10.0.3.0/24
(config—service—proxy)# acl u1 uri xxx
```

В результате: ACL a1 будет состоять из 3х сетей 10.0.0.0/24 10.0.1.0/24 10.0.2.0/24; ACL a2 будет состоять из сети 10.0.3.0/24; ACL u1 будет состоять из всех запросов, в URI которых есть слово xxx.

При поиске службой того ACL, в пределы которого попадает запрос клиента или ответ сервера, используется OR-логика: если запросу соответствует хотя бы один из элементов, перечисленных в данном ACL, то считается, что запрос попал в данный ACL, и поиск прекращается. Поэтому, для оптимизации работы службы, при определении параметров ACL лучше задавать первым тот параметр ACL, вероятность попадания в который **максимальная**.

Рассмотрим пример создания правил доступа на основе ранее созданных ACL:

```
(config—service—proxy)# http—access deny a1 u1
(config—service—proxy)# http—access permit a2
(config—service—proxy)# http—access deny all
```

При поиске службой правила, в который попадает запрос клиента или ответ сервера, используется AND-логика: запрос/ответ должен удовлетворять всем ACL, перечисленным в правиле. Поэтому,

для оптимизации работы службы, при определении правил на основе ACL лучше задавать первым тот параметр правила, вероятность попадания в который **минимальная**.

В данном примере доступ запрещается для запросов, приходящих из сетей a1, только если они пытаются получить ресурс, в URI которого есть слово xxx. Для сетей из a2 разрешается свободный доступ. Для всех остальных сетей доступ запрещён (используется встроенный ACL с именем all).

Помимо правил доступа, существуют и другие команды, использующие ACL. Например: Рассмотрим пример создания правил доступа на основе ACL:

```
(config—service—proxy)# max—reply—size a1
```

Наконец, добавим еще одно правило по оптимизации работы службы: правила на основе ACL типа srcdomain, dst, proxy-auth лучше определять как можно раньше в списке.

Важное замечание: для правила доступа любого типа рекомендуется всегда добавлять в конец списка правил данного типа запрещающее или разрешающее (зависит от контекста) правило,направленное на всех (например, http-access deny all).

Существуют ACL особого типа - комплексные ACL:

```
(config—service—proxy)# acl—any any1 a1 a2
```

Данный ACL any1 выполнится, если выполнится хотя бы один из ACL a1 или a2.

```
(config—service—proxy)# acl—all all1 a1 a2
```

Данный ACL all1 выполнится, если выполнятся все из ACL a1 или a2.

34.1.3 Регулярные выражения

В некоторых командах службы параметры задаются упрощёнными регулярными выражениями (РВ). Тип параметра в этом случае называется REGEX.

РВ - это формальный язык поиска подстрок в тексте, основанный на использовании метасимволов.

По сути РВ - это строка-образец, состоящая из символов (С) и метасимволов (МС) и задающая правило поиска.

МС - это С, который используется для замены других С или их последовательностей, приводя таким образом к символьным шаблонам.

В дальнейшем в этом разделе все символы,указанные в кавычках, должны вводиться в системе без кавычек.

Кратко опишем используемые МС:

- «?» - С перед данным МС может быть, а может и отсутствовать; например: «ax?» - это «ax» или «a»;
- «*» - количество С перед данным МС больше либо равно 0; например: «ax*» - это «a», «ax», «axx», «axxx» и т.д.;
- «+» - количество С перед данным МС больше либо равно 1; например: «ax+» - это «ax», «axx», «axxx» и т.д.;

- «[]» - задает наборы символов; например: [a-z123=] - это все строчные англ.буквы, цифры 1,2,3 и знак равно;
- «.» - любой символ; например: .* - это любое множество,в том числе пустое, любых символов;
- «{}» - задает число повторений предыдущего символа; например: [a-z]{1,3} - любые одно-, двух- и трехбуквенные слова;
- «^» - задает начало строки; например: ^123[ab]? - это все строки, начинающиеся на 123, 123a или 123b;
- «$» - задает конец строки; например: [0-9]{2}$ - это все строки, оканчивающиеся на двухзначные числа.
- «()» - задает группу символов; например: a([0-9]b){2} - это a1b4b, a5b2b и т.д., т.е. МС {2} применяется к группе ([0-9]b).

Существует возможность использовать МС как С, т.е. МС будет иметь только функцию символа, т.е. отображать сам себя. Это называется экранирование МС.

Символ экранирования (СЭ) - «\\».

При установке параметра типа REGEX,если в значении параметра используется СЭ, необходимо взять все значение параметра в двойные кавычки,например:

```
(config—service—proxy)# acl a1 urn ”\\.cgi$”
```

При удалении параметра типа REGEX,если в значении параметра используется СЭ, необходимо взять все значение параметра в двойные кавычки и продублировать СЭ,например:

```
(config—service—proxy)# no acl a1 urn ”\\\\.cgi$”
```

34.2 Общая настройка службы

Часть информации о настройке в данном разделе,а именно настройка режимов работы службы, относится п.2 Алгоритма.

Чтобы войти в режим настройки службы, следует выполнить команду:

```
(config)# service proxy
```

Необходимо также задать почтовый адрес администратора службы, которая будет указана в веб-страницах, описывающих проблему с выполнением запроса клиента, чтобы он мог ,в случае необходимости, узнать причину данной проблемы:

```
(config—service—proxy)# admin—email ivanov@company.ru
```

По умолчанию используется адрес admin@company.

Важно: система должна быть способна разрешать имена узлов через DNS. Для этого либо настройте и включите службу DNS и/или укажите в ip resolver адрес(а) серверов имен.

34.2.1 Настройка службы в обычном режиме для HTTP/FTP.

Следует настроить порт и адрес(необязательно), на которых служба будет ожидать входящие запросы:

```
(config—service—proxy)# listen 192.168.0.1 3128
```

Данная команда настраивает службу в обычном режиме, т.е. клиентам необходимо будет указать в настройках своих Интернет-браузеров адрес и порт службы.

Для проксирования FTP в обычном режиме используйте команду listen-ftp.

34.2.2 Настройка службы в прозрачном режиме для HTTP/FTP.

Обычный режим может быть не очень удобным, т.к. необходимо настраивать браузеры клиентов.

Поэтому можно настроить службу в режиме intercept (режим перехвата или прозрачный режим).

Для проксирования FTP в прозрачном режиме используйте команду listen-ftp с опцией intercept.

Рассмотрим пример настройки службы в прозрачном режиме, обслуживающую запросы, приходящие из сети 192.168.1.0/24 на 80-й порт(HTTP) на интерфейс ethernet2 с адресом 192.168.1.254:

```
(config)# ip nat—list proxy
(config—nat—proxy)# exclude in tcp dport 80 dst 192.168.1.254
(config—nat—proxy)# nat tcp dport 80 src 192.168.1.0/24 redirect port 3127
(config)#
(config)# ip access—list dropmyself
(config—acl—dropmyself)# deny tcp dport 3127 dst 192.168.1.254
(config)#
(config)# interface ethernet 2
(config—if—ethernet2)# ip nat—group proxy
(config—if—ethernet2)# ip access—group dropmyself in
(config)#
(config—service—proxy)# listen 192.168.1.254 3127 intercept
```

В данном режиме службы клиентам не нужно ничего настраивать, они автоматически, после запуска службы, будут работать через нее. В данном примере предполагалось, что клиенты обслуживаются маршрутизатором Dionis-NX, на котором и настраивается прокси-служба.

Рассмотрим вышеприведенные команды:

- правило proxy перенаправляет трафик на 80-й порт с адресов сети 192.168.1.0/24 на порт 3127, исключая при этом запросы на 80й порт локальной WEB-службы;
- правило dropmyself отбрасывает весь трафик,приходящий на адрес и порт прокси сервера, объявленные ниже как intercept: это нужно для предотвращения обработки запросов непосредственно на прерывающий сокет службы, т.е. на 192.168.1.254:3127, т.к. это может вызвать петлю перенаправления запросов, когда служба будет бесконечно слать запрос самой себе, думая что она и есть удалённый сервер;

- ip nat-group proxy и ip access-group dropmyself in применяют вышеописанные правила на интерфейсе; вместо ip nat-group нужно использовать ip nat-group-xfrm, если трафик через интерфейс предполагается заворачивать в какой-либо туннель,например DISEC: в этом случае NAT выполняется до «заворачивания» в туннель для отсылаемого через интерфейс трафика и после «разворачивания» из туннеля для принимаемого интерфейсом трафика;
- команда listen непосредственно определяет сокет (адрес и порт) для принятия HTTP-запросов и объявляет его прерывающим сокетом; пары 0.0.0.0:3128 и 127.0.0.1:3128 запрещены.

Важно: в команде listen для режима intercept можно указать опцию no-pmtu-discovery, если у некоторых клиентов иногда возникают проблемы с долгим ожиданием обработки HTTP-запроса. Это может быть связано проблемой доставки ICMP Must fragment сообщения:

```
(config—service—proxy)# listen 192.168.1.254 3127 intercept no—pmtu—discovery
```

Вы можете указать неограниченное количество сокетов обоих типов, на которых будут приниматься запросы. Единственное ограничение: один и тот же сокет может быть только одного типа.

34.3 Настройка работы с HTTPS-трафиком

34.3.1 Базовые понятия проксирования HTTPS-трафика.

Для перехвата HTTPS-трафика необходим УЦ-сертификат (CertificateAthority). По умолчанию служба использует самоподписанный сертификат, который создается автоматически при создании службы. Также можно использовать собственный УЦ-сертификат, который может быть импортирован командой **service proxy import**. Данный сертификат будет использоваться для генерации сертификатов проксируемых HTTPS-серверов, к которым подключаются клиенты. Таким образом данный сертификат становится корневым.

Проверить, является ли данный сертификат УЦ-сертификатом можно следующей командой:

```
openssl x509 —in cert.pem —noout —text | grep —A1 'Basic Constraints'
```

Значение поля CA должно быть TRUE: "CA:TRUE".

Кроме того, клиентское ПО необходимо настроить таким образом, чтобы оно доверяло данному самоподписанному сертификату. Для этого есть команда **service proxy export**, которая позволяет экспортировать сертификат службы в файл в формате PEM, а также в формате DER, который может использоваться для импорта в Интернет-браузеры клиентов.

Кроме того, могут понадобиться УЦ-сертификаты (CA) и списки отзыва сертификатов (CRL) для проверки сертификатов клиентов. Импортировать архив с указанными файлами сертификатов и CRL (опционально) можно командой **service proxy importd FILE** , где FILE - путь к архиву в tar.gz или tar.bz2 форматах. При импорте осуществляются следующие преобразования для файлов архива:

- для всех der-сертификатов создаются pem-сертификаты и наоборот.

- pem-сертификаты переименовываются в файлы с форматом HASH.0, а соответствующий ему der-сертификат в файл с форматом HASH_NAME.DER, где NAME - исходное имя файла, HASH - специальная HASH-сумма, которая может быть получена для сертификата командой **openssl x509 -hash -noout -in cert.pem**

При перехвате HTTPS трафика в режиме listen ssl-bump служба проходит через несколько этапов(шагов) процесса установки соединения TCP/TLS.

Схема создания SSL/TLS соединения:

- Client: TCP CONNECT–> Server
- Client <–TCL Accept : Server
- Client: TLS ClientHello (cipher,SNI,…) –> Server
- Client <– TLS ServerHello (cipher,…): Server
- Client <– TLS key exchange –> Server
- Client <– HTTP data exchange –> Server

Действия:

- **peek** (шаг 1, шаг 2) - на шаге 1 осуществляется переход на шаг 2, на котором из TLS ClientHello извлекается значение для SNI (имя сервера); на шаге 2 осуществляется переход на шаг 3, на котором из TLS Server Hello извлекается сертификат для дальнейшего возможного действия splice.
- **splice** (шаг 1, шаг 2 и иногда шаг 3) - создать TCP-туннель без расшифровывания трафика; клиент и сервер обмениваются данными как будто нет прокси-сервера между ними.
- **stare** (шаг 1, шаг 2) - на шаге 1 осуществляется переход на шаг 2, на котором из TLS ClientHello извлекается значение для SNI (имя сервера); на шаге 2 осуществляется переход на шаг 3, на котором из TLS Server Hello извлекается сертификат для дальнейшего возможного действия bump.
- **bump** (шаг 1, шаг 2 и иногда шаг 3) - создает TLS соединение с сервером (используя SNI, если он есть) и с клиентом (используя извлеченный сертификат сервера)
- **terminate** (шаг 1, шаг 2, шаг 3) - закрыть соединение с клиентом и сервером.

Шаг 1.

1. Выполняется всегда.

2. Получаем информацию TCP-уровня от клиента (IP-адреса и порты):
   а. В случае команды listen ssl-bump (обычный режим проксирования), происходит парсинг CONNECT-запроса б. В случае команды listen-https (прозрачный режим проксирования), происходит создания псевдо-запроса типа CONNECT на основе TCP-информации клиентского запроса
3. Выполняем указанный выше процесс создания соединения на основе CONNECT.
4. Выполняем первое подходящее ssl-bump правило, имеющее одно из следующих действий: splice,bump,peek,stare или terminate.
5. Запись в access лог CONNECT-запроса - для listen-https (прозрачный режим) записываются только IP-адреса.

Шаг 2.

1. Выполняется только если peek или stare выполнились на шаге 1.
2. Получаем SNI (если он есть) из TLS Client Hello и корректируем CONNECT-запрос из шага 1.
3. Выполняем указанный выше процесс создания соединения на основе CONNECT.
4. Выполняем первое подходящее ssl-bump правило, имеющее одно из следующих действий: splice,bump,peek,stare или terminate.
5. Если этот шаг финальный (нет шага 3), то происходит запись в access лог CONNECT-запроса.

Шаг 3.

1. Выполняется только если peek или stare выполнились на шаге 2.
2. Получаем сертификат сервера и другую информацию из TLS Server Hello.
3. Проверяем сертификат сервера.
4. Выполняем первое подходящее ssl-bump правило, имеющее одно из следующих действий: splice,bump или terminate.
5. Происходит запись в access лог CONNECT-запроса.

34.3.2 Команды для работы с HTTPS трафиком

ssl-bump <splice|peek|bump|stare|terminate> {<ACL>,5}

Команда задает правило перехвата HTTPS-трафика, в котором указано действие, выполняемое для указанных ACL. Действия описаны выше в п.“Базовые понятия проксирования HTTPS-трафика”.

ssl-cert-adapt <ALG> {<ACL>,5}

Команда задает правило генерации сертификата на основе заданного алгоритма изменения для указанных ACL. Алгоритм изменения (ALG) может быть следующим:

- valid-after - установить значение параметра NotAfter в значение параметра NotAfter из сертификата Центра сертификации, используемого для подписи сгенерированных сертификатов
- valid-before - установить значение параметра NotBefore в значение параметра NotBefore из сертификата Центра сертификации, используемого для подписи сгенерированных сертификатов

- common-name <DOMAIN|from-connect> - установить значение параметра Subject.CN в значение,извлеченное из CONNECT-запроса (для обычного режима проксирования) или в указанное значение DOMAIN (для прозрачного режима проксирования)

ssl-cert-sign <ALG> {<ACL>,5}

Команда задает правило генерации сертификата на основе указанного алгоритма подписи для указанных ACL. Алгоритм (ALG) может быть следующим:

- trusted - подписывать сертификат сертификатом Центра сертификации
- untrusted - подписывать для генерации ошибки X509_V_ERR_CERT_UNTRUSTED
- self - подписывать самоподписанным сертификатом с верным CN для генерации ошибки X509_V_ERR_DEPTH_ZERO_SELF_SIGNED_CERT

ssl-cert-error <permit|deny> {<ACL>,5}

Команда задает правило реакции на ошибки проверки сертификата - игнорировать ошибки или нет.

ssl <OPTS>

Команда настраивает различные опции SSL:

- unclean-shutdown - разрешить некорректное завершение SSL-сессии
- not-verify-peer - принимать сертификаты не прошедшие проверку
- no-default-ca - не использовать встроенный в системную SSL-библиотеку список Центров сертификации (CA)
- helpers-num - задать число процессов TLS/SSL-перехвата (по умолчанию 5)
- storage-sz - задать максимальный размер всех сохраняемых в системе сертификатов
- cert-sign-hash - задать алгоритм хеширования при подписи сгенерированных сертификатов
- session-cache-size - размер кэша для сессии SSL (в МБ)
- session-ttl - задать таймаут SSL-сессии
- version - задать версию SSL/TLS используемую при проксировании HTTPS
- opts - расширенные опции SSL:
  - nossl2/nossl3/notls10/notls11/notls12 - запретить использование SSL версий 2,3 и TLS версий 1.0,1.1 и 1.2
  - singledh - всегда создавать новый эфемерный ключ при генерации общего секрета
  - noticket - не использовать RFC5077 для сессионных ключей
  - all - обход известных ошибок в системной библиотеке SSL (может повысить уязвимость SSL/TLS к некоторым атакам)

34.3.3 Настройка службы в прозрачном режиме для HTTPS.

Данный режим аналогичен прозрачному режиму для HTTP (порт 80), за исключением того, что происходит проксирование HTTPS трафика (порт 443).

Рассмотрим основные команды для настройки перехвата HTTPS-трафика.

```
(config—service—proxy)# listen—https 10.0.0.1 3128
```

Данная команда осуществляет принятие HTTPS запросов на сокете 10.0.0.1:3128.

У этой команды существует ряд необязательных параметров:

- no-pmtu-discovery - см. описание в предыдущем пункте.
- gencert - создавать динамически сертификаты для узлов назначения; эти сертификаты подписываются сертификатом, который создается автоматически при создании службы (и может быть пересоздан командой clear proxy cert)
- cachesize N - задает объем памяти для кэширование сгенерированных сертификатов; по умолчанию - 4МБ, если задан 0 - кэширование отключено
- no-default-ca - не использовать встроенный в системную библиотеку SSL список УЦ
- no-session-reuse - не использовать сессию повторно для нового соединения, т.е. каждое соединение - это новая SSL-сессия
- verify-crl - использовать CRL, импортированные командой service proxy importd, при проверке клиентских сертификатов
- verify-crl-all - использовать CRL, импортированные командой service proxy importd, при проверке всех сертификатов в цепочке сертификатов клиента

Для осуществления проксирования необходимо настроить nat- и access-list, как это сделано в предыдущем пункте,за исключением того, что перенаправлять необходимо трафик не 80-го порта, а на 443-го. Рассмотрим скорректированные команды:

```
(config)# ip nat—list proxy
(config—nat—proxy)# exclude in tcp dport 443 dst 192.168.1.254
(config—nat—proxy)# nat tcp dport 443 src 192.168.1.0/24 redirect port 3128
(config)#
(config)# ip access—list dropmyself
(config—acl—dropmyself)# deny tcp dport 3128 dst 192.168.1.254
(config)#
(config)# interface ethernet 2
(config—if—ethernet2)# ip nat—group proxy
(config—if—ethernet2)# ip access—group dropmyself in
(config)#
(config—service—proxy)# listen—https 192.168.1.254 3128
```

34.3.4 Настройка службы в обычном режиме для HTTPS.

Данный режим аналогичен обычному режиму для HTTP (порт 80), за исключением того, что происходит проксирование HTTPS трафика (порт 443). В этом режиме служба перехватывает каждый CONNECT-запрос, совпадающий с ACL ssl-bump, устанавливает защищенное соединение между клиентом и сервером, расшифровывает HTTPS-сообщения во время их прохождения через службу, и считает их расшифрованными HTTP-сообщениями, становясь “man-in-the-middle” (“человек-посередине”).

```
(config—service—proxy)# listen 10.0.0.1 3128 ssl—bump
```

У этой команды существует ряд необязательных параметров, которые описаны в предыдущем пункте (за исключением параметра no-pmtu-discovery).

34.4 Настройка параметров кэширования.

Далее настроим параметры службы, описывающие кэш.

Настройка кэша:

```
(config—service—proxy)# cache type aufs 32 256
```

Данная команда задает тип кэша на диске aufs, размер кэша в памяти 256 Мб и на диске 32 Мб.

Другой тип кэша, который может быть использован в данной команде, это ufs. Отличие ufs от aufs в том, что первый обеспечивает синхронные файловые операции над кэшем, а второй - асинхронные, что более эффективно. Рекомендуется всегда использовать aufs.

Настройка механизма замены объектов в кэше на диске и в памяти:

```
(config—service—proxy)# cache replacement—policy disk hlru
(config—service—proxy)# cache replacement—policy memory hlru
```

Рекомендуется использовать один и тот же механизм для кэша в памяти и кэша на диске.

Опишем возможные механизмы замены объектов:

- lru : Least Recently Used (сначала заменяются самые старые объекты, к которым давно не было доступа);
- hlru : Heap-based Last Recently Used (аналогично предыдущему,но в новой, более эффективной реализации);
- gdsf : Greedly Dual Size Frequency (в кэше остаются прежде всего популярные маленькие объекты);
- lfuda : Least Frequently Used with Dinamic Aging (в кэше остаются прежде всего популярные объекты,вне зависимости от размера).

Настройка размеров кэшируемых объектов на диске(первые две команды) и в памяти(третья команда):

```
(config—service—proxy)# cache limit disk max 500
(config—service—proxy)# cache limit disk min 5
(config—service—proxy)# cache limit memory max 8
```

Размер задается в Кб. Объекты,размер которых выходит за указанные рамки, не будут кэшироваться. Минимальное ограничение для объектов в памяти равно 0 и не может быть изменено.

34.5 Настройка доступа к службе

Данный раздел относится к пп.3 и 9 Алгоритма.

С помощью этих настроек осуществляется контроль над тем, какие узлы могут иметь доступ к сети Интернет и какие данные они могут запрашивать через прокси-службу.

Для начала напомним, как формируется URI (Унифицированный идентификатор ресурса): URI=URL|URN, где | - конкатенация.

Напрмер, URI = ftp://ftp.dlink.ru/pub/ADSL/ , где: URL = ftp://ftp.dlink.ru , URN = /pub/ADSL/

Далее рассмотрим примеры.

Запрет доступа к нежелательным сайтам и фильтрация ответов сервра:

```
(config—service—proxy)# acl workers src 10.0.0.0/24
(config—service—proxy)# acl bosses src 10.0.1.0/24
(config—service—proxy)# acl mime1 rep—mime ^audio
(config—service—proxy)# acl blacklist uri ^http://jihad
(config—service—proxy)# acl blacklist uri kommunist
(config—service—proxy)# acl blacklist urn ^/cgi—bin
(config—service—proxy)# acl blacklist uri ^http://bad\\.site\\.number[0—9]+\\.sym—.\\.tail—*\\.ru$
(config—service—proxy)# acl docsite  uri ^http://documentation
(config—service—proxy)# http—access deny blacklist workers
(config—service—proxy)# http—access deny docsite bosses
(config—service—proxy)# http—access permit all
(config—service—proxy)# http—reply—access deny mime1
```

Данные правила запрещают работникам (сеть workers) посещать следующие сайты:

- сайты, начинающиеся на http://jihad;
- сайты, в URI которых есть слово kommunist;
- сайты, в URN которых есть /cgi-bin, т.е. запрет доступа к CGI-приложениям серверов;
- сайты, имеющие вид http://bad.site.number$A.sym-$B.tail-$C.ru, где:
  - $A -это набор цифр от 0 до 9;
  - $B - любой символ;
  - $C - любой набор символов, том числе и пустой.

Кроме того, данные правила запрещают директорам (сеть bosses) посещать сайты, начинающийся на http://documentation.

Другим лицам (которые не входят в категории bosses и workers) доступ на любые сайты разрешен.

Если доступ разрешен, то используется последнее правило (http-reply-access): после получения ответа от сервера анализируется тип содержимого ответа (поле Content-Type HTTP-заголовка ответа) и если типа содержимого начинается с audio, то содержимое клиенту не возвращается и не кэшируется.

34.6 Настройка фильтрации HTTP-заголовков

Данный раздел относится к п.4 и 11 Алгоритма.

Служба имеет возможность удалять HTTP-загловки и изменять их значения в запросах клиентов и/или ответах сервера.

По умолчанию никакие заголовки не удаляются.

Для исключения заголовка из HTTP-пакета запроса и ответа используются следующие команды: request-header-access deny и reply-header-access deny соответственно.

Для включения заголовка в HTTP-пакета запроса и ответа используются следующие команды: request-header-access permit и reply-header-access permit соответственно.

Для замены содержимого в ранее исключенных HTTP-заголовках запроса и ответа используются следующие команды: request-header-replace и reply-header-replace соответственно.

Следует обратить внимание, что замена содержимого работает только для заголовков, которые попадают в request/reply-header-access deny правила.

Рассмотрим пример. Скрытие от удаленных серверов названия приложения, с помощью которого клиенты локальной сети работают в сети Интернет:

```
(config—service—proxy)# acl lan src 10.0.0.0/24
(config—service—proxy)# request—header—access deny User—Agent lan
(config—service—proxy)# request—header—access permit All lan
(config—service—proxy)# request—header—replace User—Agent ”Fake Browser 1.0”
```

Данные команды заменяют информацию в заголовке User-Agent во всех исходящих из сети 10.0.0.0/24 HTTP-пакетах на Fake Browser 1.0.

Аналогично можно менять заголовки в ответах HTTP-серверов командами reply-header-access/reply-header-replace.

34.7 Настройка выборочного проксирования

Служба имеет возможность предписывать, какие запросы и для каких пользователей следует всегда направлять только через службу proxy, а какие следует передавать напрямую на сервер назначения, минуя службу proxy.

Однако ответы от сервера назначения будут также и далее кэшироваться службой, только если это не запрещано специально (см. раздел “Настройка выборочного кэширования”).

Рассмотрим примеры. Запретим проксирование всех запросов на utro.ru и его поддомены, кроме поддомена mail.utro.ru:

```
(config—service—proxy)# acl direct1 dstdomain mail.utro.ru
(config—service—proxy)# acl direct2 dstdomain .utro.ru
(config—service—proxy)# always—direct deny direct1
(config—service—proxy)# always—direct permit direct2
```

Как видно из примера, для указания запросов, перенаправляемых напрямую, используется команда always-direct.

Существует противоположный ей вариант - команда never-direct - предписывает, какие запросы не следует направлять напрямую на сверер назначения.

Рассмотрим пример ее использования:

```
(config—service—proxy)# acl local—servers dstdomain .foo.net
(config—service—proxy)# never—direct deny local—servers
(config—service—proxy)# never—direct allow all
```

Данный пример предписывает все запросы, кроме запросов на поддомены домена foo.net , выполнять через службу.

34.8 Настройка выборочного кэширования

Данный раздел относится к п.10 Алгоритма.

Служба имеет возможность предписывать, какие объекты и для каких пользователей следует кэшировать, а какие нет.

Рассмотрим примеры. Запретим кэширование объектов, расположенных в локальной сети:

```
(config—service—proxy)# acl lan dst 10.0.0.0—10.0.3.0/24
(config—service—proxy)# caching deny lan
(config—service—proxy)# caching permit all
```

Обычнно не следует кэшировать объекты, находящиеся на HTTP-серверах локальной сети, т.к. доступ к ним и так достаточно быстрый. Это сохранит место в кэше для других объектов. В данном примере все HTTP-запросы на адреса из четырех сетей (10.0.0.0/24 - 10.0.3.0/24) не будут кэшироваться.

Другой пример. Запретим в LAN кэширование всего, кроме аудиоинформации:

```
(config—service—proxy)# acl lan dst 10.0.0.0—10.0.3.0/24
(config—service—proxy)# acl aud req—mime ^audio
(config—service—proxy)# caching deny lan !aud
(config—service—proxy)# caching permit all
```

34.9 Настройка аутентификации и авторизации

Данный раздел относится к п.3 Алгоритма.

Служба имеет возможность проводить аутентификацию пользователей, желающих работать через нее. Это возможно только в обычном режиме работы службы.

В этом случае клиенты, настроив свой браузер на использование прокси, при попытке выхода в Интернет получат приглашение (с названием DionisNX-PROXY) ввести имя пользователя и пароль. Служба,

одобрив аутентификацию, может с помощью правил доступа задать разное обслуживание для разных пользователей, прошедших аутентификацию.

В службе существует 3 вида аутентифицации:

- локальная (digest)
- удаленная по LDAP протоколу (basic)
- удаленная по RADIUS протоколу (basic)

Они могу быть в системе одновременно, в этом случае необходимо учитывать следующие особенности:

- при добавлении аутентификации нового типа при включенной службе, необходимо ее перезапустить

- схемы аутентификации будет предлагаться клиенту в порядке их нахождения в конфигурации (зависит от приоритета команды auth)
- при наличии двух видов аутентификации в конфигурации для разных Интернет-браузеров (клиентов службы), может использоваться свой тип аутентификации: выбор зависит от браузера и удаленной системы; например, может выбраться первая предложенная службой аутентификация, а может и более криптографически сильная из двух.

34.9.1 Локальная аутентификация.

Для аутентификации данного типа необходимо добавить в конфигурацию системы имена пользователей и их пароли. Рассмотрим пример.

Сначала включим локальную аутентификацию:

```
(config—service—proxy)# auth local 3
```

Указываем число параллельных процессов аутентификации.

Далее добавим пользователей:

```
(config—service—proxy)# user ivan pas1
(config—service—proxy)# user liza pas2
(config—service—proxy)# user oleg pas3
(config—service—proxy)# user mary pas4
```

Например, у пользователя ivan пароль pas1.

Далее создадим список доступа для пользователей:

```
(config—service—proxy)# acl allusers proxy—auth—all
(config—service—proxy)# acl vipusers proxy—auth ”oleg mary”
(config—service—proxy)# acl url1 url—regex ^http://vk
(config—service—proxy)# http—access permit url1 vipusers
(config—service—proxy)# http—access deny url1 allusers
(config—service—proxy)# http—access permit allusers
(config—service—proxy)# http—access deny all
```

Данные правила разрешают доступ в Интернет только тем пользователям, которые прошли аутентификацию. Среди аутентифицированных пользователей запрещен доступ к сайтам, начинающимся на http://vk, всем, кроме пользователей oleg и mary (vipusers).

Также vipusers можно задать в следующем виде:

```
(config—service—proxy)# acl vipusers2 proxy—auth oleg
(config—service—proxy)# acl vipusers2 proxy—auth mary
```

В данном случае ACL vipusers и vipusers2 полностью идентичны.

34.9.2 Удаленная LDAP-аутентификация

В данном случае аутентификация проходит на удаленном сервере аутентификации (далее CA), поддерживающем базу данных пользователей. Этот сервер должен поддерживать LDAP протокол. Например, это может быть Windows Server с поддержкой Active Directory (данная подсистема имеет LDAP-компонент).

Формат команды для включения такой аутентификации:

auth ldap <IP> <USERS_DN> [user <BIND_DN> <PAS>] [filter FILTER [lookup]] [NUM] [credttl SEC] [ssltls]

Параметры команды следующие:

- IP - IP-адрес CA
- USERS_DN - имя LDAP-каталога, где содержится информация о пользователях; например “dc=domain,dc=int”
- BIND_DN - указывает пользователя, от имени которого будет производится поиск в LDAP-каталоге; например “cn=bob,cn=users,dc=domain,dc=int”
- PAS - пароль для BIND_DN пользователя
- NUM - число параллельных процессов аутентификации на локальной системе.
- FILTER - задает имя атрибута по которому искать пользователя в каталоге, например uid или cn; по умолчанию sAMAccountName - для поиска в каталогах Windows AD;
- SEC - время жизни сессий аутентифицировавшихся пользователей (в секундах);
- ssltls - шифровать LDAP трафик по SSL/TLS протоколу (необходима поддержка SSL/TLS на CA);
- lookup - в случае задания фильтра и данной опции среди возвращаемых DN пользователей от LDAP-сервера будет искаться строка “^CN|cn=LOGIN,”, где LOGIN - введенное в браузере имя пользователя; если данная строка найдена, то данный DN пользователя будет использован для аутентификации, если строка не найдена, то доступ будет запрещен (т.к. не найден пользователь); без этой опции для аутентификации будет использован первый пользователь в списке DN пользователей возвращенных от LDAP-сервера; в обоих случаях пароль будет использован тот, который введен пользователем в окне запроса логина и пароля в Интернет-браузере.

Для работы с Microsoft AD имена атрибутов (DC, CN, OU) должны быть прописные.

Формат значения параметра FILTER описывается языком, описанным в RFC 2254. Приведем основные особенности этого языка:

- фильтр должен начинаться и завешаться скобками: “(<ФИЛЬТР>)”

- фильтр состоит из элементов и логических операций над ними
- логические операции могут быть: !(ЭЛЕМЕНТ) - отрицание, &(ФИЛЬТРЫ) - логическое И, |(ФИЛЬТРЫ) - логическое ИЛИ;

- формат элемента: <АТРИБУТ><ОПЕРАЦИЯ><ЗНАЧЕНИЕ>, где ОПЕРАЦИЯ - это “=”, “~=”, “<=”, “>=”, т.е. равно, не равно, меньше и больше

- в значениях можно использовать символ “*“, который заменяет любое количество символов
- вместо всего ЗНАЧЕНИЯ можно указать знак “*“; такая форма используется для отбора записей, у которых присутствует определённый атрибут

Примеры фильтров:

Записи, в которых атрибут cn равен значению Ivan Petrov:

```
(cn=Ivan Petrov)
```

Записи, тип которых Class1 и в которые входит IPetrov (атрибут member):

```
(&(ObjectClass=Class1)(member=uid=IPetrov,ou=Employee,ou=Org,o=com))
```

Видим операцию AND над двумя элементами

Записи, в которых имеются члены (атрибут uniquemember):

```
(uniquemember=*)
```

Записи, в которых атрибут cn начинается на I и заканчивается на an:

```
(cn=I*an)
```

Параметры USERS_DN и BIND_DN можно вводить как в полной форме (“dc=domain,dc=int,cn=user”), так и в краткой.

Формат краткой формы - [U1[.U2[.U3…]]]@]D1[.D2[.D3…]]
В данном формате:
- U1..Un - необязательное имя пользователя, в полной форме становится CN=U1,CN=U2,…,CN=Un.
- D1..Dn - необязательное имя домена, в полной форме становится DC=D1,DC=D2,…,DC=Dn.

Рассмотрим пример:

```
(config—service—proxy)# auth ldap 10.0.0.1 factor—ts.int user admin@factor—ts.int pass123
```

Далее необходимо настроить правила аутентификации также, как это сделано в локальной аутентификации. Например:

```
(config—service—proxy)# acl allusers proxy—auth—all
(config—service—proxy)# http—access permit allusers
(config—service—proxy)# http—access deny all
```

34.9.3 Удаленная RADIUS-аутентификация

В данном случае аутентификация проходит на удаленном Radius-сервере аутентификации.

Формат команды для включения такой аутентификации:

auth radius <IP> <PAS> [timeout TO] [NUM] [credttl SEC]

Параметры команды следующие:

- IP - IP-адрес Radius-сервера
- PAS - общий секрет для связи с Radius-сервером
- TO - время ожидания выполнения запроса к Radius-серверу (в секундах)
- NUM - число параллельных процессов аутентификации на локальной системе
- SEC - время жизни сессий аутентифицировавшихся пользователей (в секундах)

34.9.4 Удаленная Kerberos-аутентификация (вариант с Windows DNS-сервером)

Kerberos — сетевой протокол аутентификации, который предлагает механизм взаимной аутентификации клиента и сервера перед установлением связи между ними, причём в протоколе учтён тот факт, что начальный обмен информацией между клиентом и сервером происходит в незащищенной среде, а передаваемые пакеты могут быть перехвачены и модифицированы.

Основные понятия протокола:

- Key Distribution Center (KDC) - хранилище информации о паролях пользователей
- Admin server - основной сервер Kerberos
- Realm - “среда”, в которой производится аутентификация (в терминах Windows - это домен службы каталогов Active Directory, далее - домен)
- Principal - принципал, т.е. пользователь или сервис, участвующий в механизме аутентификации

В системе Dionis-NX предполагается, что KDC и Admin server находятся на одной машине под управлением ОС Windows Server 2008 (или более новой).

Рассмотрим процедуру настройки Kerberos-аутентификации на примере следующего стенда:

- **Windows-клиент** - клиент, который хочет получить доступ к сети Интернет при помощи интернет-браузера Firefox;
  параметры клиента: ОС Windows 7 (или более новая), имя машины - win7, IP-адрес клиента 192.168.0.3

- **Dionis-NX** - маршрутизатор, который будет осуществлять проксирование и аутентификацию пользователя, работающего на Windows-клиенте;
  аутентификация будет происходить на KDC Windows-сервера;
  параметры маршрутизатора: ПО Dionis-NX, имя машины - gate, IP-адрес: 192.168.0.254, осуществляет доступ в Интернет других узлов

- **Windows-сервер** - сервер, который хранит информацию о паролях пользователей в службе каталогов Active Dircetory, при этом являясь также Admin/KDC-сервером в терминах Kerberos, а также DNS-сервером; параметры сервера: ОС Windows Server 2008 (или более новая), имя машины - server, IP-адрес: 192.168.0.1

Как видно из стенда все узлы домена находятся в одной сети 192.168.0.0/24. Однако в Dionis-NX имеется также и второй интерфейс (сеть 192.168.33.0/24) для выхода в сеть Интернет.

Общая схема работы стенда:

34.9.4.1 Общая схема работы Kerberos

Рассмотрим общую схему работы Kerberos-аутентификации в Active Directory через прокси-сервер на основе вышеописанного стенда:

Предположим, что Windows-сервер настроен следующим образом:

1. установлена служба Active Directory, в которую добавлен домен DOMAIN и пользователь user в этом домене (с некоторым паролем pass)
2. система является KDC и Admin server с точки зрения Kerberos

3. система является контроллером домена DOMAIN

Краткое описание алгоритма аутентификации:

1. доменный пользователь **user** вводит свой пароль (например, **pass**) для входа на системе Windows-клиента в домен **DOMAIN**.
2. клиент: система Windows-клиент посылает запрос на контроллер домена (систему Windows-сервер) на получение TGT (Ticket-Grant-Ticket, т.е. “Билет для получения билетов”), в этом запросе содержится имя пользователя **user**

3. сервер: если пользователь найден, сервер посылает TGT клиенту; TGT, упрощенно, состоит из: **UserPass(SessKey, AdPass(SessKey))**, т.е. паролем пользователя зашифрована пара {случайный сессионный ключ; зашифрованный паролем администратора домена сессионный ключ}

4. клиент: если клиент может расшифровать пару TGT и получить **SessKey**, то он понимает, что аутентификация прошла успешна - пароль, предоставленный пользователем, верный

5. клиент: пользователь успешно вошел в систему и пытается войти в интернет через прокси-сервер **gate**

6. клиент: прокси-сервер должен каким-то образом понимать, что запросы через него идут от аутентифицированного пользователя, т.е. прокси-сервер должен выполнить аутентификацию пользователя на контроллере домена, используя при этом базу данных пользователей Active Directory и протокол Kerberos; с точки зрения Kerberos эта задача сводится к взаимодействию принципалов: пользователя и прокси-сервиса
7. маршрутизатор: с точки зрения Kerberos прокси-сервис - это принципал-сервис, работающий на некоторой системе gate, по аналогии с принципалом-пользователем user, работающем на системе windows-клиент; о принципале-пользователе сервер уже знает, осталось сообщить ему о принципале-сервисе, работающем на системе **gate** в домене **DOMAIN**

8. сервер: добавить пользователя **proxy** с некоторым паролем **proxypass**

9. сервер: связать пользователя **proxy** с принципалом-сервисом **HTTP/gate.domain@DOMAIN**, где HTTP - тип сервиса, gate.domain - имя маршрутизатора, DOMAIN - имя домена; связка происходит программой **ktpass**, которая выдает в результате ключ для принципала-сервиса HTTP/gate.domain@DOMAIN; этот ключ копируется на маршрутизатор и далее будет называться **keytab**
10. маршрутизатор: настроить клиентскую библиотека Kerberos, указав имя домена и имя KDC-сервера; теперь прокси-сервис от имени принципала-сервиса HTTP/gate.domain@DOMAIN может посылать запросы на сервер для получения TGT, как это делал ранее обычный пользователь в системе windows-клиент (п.1)

11. клиент: пользователь запускает браузер, который настроен на использование прокси-сервера **gate.domain**; браузер, послав полученный ранее **AdPass(SessKey)** и имя пользователя **user**, запросит у сервера билет на доступ к **HTTP**

12. сервер: в п.8 программой **ktpass** произошла связка принципала-сервиса **HTTP/gate.domain**, имеющим тип **HTTP**, с пользователем **proxy**, поэтому сервер знает, что для выдачи билета на доступ к **HTTP** необходим пароль пользователя **proxy**, т.к. именно он отображен на принципала **HTTP/gate.domain**

13. сервер: сервер расшифрует своим паролем администратора домена последовательность **AdPass(SessKey)** и получит **SessKey**;

14. сервер: зашифрует сессионным ключом зашифрованное паролем принципала-сервиса proxy имя пользователя user: **SessKey(proxypass(user))**

15. сервер: пошлет **SessKey(proxypass(user))** клиенту

16. клиент: получив **SessKey(proxypass(user))**, браузер расшифровывает сессионным ключом

последовательность **proxypass(user)** и отправляет ее на маршрутизатор прокси-сервису

17. маршрутизатор: прокси-служба получив **proxypass(user)** расшифрует его при помощи ключа **keytab** и получит имя пользователя

18. маршрутизатор: таким образом, прокси-служба понимает, что имеет дело с аутентифицированным пользователем и предоставляет ему дальнейший доступ к Интернет

34.9.4.2 Предварительная настройка

Перед настройкой узлов стенда необходимо учесть следующие важные особенности:

- время на всех узлах домена (на клиенте, сервере и маршрутизаторе) должно быть синхронизировано и одинаково

- все узлы домена должны иметь FQDN-имена (возможно, помимо сокращенных имен) и должны уметь обращаться друг к другу не только по IP-адресу, но и по FQDN-имени

34.9.4.3 Настройка Windows-сервера. Шаг 1.

Рассмотрим настройку Windows-сервера для стенда сразу после установки новой системы Windows Server 2008.

После выполнения некоторых пунктов, если система попросит перезагрузку, необходимо ее выполнить и продолжить настройку после перезагрузки.

1. Свойства компьютера, Изменить имя компьютера на **server**

2. Свойства сетевого адаптера, Задать статические настройки внутреннего интерфейса:
    - IP-адрес: 192.168.0.1
    - шлюз по-умолчанию: 192.168.0.254
    - DNS-сервера: 192.168.0.254, 192.168.0.1

3. Пуск, Администрирование, Диспетчер сервера, Роли, Добавить роль:
    - DNS-сервер

4. Пуск, Администрирование, Диспетчер сервера, DNS-сервер, Настроить DNS:
    - создать прямую зону (первичную) **office.local**
    - создать обратную зону (первичную) **192.168.0**

- запретить Динамические обновления

5. Пуск, Администрирование, Диспетчер сервера, Компоненты, Добавить компонент:
   - Инфраструктура **.Net Framework**

6. Пуск, Администрирование,Диспетчер сервера, Роли, Добавить роль:
   - Службы каталогов **Active Directory**

7. Пуск, Консоль Администратора, **dcpromo.exe**:
   - Новый домен в новом лесу: **office.local**
   - Режим леса: Windows 2008 (выбираем в соответствии со своей ОС сервера) debug
   - перезагрузить сервер

8. Пуск, Администрирование, Диспетчер сервера, Роли, DNS: Прямая зона, office.local зона, Свойства зоны, Общие:
   - Тип, Изменить, **Хранить зону в Active Directory**
   - Динамические обновления: **Только безопасные**

9. Пуск, Администрирование, Диспетчер сервера, Роли, DNS: Прямая зона, office.local зона, Свойства зоны, Сервера имен, server, Изменить:
   - изменить имя сервера на **server.office.local**
   - Разрешить(Resolve) имя сервера

10. Пуск, Администрирование, Управление групповой политикой, Лес, Домен, office.local, Политика домена по-умолчанию, Настройки, Править, Конфигурация Компьютера, Политики, Настройки Windows, Настройки безопасности, Настройки учетных записей, Проверка пароля:
    - устанавливаем необходимые параметры сложности пароля (минимальная длина, максимальное время использования и т.д.)

11. Пуск, Консоль Администратора, **gpupdate.exe**:
    - произойдет обновление групповой политики безопасности (измененной в п.10)

12. Пуск, Администрирование, Пользователи и компьютеры Active Directory, Пользователи, Новый, Пользователь:
    - задаем имя и логин нового пользователя, например **oleg**
    - пароль нового пользователя, удовлетворяющий параметрам сложности паролей, установленным в п.10, например пароль "**o**"

34.9.4.4 Настройка Windows-клиента. Шаг 1.

Рассмотрим настройку Windows-клиента для стенда сразу после установки новой системы Windows 7.

После выполнения некоторых пунктов, если система попросит перезагрузку, необходимо ее выполнить и продолжить настройку после перезагрузки.

1. Свойства компьютера, Изменить имя компьютера:
   - имя компьютера: **win7**
   - домен: **office.local**
   - потребуется ввести пароль Администратора домена (пользователь Администратор и его локальный пароль на Windows-сервере)
2. Свойства сетевого адаптера, Задать статические или динамические настройки внутреннего интерфейса, например:
   - IP-адрес: 192.168.0.3
   - шлюз по-умолчанию: 192.168.0.254
   - DNS-сервера: 192.168.0.254, 192.168.0.1
3. Войти в систему доменным пользователем:
   - пользователь **oleg@OFFICE.LOCAL**
   - пароль "**o**", установленный в п.12 "Настройка Windows-сервера. Шаг 1."
4. (Необязательно) Сеть, Включить сетевой обнаружение, потребуется ввести пароль Администратора домена (пользователь Администратор и его локальный пароль на Windows-сервере)

34.9.4.5 Настройка Windows-сервера. Шаг 2.

Продолжаем настройку Windows-сервера, осуществив успешный вход доменным пользователем на машину win7 (см. п.3 "Настройка Windows-клиента. Шаг 1.")

1. Пуск, Администрирование, Диспетчер сервера, Роли, Active Directory, Пользователи и компьютеры:
   - проверяем наличие машины **WIN7** в списке узлов домена (это можно проверить уже после п.1 "Настройка Windows-клиента. Шаг 1.")

2. Пуск, Администрирование, Диспетчер сервера, Роли, DNS:

   - проверяем наличие записей для **win7.office.local** в прямой зоне office.local и обратной зоне (это можно проверить уже после п.1 "Настройка Windows-клиента. Шаг 1.")

3. Пуск, Администрирование, Диспетчер сервера, Роли, DNS: Прямая зона, office.local зона, Новая A/AAAA-запись:

   - создадим A-запись для маршрутизатора: имя узла **gate**, полное имя **gate.office.local**, IP-адрес 192.169.0.254

4. Пуск, Администрирование, Пользователи и компьютеры Active Directory, Пользователи, Новый, Пользователь:

   - заведем псевдо-пользователя, например **proxy**

   - зададим пароль псевдо-пользователя, удовлетворяющий параметрам сложности паролей, установленным в п.10 "Настройка Windows-сервера. Шаг 1.", например пароль "**p**"

5. Свяжем псевдо-пользователя **proxy** с принципалом сервиса HTTP, который называется **HTTP/gate.office.local**: Пуск, Консоль Администратора:

   - **ktpass.exe** -princ HTTP/gate.office.local@OFFICE.LOCAL -mapuser proxy -pass p -out proxy.keytab

6. Скопировать полученный в п.5 файл **proxy.keytab** безопасным способом (например, утилитой WinSCP) на маршрутизатор **gate**.

34.9.4.6 Настройка Dionis-NX маршрутизатора.

Получив файл таблицы Kerberos-ключей от Windows-сервера, настроим Dionis-NX маршрутизатор.

1. Зададим общие настройки системы:

   - имя узла должно быть в формате FQDN
   - сервера имен необходимы для работы службы proxy
   - включим маршрутизацию транзитных IP-пакетов
   - (необязательно) зададим имя домена для разрешения имен маршрутизатором, равным имени Windows-домена

   - (необязательно) статические записи для **gate.office.local** и **server.office.local** можно не задавать, если маршрутизатор может их разрешить другим способом (например, через Windows-сервер или записи в локальной службе DNS).

Конфигурация:

```
hostname gate.office.local
ip resolver domain office.local
ip resolver nameserver 192.168.0.254
ip resolver nameserver 192.168.0.1
ip resolver nameserver 192.168.33.254
ip resolver host 192.168.0.1 server.office.local server
ip resolver host 192.168.0.254 gate.office.local gate
ip forwarding
```

2. Настроим интерфейсы и NAT:

   - настроим трансляцию адресов NAT для внутренней сети домена (сеть 192.168.0.0/24)
   - интерфейс ethernet0 используется для выхода в Интернет (сеть 192.168.33.0/24)
   - интерфейс ethernet1 используется для работы внутренней сети домена (сеть 192.168.0.0/24)
   - маршрут по-умолчанию обслуживает запросы на все адреса, кроме адресов из сети 192.168.0.0/24

Конфигурация:

```
ip nat—list masq
 1 nat src 192.168.0.0/24 masquerade

interface ethernet 0
 ip address dhcp iponly
 ip nat—group masq
 enable
```

3. Настройка синхронизации времени:

```
service ntp
 server 0.ru.pool.ntp.org
 server 1.ru.pool.ntp.org
 server 2.ru.pool.ntp.org
 server 3.ru.pool.ntp.org
 enable
```

4. (Необязательно) В зависимости от схемы, настройка DNS:

```
service dns
 listen 192.168.0.254
 1 view default
  zone .
   auto static
 enable
```

5. Произведем импорт файла Kerberos-ключей, полученного в п.6 "Настройка Windows-сервера. Шаг 2.":

   - (необязательно) очистим таблицу ключей: **do clear proxy keytab all**
   - импортируем новый ключ: **service proxy import keytab file:/proxy.keytab**
   - (необязательно) проверим таблицу ключей: **show service proxy keytab**
   - (необязательно) проверим связь маршрутизатора gate с сервером AD/KDC (server) по протоколу Kerberos, получив от него билет TGT для пользователя oleg:
     - **service proxy kerberos-user-check** office.local server oleg o
     - команда должна вернуть информацию о выданном TGT билете для пользователя oleg
     - успешное выполнение данной команды обязательно для работы Kerberos

6. Настройка службы proxy (простейший вариант для стенда):

   - команда **auth kerberos** в указанной ниже конфигурации задает следующие параметры:
     - имя домена и среды (realm): **office.local**
     - KDC и Admin Server: **server.office.local** (часть имени office.local в нашем случае будет добавлена автоматически)

Конфигурация:

```
service proxy
 listen 192.168.0.254 3128
 acl auth proxy—auth—all
 log access
 cache type aufs 16 32
 http—access permit auth
 http—access deny all
 auth kerberos office.local server HTTP/gate.office.local
 enable
```

34.9.4.7 Настройка Windows-клиента. Шаг 2.

На данном шаге осуществим непосредственно доступ к сети Интернет посредством браузера (например, Firefox) через маршрутизатор **gate.office.local**.

1. Открыть браузер Firefox, Настройки, Параметры сети:

   - задать имя Proxy-сервера: gate.office.local
   - задать порт Proxy-сервера: 3128

2. Осуществить вход в Интернет посредством браузера

34.9.5 Удаленная Kerberos-аутентификация (вариант с Dionis-NX DNS-службой)

Данный вариант настройки предполагает, что DHCP- и DNS-служба установлены только на марш-рутизаторе DionisNX.

На сервере Windows установлена только служба каталогов Active Directory.

В этом случае схема настройки повторяет предыдущую, за исключением следующих далее измене-ний.

34.9.5.1 Изменения в настройке Windows-сервера

1. Не нужно добавлять роль DNS, либо удалить ее, если она установлена

2. Не нужно добавлять роль DHCP, либо удалить ее, если она установлена

3. Настроить интерфейс: адрес шлюза по-умолчанию и адрес DNS-сервера по-умолчанию устано-вить на IP-адрес маршрутизатора Dionis-NX

34.9.5.2 Изменения в настройке Dionis-NX - маршрутизатора

1. Зададим общие настройки системы:

    - имя узла должно быть в формате FQDN
    - сервер имен необходим для работы службы proxy
    - включим маршрутизацию транзитных IP-пакетов
    - зададим имя домена для разрешения имен маршрутизатором, равным имени Windows-домена

Конфигурация:

```
hostname gate.office.local
1 ip resolver domain office.local
1 ip resolver nameserver 192.168.0.254
```

2. Настройка DHCP-службы.

3. Настройка DNS-службы:

Конфигурация:

```
service dns
 1 acl ad 192.168.0.1
 listen 192.168.0.254
 1 view default
  zone .
```

```
  auto static
 zone master office.local.
  soa master gate
  a 192.168.0.1 server
  a 192.168.0.254 gate
  ns gate
  update acl ad
 zone master 0.168.192.in—addr.arpa.
  soa master gate.office.local.
  ns gate.office.local.
  ptr 1 server.office.local.
  ptr 254 gate.office.local.
  update acl ad
enable
```

Как видно из настройки DNS, мы разрешаем обновления ресурсных записей в прямой и обратной доменной зоне Windows-сервером. Эти ресурсные записи необходимы для работы различных подсистем Windows-сервера. Создание обратное зоны не является обязательным, но рекомендуется для работы с Windows-доменом.

34.9.6 Удаленная Kerberos-авторизация по доменным группам

Помимо аутентификации, можно настроить авторизацию пользователей на основе групп пользователей. В Windows-домене пользователи входят в группы пользователей.

Пример конфигурации с включенным режимом отладки Kerberos, доступ разрешен только пользователям из группы Vip и группы Other.

```
service proxy
 listen 192.168.0.254 3128
 log kerberos
 acl vip krb—ldap—group office.local Vip Other
 acl users krb—ldap—group office.local ”Domain Users”
 cache type aufs 16 32
 http—access permit vip
 http—access deny users
 auth kerberos office.local server.office.local HTTP/gate.office.local 3 no—keep—alive
 enable
```

Замечание: особенность задания acl типа krb-ldap-group: если имя группы содержит пробел, то можно указать только одну группу при задании acl.

34.10 Настройка контроля пропускной способности сети

Данный раздел относится к п.8 Алгоритма.

Служба может контролировать скорость потока данных от удалённых серверов к пользователям: ограничения накладываются **только** на кэш-промахи, кэш-попадания не ограничиваются по скорости.

Контроль пропускной способности сети действует при помощи пулов задержки.

Для лучшего понимания настройки рассмотрим механизм пулов задержки на основе аналогии с ведрами и кранами.

Существует 4 типа ведер:

- общее ведро: может ограничивать весь трафик (тип agr);
- общее ведро, разделенное на 256 индивидуальных ведер: может ограничивать весь трафик и трафик по каждому из 256 узлов (тип agr-host24);
- общее ведро, разделенное на 256 сетевых ведер, разделенных на 256 индивидуальных ведер: может ограничивать весь трафика, трафик по каждой из 256 С-сетей и трафик по каждому из 65536 узлов (т.е. 256 узлов из 256 сетей) (тип agr-net24-host16);
- аналогично предыдущему типу, только добавляется контроль трафика по каждому аутентифицированному пользователю (тип agr-net24-host16-user).

Рассмотрим алгоритм работы контроля скорости трафика на основе общего ведра.

Весь входящий внешний трафик заполняет общее ведро. В этом общем ведре есть один большой кран, из которого пользователи службы получают данные удаленных серверов, которые наполняют ведро своим трафиком.

У ведер есть параметры: размер и скорость наполнения. Размеры и скорость указываются в байтах и байтах в секунду соответственно. Возможно использование префиксов kb,mb,gb - для кило,мега и гигабайт (размер умножается на 10^3,10^6 и 10^9 соответственно).

Как только клиенты службы через один большой кран сольют себе весь трафик из ведра (в случае если их общая скорость слива превысит скорость наполнения ведра) и ведро перестанет успевать наполняться, включается общее ограничение (если оно задано) на скорость подачи трафика из крана,которое становится равным скорости наполнения ведра. Если скорость слива ниже скорости наполнения, ведро будет заполняться (до установленного объёма).

Чтобы настроить общее ограничение, следует выполнить команду:

```
(config—service—proxy)# delay—pool pool1 agr 1mbs/100mb
```

Данное ведро имеет размер 100Мб и скорость наполнения 1mbs. Пока оно не пусто - клиенты не ограничены. Как только оно становится пустым, общая скорость для всех клиентов падает до 1mbs.

Механизм для остальных ведер аналогичен. Отличие только в размере и скорости наполнения ведер: например, по каждой сети типа С, по каждому узлу, по каждому пользователю.

Рассмотрим пример:

```
(config—service—proxy)# delay—pool pool2 agr—host24 10mbs/100mb 1mbs/10mb
(config—service—proxy)# delay—pool pool3 agr—net24—host16 10mbs/100mb 1mbs/10mb 100kbs/1mb
(config—service—proxy)# delay—pool pool4 agr—net24—host16—user 10mbs/100mb 1mbs/10mb
        100kbs/1mb 50kbs/1mb
(config—service—proxy)# delay—pool pool5 agr—host24 10mbs/100mb —1/—1
```

Рассмотрим подбробно, какие ведра определены данными командами:

- pool2: общее ведро размером 100Мб и скоростью наполнения 10Мбс, внутри которого есть 256 ведер 1Мбс/10Мб для каждой С-сети;
- pool3: общее ведро размером 100Мб и скоростью наполнения 10Мбс, внутри которого есть 256 ведер 1Мбс/10Мб для каждой С-сети, и в каждом из таких ведер есть 256 ведер, задающих ограничение 100kbs/1mb на каждый хост;
- pool4: аналогично pool3, только кроме этого задается ведро 50kbs/1mb на каждого пользователя;
- pool5: аналогичен pool2, только ограничение на С-сеть не накладывается (-1 обозначает отсутствие ограничения).

Разрешающие правила (delay-access permit) ограничивают трафик ведром, указанным в правиле, для пользователей, попадающих под действие указанного в правиле ACL. Если такое правило найдено - просмотр правил прекращается и правило применяется.

Запрещающие правила (delay-access deny) не ограничивают трафик ведром, указанным в правиле, для пользователей, попадающих под действие указанного в правиле ACL. Если такое правило найдено - просмотр правил для ведра данного типа прекращается, однако продолжается для ведер других типов.

34.11 Правила проверки объектов в кэше на свежесть

Данный раздел относится к пп.6-8 Алгоритма.

Служба имеет возможность настраивать правила, по которым те или иные объекты кэша будут считаться службой свежими или несвежими.

Свежесть объекта, в свою очередь, определяет, необходимо ли обновлять в кэше объект с удаленных серверов или нет.

Говоря кратко, чем чаще объекты определяются как свежие, тем больше будет процент попадания в кэш и ,как следствие, большая скорость обслуживания клиентов.

Далее под ресурсом будем понимать именованную последовательность данных, расположенную на HTTP-сервере и доступную по ее URI.

Под объектом будем понимать размещенный в кэше службы ресурс,полученный с HTTP-сервера.

Перед рассмотрением правил напомним и введем несколько понятий.

Заголовки HTTP, необходимые для понимания правил:

- Date : дата генерации HTTP-ответа сервером;
- Last-Modified : дата последней модификации ресурса;
- Expires : дата предполагаемого истечения срока актуальности ресурса;
- If-Modified-Since : выполнять указанный HTTP-метод если ресурс изменился с указанного момента.

Директивы HTTP-заголовка Cache-Contol, который управляет кэшированием:

- no-cache : сервер не должен использовать кэшированный ответ;

- no-store : ответ на этот запрос не должен кэшироваться;
- max-age : максимальное время хранения объекта в кэше;
- must-revalidate : если кэшированный ответ устарел, то должен быть обновлен, вне зависимости от правил кэш-службы.

Другие термины:

- возраст объекта (ВО) - это разница между Date и Last-Modified объекта; Date обновляется, а Last-Modified может обновиться во время последнего запроса объекта с HTTP-сервера;
- возраст реакции (ВР) - это степень несвежести объекта, т.е. время прошедшее с момента последнего запроса объекта клиентом; иначе говоря, это время с момента последней валидации службой объекта на свежесть: возврата свежего объекта из кэша, получения отсутствующего или обновление несвежего объекта с сервера;
- LM-фактор - это отношение ВР к ВО.

Рассмотрим, как создаются правила проверки на свежесть:

```
(config—service—proxy)# refresh \\.jpg$ 1000 70% 5000
```

Все параметры команды refresh обязательны. Разберем их по порядку появления в строке команды:

- \\.jpg$ - это РВ, задающее URI объектов, для которых будет применяться данное правило; в данном случае это картинки с расширением jpg;
- 5000 - это максимальный ВР в минутах (ВРМАКС); объект несвеж, если его ВР больше ВРМАКС, в данном случае 5000 мин;
- 70% - это максимальный LM-фактор (ЛММАКС); объект несвеж, если его LM-фактор больше ЛММАКС, в данном случае 70%;
- 1000 - это минимальный ВР в минутах (ВРМИН); объект свеж, если его ВР меньше ВРМИН, в данном случае 1000 мин.

Правила проверки на свежесть будут работать только для объектов, у которых не указан Expires, если обратное не указано специальной опцией правила (см.ниже).

Чем больше прошло времени с момента последней проверки на свежесть, тем менее свеж объект и тем больше LM-фактор. Когда LM превысит значение, заданное в команде, объект перестанет быть свежим.

Порядок правил важен - как только запрашиваемый объект попадет в одно из правил, поиск правил будет остановлен для данного объекта.

Рассмотрим алгоритм проверки на свежесть в случае,если не заданы дополнительные опции правила:

1. проверка по Expires,если он есть в пакете: если дата прошла,то объект не является свежим; если дата еще не наступила,то объект является свежим; выход;
2. если ВР больше чем ВР_МАКС, то объект не является свежим и требуется его обновление с удаленного сервера;

3. если ВР меньше чем ВР_МАКС, то осуществляется проверка по LM-фактору: если объект является свежим - выход;
4. если ВР меньше ВР_МИН, то несвежесть по LM-фактору не убедительна: объект был совсем недавно получен и мог быть создан также совсем недавно,в результате LM-фактор стал высоким; т.о если ВР меньше ВР_МИН,то объект является свежим - выход;
5. иначе, несвежесть по LM-фактору убедительна, объект признается устаревшим и необходимо его обновление с удаленного сервера.

Рассмотрим дополнительные опции правила, которые могут изменить вышеописанный алгоритм:

- override-expire : игнорировать заголовок Expires;
- override-lastmod : игнорировать заголовок Last-Modified;
- reload-into-ims : использовать заголовок If-Modified-Since (только при наличии Last-Modified) вместо директивы no-cache;
- ignore-no-cache : игнорировать директиву no-cache;
- ignore-no-store : игнорировать директиву no-store;
- ignore-reload : игнорировать директивы no-cache и reload;
- ignore-must-revalidate : игнорировать директиву must-revalidate;
- refresh-ims : всегда обновлять объект, вне зависимости от наличия If-Modified-Since в запросе клиента.

Пример использования дополнительных опций:

```
(config—service—proxy)# refresh \\.jpg$ 10000 70% 20000 reload—into—ims ignore—reload
```

34.12 Настройка адаптации содержимого

Служба имеет возможность анализировать, блокировать или изменять содержимое проксируемых через нее сообщений. Это называется адаптация содержимого, хотя в некоторых случаях никакого изменения содержимого не производится.

Для настройки адаптации содержимого необходимо указать сервер адаптации:

adapt-srv NAME <icap|ecap> <request|response> <URL:PORT[URN]> [bypass] [routing] on-overload TYPE [CONLIM]

Параметры команды:

- NAME - краткое имя-идентификатор задаваемого сервера адаптации
- icap|ecap - протокол сервера адаптации (ICAP либо ECAP)
- request|response - задает тип сообщений, обрабатываемых данным сервером - запросы или ответы.

- URL:PORT[URN] - задает имя домена сервера адаптации и его порт, а также URN - необязательное имя ресурса (путь к приложению на сервере адаптации)

- bypass - этот сервер является необязательным, в случае его недоступности,можно использовать следующий сервер в цепочке серверов или в множестве серверов (см.ниже)
- routing - динамическое изменение плана адаптации содержимого текущего обрабатываемого сообщения
- TYPE - (только для icap) задает что действие при достижении максимального числа соединений: block(послать клиенту сообщение об ошибке), bypass (не использовать перегруженный сервер адаптации), wait (ожидать уменьшения числа соединений), force(игнорировать превышение и продолжать использовать указанный сервер адаптации)
- ipv6 - (только для icap) использовать IPv6 для соединения с сервером адаптации

Следующая команда задает именованное множество (с именем NAME) серверов адаптации:

adapt-set NAME {SRV,5}

Множество используется для повышения доступности адаптационного механизма - если один из серверов множества недоступен, сообщение передается следующему в списке серверу.

Следующая команда задает именованную цепочку (с именем NAME) серверов адаптации:

adapt-chain NAME {SRV,5}

Цепочка используется для множественной обработки сообщения разными серверами адаптации, т.е. сначала сообщение обрабатывается 1-м в цепочке сервером, затем результат обработки передается 2-му серверу и т.д.

Можно также задать правила адаптации, описывающие связь между серверами адаптации и ACL:

adapt-access <permit|deny> <NAME> {ACL,5}

Команда задает правило использования сервера адаптации (а также цепочки или множества серверов адаптации) NAME при попадании сообщения в указанные ACL.

Для задания прочих опций адаптации содержимого используйте следующую команду:

adapt <send-username HNAME [encode]| send-client-ip | icap-srvfail-limit N | icap-io-timeout N | icap-connect-timeout N | icap-service-revival-delay N>

Параметры:

- send-username - задать имя HTTP-заголовка в котором будет пересылаться имя аутентифицированного пользователя; encode - закодировать имя пользователя алгоритмом BASE-64

- send-client-ip - сообщать серверам адаптации IP адрес клиента
- icap-srvfail-limit - максимальное число попыток установления TCP-соединения с ICAP-сервером адаптации
- icap-io-timeout - максимальное время ожидания передачи данных по установленному ICAP-соединению
- icap-connect-timeout - как долго ждать установления TCP-соединения с ICAP-сервером
- icap-service-revival-delay - как долго ждать после запроса опций ICAP, прежде чем послать новый запрос опций ICAP.

34.13 Прочие настройки службы

Как долго ждать выполнения запроса на TCP-соединения (в сек.):

```
(config—service—proxy)# timeout connect \<VAL\>
```

По умолчанию: 60 сек.

Как долго ждать ответа на DNS-запрос (в сек.):

```
(config—service—proxy)# timeout dns \<VAL\>
```

По умолчанию: 30 сек.

Использовать IPv4 DNS-сервера в первую очередь и только затем IPv6 DNS-сервера:

```
(config—service—proxy)# ipv6 dns4—first
```

Максимальное количество открытых службой файлов:

```
(config—service—proxy)# limit filedesc \<VAL\>
```

По умолчанию: 4096

34.14 Настройка журналов службы

Служба имеет систему журналов. Всего их существует 5 типов:

- cache: общий журнал службы; имеет три уровня подробности журналирования, причем каждый следующий уровень включает сообщения предыдущего:
  - low: : сообщения о критических и фатальных ошибках;
  - middle: : сообщения о важных проблемах службы, предупреждения (по-умолчанию);
  - high : сообщения о небольших проблемах службы и высоко-уровневые операции службы;
- access : информация о доступе к службе и клиентских запросах;
- store : информация об объектах, сохраняемых в кэше и удаляемых из него;
- useragent : информация о клиентских браузерах (на основе заголовка User-Agent HTTP-запроса);
- referer : информация о доменах, посещаемых клиентами (на основе заголовка Referer HTTP-запроса);
- icap [mod-only] : лог ICAP-транзакций, опционально - в лог попадают только модифицированные сервером адаптации запросы

При нормальной работе системы не следует устанавливать уровень cache журнала выше middle, чтобы не замедлять работу системы.

Обычно, если нет проблем в работе службы, достаточно бывает следующих журналов:

```
(config—service—proxy)# log access
(config—service—proxy)# log cache
```

Для отправки журнала доступа на удаленный syslog-сервер введите команду:

```
(config—service—proxy)# log access—remote <PROTO> <HOST:PORT>
```

Также можно отключить журнал доступа, оставив только общий журнал службы (cache), который будет работать по умолчанию в режиме middle.

Формат строки журнала доступа для команды **log access** следующий:

```
logformat squidnx %tl %un %>a %rm %ru (%>ru) %>st —> %tr %Ss/%03>Hs/%03<Hs %<st
        %Sh/%<a(%<A) %mt
```

Формат строки журнала доступа для команды **log access ssl** следующий:

```
logformat squidnx %tl %un %>a %rm %ru (%>ru) %>st —> %tr %Ss/%03>Hs/%03<Hs %<st
        %Sh/%<a(%<A) %mt [ssl: %ssl::bump_mode/%ssl::>sni]
```

Формат строки журнала доступа для команды **log icap** следующий:

```
logformat icapnx %tl %un %>a %icap::rm %ru (%icap::>st b/%icap::tio ms) ——> %icap::ru
        %icap::<service_name ——> %icap::to %03icap::Hs (%icap::<st b/%icap::tr ms)
```

Для просмотра журналов службы следует выполнить команду:

```
(config—service—proxy)# do show service proxy log
```

По умолчанию будет показан общий журнал службы.

Чтобы посмотреть, например, журнал доступа, следует выполнить команду:

```
(config—service—proxy)# do show service proxy log access
```

Иногда бывает необходимо вести журналы клиентских запросов, но делать это не для всех клиентов. Для этого можно задать ACL в команде log access. Рассмотрим пример:

```
(config—service—proxy)# acl vip1 src 10.0.0.1 10.0.0.2
(config—service—proxy)# log access acls vip1
```

Данные правила разрешают вести журнал доступа только для узлов с адресами 10.0.0.1,10.0.0.2.

Существует возможность скрыть IP-адреса из log access: (config-service-proxy)# acl vip1 src 10.0.0.1 10.0.0.2 (config-service-proxy)# log access 24 acls vip1

Данные правила говорят делать операцию AND между IP-адресом клиента и маской 255.255.255.0 (соответствует маске 24), т.е. младшая часть адреса будет обнулена в логе.

По умолчанию в лог access не пишутся детали запросов клиентов. Существует возможность отключить это параметром query-terms команды log access:

```
(config—service—proxy)# log access query—terms
```

Рассмотрим формат строки журнала доступа на следующем примере:

```
16/Apr/2012:16:50:32 +0400  555 192.168.33.116 TCP_MISS/200 250 GET
        http://notify5.dropbox.com/subscribe? — DIRECT/199.47.216.147 text/plain
```

Рассмотрим поля строки журнала по порядку слева направо:

- 16/Apr/2012:16:50:32 +0400 - время (UTC) завершения обработки запроса службой;
- 555 - продолжительность (в милисекундах) обработки запроса;
- 192.168.33.116 - адерс клиента;
- TCP_MISS/200 - код результата обработки запроса службой (коды перечислены ниже) и код HTTP-ответа;
- 250 - полный размер в байтах HTTP-пакета (размер HTTP-заголовков и размер,указанный в Content-Length);
- GET -метод HTTP;
- http://notify5.dropbox.com/subscribe? - URI запроса;
- DIRECT - код узла, обозначающий,что запрос перенаправляется службой напрямую на удаленный сервер; NONE - запрос не перенаправляется никуда;
- 199.47.216.147 - адрес удалённого сервера, в случае если код узла - DIRECT;
- text/plain - тип содержимого HTTP-ответа, берется из Content-Type заголовка HTTP-ответа.

Коды результата обработки запроса службой приведены ниже:

- TCP_HIT : в кэше найдена свежая копия ресура; копия возвращена клиенту;
- TCP_MISS : в кэше не найдена копия ресурса;
- TCP_REFRESH_HIT : в кэше найдена возможно несвежая копия ресурса; послан запрос валидации ресурса на УС; УС вернул ответ «Not Modified»,т.е. изначально копия была свежа;
- TCP_REF_FAIL_HIT : в кэше найдена возможно несвежая копия ресурса; послан запрос валидации ресурса на УС; УС не вернул ответ или вернул непонятный службе ответ; служба возвратила клиенту возможно несвежую копию ресурса;
- TCP_REFRESH_MISS : в кэше найдена возможно несвежая копия ресурса; послан запрос валидации ресурса на УС; УС вернул возвратил обновленные данные ресурса, т.е. копия в кэше была действительно несвежая;
- TCP_CLIENT_REFRESH_MISS : в кэше найдена копия ресурса; клиентский запрос содержал «Cache-Control: no-cache»; запрос перенаправлен УС, т.е. копия в кэше была принудительно обновлена;
- TCP_IMS_HIT : клиент прислал запрос валидации ресурса, т.к. он его уже имеет, при помощи IfModifiedSince; в кэше найдена более свежая копия; копия возвращена клиенту;
- TCP_SWAPFAIL_MISS : в кэше найдена свежая копия ресура; служба не смогла загрузить копию из своего кэша; послан запрос на УС,как будто это был кэш-промах;
- TCP_NEGATIVE_HIT : отрицательный HTTP-ответ (например «Connection refused» или «404 Not Found») был ранее закэширован; клиенту возвращён закэшированный отрицательный ответ;
- TCP_MEM_HIT : в памяти найдена свежая копия ресурса и возвращена клиенту;
- TCP_DENIED : запрос клиента запрещен из-за правил доступа (http-access или http-reply-access);
- TCP_OFFLINE_HIT : если включен режим offline, любой ресурсы,найденный в кэше возвращается клиенту,без проверки на свежесть;
- NONE - другой результат; используется при разного рода ошибках в запросе, например непонятный URI ресурса.

34.15 Работа со службой

Далее будут описаны основные команды режима enable, нужные для текущей работы со службой proxy.

34.15.1 Информационные команды.

Для просмотра статуса и правильности настройки службы следует выполнить команду:

```
(config—service—proxy)# do show service proxy status
```

Строка «Info: config is ok» говорит о том, что все в порядке. Любые строки, начинающиеся с «Error: » говорят об ошибках.

Чтобы посмотреть различную информацию о работе службы:

```
(config—service—proxy)# do show service proxy info
```

Существует много типов информации, которую можно посмотреть, используя данную команду. Эта команда описана более подробно в следующем подразделе «Мониторинг службы».

34.15.2 Кэш объектов службы.

Чтобы удалить объект из кэша службы по его URI, следует выполнить команду:

```
(config—service—proxy)# do service proxy remove—object http://example.com/pic.jpg
```

Команда будет выполнена, только если служба включена.

Очистка всего кэша:

```
(config—service—proxy)# do service proxy remove—object all
```

Команда будет выполнена, только если служба отключена.

Обновление объекта в кэше:

```
(config—service—proxy)# do service proxy reload—object http://example.com/pic.jpg
```

Команда будет выполнена, только если служба включена.

34.15.3 Сертификаты.

Экспорт сертификата службы как есть, либо в DER-формате (для импорта в браузеры клиентов):

```
(config—service—proxy)# do service proxy export cert FILE [der]
```

Импорт сертификата для использования службой при проксировании HTTPS-трафика:

```
(config—service—proxy)# do service proxy import cert FILE
```

Создать новый сертификат для использования службой при проксировании HTTPS-трафика:

```
(config—service—proxy)# do clear proxy cert
```

Экспортировать архив (tar.gz,tar.bz2), состоящий из УЦ-сертификатов и, возможно, CRL файлов, необходимых для проверки сертификатов клиентов:

```
(config—service—proxy)# do service proxy export certd FILE
```

Импортировать архив (tar.gz,tar.bz2), состоящий из УЦ-сертификатов и, возможно, CRL файлов, необходимых для проверки сертификатов клиентов:

```
(config—service—proxy)# do service proxy import certd FILE
```

Удалить из системы ранее импортированные УЦ-сертификаты и, возможно, CRL файлы, необходимые для проверки сертификатов клиентов:

```
(config—service—proxy)# do clear proxy certd
```

34.15.4 Файлы-списки.

Для некоторых типов команды acl существует возможность задать аргументы команды acl в виде файла-списка. Это бывает полезно, если таких аргументов очень много, чтобы писать их в командой строке. Данный список необходимо сначала импортировать в службу под некоторым именем. А затем использовать именованный список в качестве аргумента опции -list команды acl.

Импорт файла как списка параметров для дальнейшего использования в подкоманде -list команды acl:

```
(config—service—proxy)# do service proxy import list FILE [NAME [normalize]]
```

При задании параметра NAME списку будет присвоено указанное имя, иначе имя списка будет равно имени файла. При задании параметра normalize будет произведен процесс экранирования спецсимволов содержимого списка. Например это может понадобиться для списка URL, которые необходимо искать как есть, т.к. параметр команды acl типа uri - регулярное выражение.

Экспорт файла-списка:

```
(config—service—proxy)# do service proxy export list FILE
```

Удаление файла-списка:

```
(config—service—proxy)# do clear proxy list
```

34.16 Мониторинг службы

Служба прокси Dionis-NX имеет мощную систему мониторинга своего состояния. Мониторинг службы осуществляется различными подкомандами команды show service proxy info.

Рассмотрим данные подкоманды.

34.16.1 Общая информация

Общая информация о работе службы:

```
# show service proxy info
```

Будучи вызвана без параметров, эквивалентна команде:

```
# show service proxy info general
```

Формат вывода данной команды:

- Squid Object Cache - версия службы;
- Start Time - время запуска службы;
- Current Time - текущее время;
- Connection information for squid - информация о соединениях и клиентах:
  - Number of clients accessing cache - число обслуженных клиентов; клиенты различаются по из IP-адресам;
  - Number of HTTP requests received - число полученных HTTP-запросов;
  - Request failure ratio - соотношение неуспешных запросов (ошибка в TCP соединении,ошибка DNS или аппаратная сетевая ошибка) к успешным;
  - Average HTTP requests per minute since start - среднее число HTTP запросов в минуту;
  - Select loop called - среднее число вызовов select()/poll() и среднее время между ними;
- Cache information for squid - информация о кэше:
  - Hits as % of all requests - процент кэш-попаданий во всех запросов за последние 5 и 60 минут;
  - Hits as % of bytes sent - процент трафика кэш-попаданий во всем трафике за последние 5 и 60 минут;
  - Memory hits as % of hit requests - процент кэш-попаданий в памяти (TCP_MEM_HIT) во всех кэш-попаданиях;
  - Disk hits as % of hit requests - процент кэш-попаданий на диске (TCP_HIT) во всех кэш-попаданиях;
  - Storage Swap size - размер (Кб) данных в кэше на диске;
  - Storage Swap capacity - процент заполнения данными кэша на диске;
  - Storage Mem size - размер (Кб) данных в кэше в памяти;
  - Storage Mem capacity - процент заполнения данными кэша в памяти;

- Median Service Times (seconds) - среднее время выполнения операций (в секундах):
  - HTTP Requests (All) - среднее время выполнения запроса;
  - Cache Misses - среднее время выполнения запроса при кэш-промахе;
  - Cache Hits - среднее время выполнения запроса при кэш-попадании (запросы с результатом TCP_HIT, TCP_MEM_HIT);
  - Near Hits - среднее время выполнения запроса при кэш-обновлении (запросы с результатом TCP_REFRESH_HIT);
  - Not-Modified Replies - среднее время выполнения запроса If-modified-since (запросы с результатом TCP_IMS_HIT);
  - DNS Lookups - среднее время выполнения DNS-разрешения (прямого и обратного);
- Resource usage for squid - статистика использования ЦПУ и памяти службой (значения времени - в секундах):
  - UP Time - время работы службы в секундах;
  - CPU Time - процессорное время,использованное службой;
  - CPU Usage - процент использования процессора службой; отношение CPU Time к UP Time;
  - CPU Usage, 5 minute avg - аналогично предыдущему, но за последние 5 минут;
  - CPU Usage, 60 minute avg - аналогично предыдущему, но за последние 60 минут;
  - Process Data Segment Size via sbrk() - размер сегмента данных службы (Кб);
  - Maximum Resident Size - максимальный размер памяти, доступный службе;
  - Page faults with physical i/o - число ошибок «Отсутствие страницы в памяти»;
- Memory usage for squid via mallinfo() - статистика использования памяти службой:
  - Total space in arena - общий объем памяти,выделенный службе;
  - Ordinary blocks,Small blocks и др. - прочая информация о памяти возвращаемая функцией mallinfo();
- Memory accounted for - информация по учету выделенной памяти (размер указан в Кб):
  - Total accounted - общий объем отслеживаемой памяти;
  - memPool accounted - общий объем отслеживаемой памяти пулов (стуктур фиксированного размера);
  - memPool unaccounted - общий объем неотслеживаемой памяти для пулов;
  - memPoolAlloc calls - число вызовов функции memPoolAlloc, выделяющей пулы;
  - memPoolFree calls - число вызовов функции memPoolAlloc, освобождающей пулы;
- File descriptor usage for squid - статистика использования файловых дескрипторов (ФД):
  - Maximum number of file descriptors - максимально возможное число ФД для службы;
  - Largest file desc currently in use - максимальное число ФД, задействованных на данный момент;
  - Number of file desc currently in use - число ФД, используемых в настоящее время;
  - Files queued for open - число файлов в очереди на открытие (ненулевое значение возможно только для типа кэша aufs);
  - Available number of file descriptors - доступное число ФД;
  - Reserved number of file descriptors - зарезервированное число ФД;
  - Store Disk files open - число открытых файлов в настоящий момент;

- Internal Data Structures - статистика мест хранения объектов:
  - StoreEntries - число закэшированных службой объектов; каждый объект потребляет около 100 байт памяти службы (пул StoreEntry; см. далее info mem);
  - StoreEntries with MemObjects - число объектов, закэшированных в памяти, и объектов, к которым обращаются в настоящий момент;
  - Hot Object Cache Items - число объектов, закэшированных в памяти;
  - on-disk objects - число объектов закэшированных на диске.

34.16.2 Активные соединения

Следующая команда показывает список активных соединений на данный момент времени:

```
# show service proxy info active—requests
```

Формат вывода команды:

- Connection - адрес памяти структуры соединения;
- FD - дексриптор сокета для TCP соединения, после которого следует объем принятых и переданных через него данных;
- FD desc - краткое описание сокета, обычно это URI;
- in - адрес памяти буфера приема, смещение следующих принятых данных и размер буфера приема;
- peer - удаленный сокет для TCP-соединения (для режима перерывания - это сокет клиента);
- me - локальный сокет для TCP-соединения удаленного сервера (для режима перерывания - это сокет клиента);
- nrequests - число запросов,полученных по данному соединению;
- defer - осуществляется ли отложенное чтение на сокете соединения;
- uri - запрашиваемый URI из запроса клиента;
- log_type - статус кэша по данному запросу - то, что появится в журнале доступа после завершения обработки запроса;
- out.offset - смещение по которому запрашиваются данные (имеет смысл во время скачивания больших файлов);
- out.size - размер данных в ответе;
- req_sz - размер HTTP-запроса клиента;
- entry - адрес памяти структуры хранения (StoreEntry) и ее хэш;
- start - сколько секунд назад инициировано соединение.

34.16.3 AUFS

Следующая команда выводит данные по асинхронной обработке запросов при типе кэша aufs:

```
# show service proxy info aufs
```

Формат вывода команды:

- open,close,read,write и т.д. - счетчики асинхронно выполненных файлов операций (открытие,закрытие,чтение,запись и т.д.);
- Threads status: число асинхронных файловых операций;
- queue: размер очереди запросов; при превышении 80 (5*16, где 16 - число асинхронных потоков) - будет выдано предупреждение о загруженности системы.

34.16.4 Аутентификация

Следующая команда показывает общую статистику по аутентификации:

```
# show service proxy info <auth—digest|auth—basic>
```

Имеет смысл только при включённой аутентификации пользователей.

34.16.5 Аутентификация по пользователям

Следующая команда показывает статистику по аутентификации пользователей:

```
# show service proxy info <users>
```

Имеет смысл только при включённой аутентификации пользователей.

34.16.6 Клиенты

Следующая команда показывает статистику по по клиентам службы:

```
# show service proxy info clients
```

Формат вывода команды:

- Address: IP-адрес клиента;
- Name: FQDN-имя клиента;
- Currently established connections: число открытых соединений клиента и службы;
- HTTP Requests: число HTTP запросов клиента;
- TCP_HIT,TCP_MISS,…: статистика результатов запросов.

34.16.7 Трафик и ресурсы

Следующая команда показывает статистику по трафику и ресурсам:

```
# show service proxy info counters
```

Формат вывода команды:

- sample_time - время последнего расчета счетчиков; расчет производится не реже, чем один раз в минуту;
- client_http.requests - число HTTP-запросов, полученных от клиентов;
- client_http.hits - число кэш-попаданий в ответ на HTTP-запросы клиентов; соответствует числу строк с типом результата TCP_HIT в журнале доступа;
- client_http.errors - число клиентских транзакций, приведших к ошибке;
- client_http.kbytes_in - объем HTTP-трафика в Кб, полученного от клиентов (HTTP-запросы);
- client_http.kbytes_out - объем HTTP-трафика в Кб, переданного клиентам (HTTP-ответы);
- client_http.hit_kbytes_out - объем HTTP-трафика в Кб, переданного клиентам (HTTP-ответы) в результате кэш-попаданий, включая ответы с кодом 304 (Not Modified);
- server.all.requests - число запросов, переданных на УС;
- server.all.errors - число запросов на УС, приведших к ошибке;
- server.all.kbytes_in - объем трафика в Кб, полученого с УС;
- server.all.kbytes_out - объем трафика в Кб, переданного на УС;
- server.http… - аналогично server.all, но только для HTTP-запросов;
- server.ftp… - аналогично server.all, но только для FTP-запросов;
- server.other… - аналогично server.all, но только для прочих запросов (Gopher, WAIS, SSL);
- page_faults - число ошибок «Обращение к отсутствующей странице»;
- select_loops - число раз вывзова seclect()/poll() в главном цикле ввода-вывода службы;
- cpu_time - накопленное время CPU в секундах;
- wall_time - время, прошедшее с последнего расчета счетчиков;
- swap.outs - количество объектов(файлов), записанных в дисковый кэш службы;
- swap.ins - количество объектов(файлов), считанных с дискового кэша;
- swap.files_cleaned - количество объектов(файлов), удаленных периодической процедурой очистки;
- aborted_requests - число отмененных запросов на УС, произошедших из-за отмены их клиентами.

34.16.8 Пулы задержки

Следующая команда показывает статистику по пулам задержки (в байтах):

```
# show service proxy info delay
```

Формат вывода команды:

- Aggregate/Network/Individual - тип пула (общий/сетевой/индивидуальный);

- Max - размер пула;
- Restore(Rate) - объем данных,добавляемых к пулу каждую секунду;
- Current - текущий объем пула; узлы опредлеляются последним значимым октетом.

Пример для agr: Current: 12345

Пример для agr-host24: Current: 1:1234 2:5678, где 1,2 - последний октет IP-адреса хоста.

Пример для agr-net24-host16: Current [Network 5]: 1:1234 2:5678, где 1,2 - последний октет IP-адреса хоста, 5 - предпоследний октет IP-адреса сети, например 10.0.5.0/24.

34.16.9 Открытые файлы

Следующая команда показывает статистику по файлам,открытым службой:

```
# show service proxy info filedescriptors
```

Формат вывода команды:

- File - дескриптор файла;
- Type - тип файла: File - для дискового кэша, лог-файла; Pipe - канал для IPC, межпроцсессного взаимодействия; Socket - соект для свзяи с клиентами,УС и IPC;
- Tout - таймаут для файлов типа Socket; такой файл закроется,если по нему не будет активности по истечении указанного таймаута;
- Nread - количество байт, прочитанных из файла;
- Nwrite - количество байт, записанных в файл;
- Remote Address - для файлов типа Socket это удаленный TCP-адрес соединения (удаленный сокет);
- Description - описание файла для типов Socket/Pipe или путь к файлу для типа File.

34.16.10 X-Forwarded-For-заголовки

Следующая команда показывает полученные X-Forwarded-For заголовки:

```
# show service proxy info forward—headers
```

X-Forwarded-For-заголовок содержит IP-адреса клиентов,для которых данный HTTP-запрос был ретранслирован прокси-сервером. Например,если служба получает HTTP-запрос от узла, в котором содержится X-Forwarded-For-заголовок с адресом 1.1.1.1, это значит, что служба получила данный запрос от другого прокси-сервера, который, в свою очередь, получил этот запрос от узла 1.1.1.1.

34.16.11 Via-заголовки

Следующая команда показывает полученные Via-заголовки:

```
# show service proxy info via—headers
```

Via-заголовок содержит имена и,возможно, порты и другую идентифицирующую информацию прокси-сереверов, через которые прошел полученный службой HTTP-запрос.

34.16.12 HTTP-заголовки

Следующая команда показывает статистику по HTTP-заголовкам:

```
# show service proxy info http—headers
```

Формат вывода команды:

- Header Stats: request - статистика по HTTP-запросам;
  - Field type distribution - распределение HTTP-заголовков:
    * id - внутренний идентификатор;
    * name - имя заголовка;
    * count - количество заголовков;
    * #/header - частота появления заголовков; например, частота 1.0 означает,что заголовок присутствует в каждом запросе;
  - Cache-control directives distribution - распределение директив кэширования заголовка Cache-Control:
    * id - внутренний идентификатор;
    * name - имя директивы;
    * count - количество директив;
    * #/cc_field - частота появления директив кэширования;
  - Number of fields per header distribution - распределение количества HTTP-заголовков:
    * id - внутренний идентификатор;
    * #flds - количество заголовков;
    * count - число запросов с указанным в поле #flds количеством заголовков;
    * %total - доля запросов с указанным в поле #flds количеством заголовков в общем числе запросов;
- Header Stats: reply - статистика по HTTP-ответам. Поля аналогичны полям для Header Stats: request;
- Http Fields Stats (replies and requests) - статистика по HTTP-запросам и HTTP-ответам:
  - id - внутренний идентификатор;
  - name - имя заголовка;

  - #alive - количество заголовков типа name, хранаящихся в данный момент в памяти (заголовки активных содеинений и для объектов,хранящихся в кэш-памяти службы;
  - %err - процент ошибочных заголовков типа name;
  - %repeat - доля запросов/ответов, с повторяющимися однотипными заголовками типа name;

- Headers Parsed - число обработанных HTTP-запросов/ответов;
- Hdr Fields Parsed - число обработанных HTTP-заголовков.

34.16.13 DNS-клиент

Следующая команда показывает статистику по внутреннему DNS-клиенту службы:

```
# show service proxy info idns
```

Формат вывода команды:

- The Queue - очередь неразрешенных DNS-запросов:
  - ID - внутренний идентификатор;
  - SIZE - размер запроса;
  - SENDS - число попыток запроса;
  - FIRST SEND / LAST SEND - промежуток времени между последним и первым запросом;
- Nameservers - статистика запросов/ответов на сервера имен:
  - IP ADDRESS - адрес сервера имен;
  - # QUERIES - число посланных на сервер имен запросов;
  - # REPLIES - число полученных от сервера имен ответов;
- Rcode Matrix - статистика DNS-ответов:
  - RCODE - код ответа:
    * 0 - успешный ответ;
    * 1 - сервер имен не смог понять запрос (Format Error);
    * 2 - проблема с сервером имен (Server Failure);
    * 3 - доменное имя не существует (Name Error);
    * 4 - сервер имен не поддерживает указанный тип запроса (Not Implemented);
    * 5 - отказ в обработке запроса из-за политики безопасности сервера (Refused);
  - ATTEMPT1 - число одинарных повторных попыток запроса в результате получения RCODE=2;
  - ATTEMPT2 - число двойных повторных попыток запроса в результате получения RCODE=2;
  - ATTEMPT3 - число тройных повторных попыток запроса в результате получения RCODE=2;
- Search list - список доменов,используемых при разрешении имен.

34.16.14 DNS-кэш

Следующая команда показывает статистику по внутреннему DNS-кэшу службы:

```
# show service proxy info ipcache
```

Формат вывода команды:

- IP Cache Statistics - статистика по DNS-кэшу:
  - IPcache Entries In Use - количество записей кэша, используемых в настоящее время;
  - IPcache Entries Cached - количество записей кэша;
  - IPcache Requests - число DNS-запросов;
  - IPcache Hits - число DNS-запросов, разрешенных из кэша DNS-службы;
  - IPcache Negative Hits - число DNS-запросов, негативно разрешенных из кэша DNS-службы;
  - IPcache Numeric Hits - число запросов разрешения адреса в имя, разрешенных из кэша DNS-службы;
  - IPcache Misses - число DNS-запросов, разрешенных через DNS-сервер,а не из кэша DNS-службы;
  - IPcache Retrieved A - число полученных A-записей;
  - IPcache Retrieved AAAA - число полученных AAAA-записей;
  - IPcache Retrieved CNAME - число полученных CNAME-записей;
  - IPcache CNAME-Only Response - число полученных только CNAME-записей;
  - IPcache Invalid Request - число неверных DNS-запросов;
- IP Cache Contents - кэш DNS-службы для наиболее популярных имен:
  - Hostname - доменное имя;
  - Flg - флаги: N - кэширование негативно разрешенного имени; H - разрешение пришло из статического назначения системы (см. ip resolver hosts);
  - lstref - показывает, сколько секунд назад запись использовалась последний раз;
  - TTL - время жизни записи в кэше (секунд);
  - N(b) - N: число IP-адресов имени (адреса с суффиксом OK); b: число IP-адресов имени, которые недоступны в настоящее время (адреса с суффиксом BAD);
  - последняя колонка - показывает IP-адреса с суффиксом OK или BAD (см. N(b) колонку).

34.16.15 Память

Следующая команда показывает статистику использования памяти службой:

```
# show service proxy info mem
```

Формат вывода команды:

- Largest pools stats - статистика по двум максимальным пулам фиксированных структур:

  - Pool name - имя пула: StoreEntry - пул структур,создаваемая на каждый кэшированный объект; MD5-digest - пул хэшей ответа; All-pools - все пулы;
  - Size - размер экзампляра стуктуры (байт);
  - Number - число структур в пуле;
  - TotSize - общий размер пула;
  - HiSize - максимально наблюдаемый размер пула;

- Cumulative allocated volume - общий объем памяти, выделенной службе; часть ее может быть освобождена; учитывается только выделяемая память;
- Total Pools created - общее число созданных пулов;
- Pools ever used - число использованных по настоящий момент пулов;
- Currently in use - число используемых в настоящий момент пулов.

34.16.16 Свежесть

Следующая команда показывает статистику алгоритма проверки на свежесть:

```
# show service proxy info refresh
```

Формат вывода команды:

- HTTP histogram - показывает распределение проверок на свежесть, приведших к решению о свежести объекта (Fresh/Stale - свежий/несвежий), при запросе его клиентом:
  - Count - общее число проверок данного типа;
  - %Total - доля проверок данного типа;
  - Category - тип проверки на свежесть: —причины свежести—
    * Fresh: request max-stale wildcard - в запросе была директива max-stale, т.е. клиент хочет принять объект любой свежести;
    * Fresh: request max-stale value -в запросе была директива max-stale со значением больше времени,прошедшего с момента истечения времени жизни объекта (Expires);
    * Fresh: expires time not reached - момент истечения времени жизни объекта еще не наступил;
    * Fresh: refresh_pattern last-mod factor percentage - объект подпадает под refresh-правило; LM-фактор объекта меньше указанного в правиле LM-фактора;
    * Fresh: refresh_pattern min value - возраст объекта меньше указанного в значении min правила refresh, в которое попадает объект;
    * Fresh: refresh_pattern override expires - объект подпадает под refresh-правило с параметром override-expire;
    * Fresh: refresh_pattern override lastmod - объект подпадает под refresh-правило с параметром override-lastmod; —причины несвежести—
    * Stale: response has must-revalidate - запрос содержит директиву кэширования Cache-Control: must-revalidate;
    * Stale: changed reload into IMS - объект подпадает под refresh правило с параметром reload-into-ims;

    * Stale: request has no-cache directive - запрос содержит директиву кэширования Cache-Control: no-cache;
    * Stale: age exceeds request max-age value - в запросе была директива max-age со значением меньше возраста объекта;
    * Stale: expires time reached - момент истечения времени жизни объекта наступил;
    * Stale: refresh_pattern max age rule - возраст объекта больше указанного в значении max правила refresh, в которое попадает объект;
    * Stale: refresh_pattern last-mod factor percentage - объект подпадает под refresh-правило; LM-фактор объекта не меньше указанного в правиле LM-фактора;
    * Stale: by default - объект не подпадает ни под один критерий алгоритма проверки, в связи с чем признается несвежим по умолчанию;
    * TOTAL - общее число проверок;

- On Store histogram - аналогично HTTP histogram, но для проверок на свежесть ответов УС для кэш-промахов.

34.16.17 Ретрансляция запросов

Следующая команда показывает статистику ретрансляции запросов УС:

```
# show service proxy info requests
```

Формат вывода команды:

- Status - код HTTP-ответа:
    - 1xx: Informational (информационные);
    - 2xx: Success (успешно);
    - 3xx: Redirection (перенаправление);
    - 4xx: Client Error (ошибка клиента);
    - 5xx: Server Error (ошибка сервера);
- try#1-10 - число попыток, предпринятых для получения ответа типа Status (try#1 - одна попытка,…,try#10 - десять попыток).

34.16.18 Кэш

Следующая команда показывает статистику по кэшу:

```
# show service proxy info storage
```

Формат вывода команды:

- Store Entries - число кэшированных объектов;
- Store Directory #0 - указывает тип кэша (aufs);

- FS Block Size - размер блока файловой системы (байт);
- First level subdirectories - число директорий первого уровня;
- Second level subdirectories - число директорий второго уровня;
- Maximum Size - максимальный размер кэша;
- Current Size - текущий размер кэша;
- Percent Used - текущая заполненность кэша;
- Filemap bits in use - сколько битов файловой карты использовано;
- Filesystem Space in use - объем места на диске, использованного службой;
- Filesystem Inodes in use - сколько использовано инодов;
- Flags: SELECTED - всегда значение SELECTED; значит что кэш в режиме чтения-запись; формат read-only для кэша не поддерживается;
- Removal policy - типа политики замены объектов в дисковом кэше: lru - для LRU, heap - для LFUDA, GDSF или HLRU;
- LRU reference age - показывает дату самого старого объекта кэша; только для политики замены объектов LRU.

34.16.19 Рекомендации по настройке

34.16.19.1 Размер дискового кэша

Размер дискового кэша определяется опытным путем в зависимости от нужд клиентов службы, их числа, а также размера доступной оперативной памяти.

Например при использовании 4Гб дискового кэша объем потребляемой памяти при полном заполнении кэша будет варьироваться в пределах 150-200Мб.

34.16.20 Примеры

34.16.20.1 Проксирование HTTPS с подменой сертификата и расшифровкой трафика

Этот режим полезен,если мы хотим осуществлять проксирование HTTPS посредством подмены сертификатов. В этом случае служба соединяется с удаленным HTTPS-сервером используя автоматически созданный сертификат, на основе информации полученной на шаге 1 (peek), и затем соединяется с клиентом используя импортированный в браузер сертификат из службы. Таким образом мы имеем полную расшифровку службой HTTPS-трафика клиента.

```
(config—service—proxy)# listen 10.0.0.1 3128 ssl—bump gencert
(config—service—proxy)# ssl not—verify—peer
(config—service—proxy)# acl step1 ssl—bump step1
(config—service—proxy)# ssl—bump peek step1
(config—service—proxy)# ssl—bump bump all
(config—service—proxy)# ssl—cert—error permit all
```

После этого необходимо экспортировать сертификат из службы в DER формате и передать его клиентам, чтобы они импортировали его в свой Интернет-бразуер:

```
(config—service—proxy)# do service proxy export FILE_PATH der
```

34.16.20.2 Проксирование HTTPS без подмены сертификата и без расшифровки трафика.

Этот режим полезен,если мы хотим осуществлять прозрачное проксирование и ,например, запрет HTTPS запросов на определенные домены: для этого вовсе необязательно расшифровывать трафик,т.к. имя домена сообщается уже во время установления HTTPS-сессии.

```
(config—service—proxy)# listen—https 10.0.0.1 3129
(config—service—proxy)# ssl not—verify—peer
(config—service—proxy)# acl step1 ssl—bump step1
(config—service—proxy)# acl bad ssl—srv site.com
(config—service—proxy)# ssl—bump peek step1
(config—service—proxy)# ssl—bump terminate bad
(config—service—proxy)# ssl—bump splice all
(config—service—proxy)# ssl—cert—error permit all
```

35. Служба WCF

Dionis-NX имеет службу WCF, которая используется для фильтрации HTTP/HTTPS трафика.

Основные возможности службы:

- фильтрация содержимого WEB-страниц по ключевым словам
- возможность проверки файлов, получаемых из WEB, антивирусом
- поддержка режимов аутентификации пользователей: IP-, digest-, basic-аутентификация
- белые, черные и серые списки для фильтрации WEB-трафика
- регулярные выражения для фильтрации WEB-трафика
- регулярные выражения для подмены WEB-трафика
- глубокий поиск URL: детектирование вложенных URL
- ограничения размеров посылки данных методом POST
- фильтрация WEB-трафика на основе меток времени

35.1 Быстрая настройка.

Для быстрой настройки введите, например, следующие команды:

```
(config—service—wcf)# listen ip 192.168.0.1 8080
(config—service—wcf)# auto—list all
```

Данные команды установят режим аутентификации по IP-адресу, а также включат списки контроля доступа и фильтрации по-умолчанию.

Далее рассмотрим основные понятия службы.

35.2 Аутентификация клиентов службы

Можно задавать различные настройки службы для разных клиентов.

Клиенты могут дифференцироваться по имени пользователя или по IP-адресу.

В 1м случае необходима работа службы в связке с proxy-сервером, который проводит непосредственно аутентификацию пользователей, и далее направляет WEB-трафик на службу wcf, в том числе заголовок Proxy-Authorization, из которого служба wcf узнает имя пользователя.

В 2м случае proxy-сервер не нужен и клиенты будут определяться по их IP-адресу.

Режим аутентификации жестко связан с сокетом, на котором служба принимает запросы.

Для задания режима аутентификации:

```
(config—service—wcf)# listen <basic|digest|ip> IP PORT
```

Если задано несколько режимов аутентификации, необходимо иметь разные пары IP PORT для них. Кроме того, basic не может существовать параллельно с digest или ip, поэтому в случае задания нескольких режимов basic-режим не будет использоваться.

35.2.1 IP-аутентификация

Для работы IP-аутентификации достаточно настроить только службу wcf. Рассмотрим пример:

```
adm@DionisNX(config—service—wcf)# do show
listen ip 10.0.0.1 8080
1 group def
 sitelist  banned
  add mail.ru
2 group novk
 match ip 10.0.0.2
 sitelist  banned
  add vk.com
enable
```

В данном случае для клиента с адресом 10.0.0.2 будет выбраны настройки группы novk, в которой запрещен доступ к домену vk.com. Для все остальных клиентов запрещен доступ к домену mail.ru.

35.2.2 Proxy-аутентификация

Для работы Proxy-аутентификации необходимо настроить как службу wcf, так и службу proxy. Рассмотрим пример:

```
adm@DionisNX(config—service—wcf)# do show
listen basic 127.0.0.1 8080
1 group def
 sitelist  banned
  add mail.ru
2 group novk
 match user oleg
 sitelist  banned
  add vk.com
enable

adm@DionisNX(config—service—proxy)# do show
listen 10.0.0.1 8081
```

```
1 acl auth proxy—auth—all
cache peer parent 127.0.0.1 8080
1 http—access permit auth
2 http—access deny all
1 auth local 3
user bob 555
user oleg 123
enable
```

В данном случае аутентификация производится в службе proxy: настроена локальная аутентификация, но возможно настроить и любую другую. Служба proxy после аутентификации перенаправляет запрос далее в службу wcf при помощи команды **cache peer**, однако возможно настроить связь двух служб и по-другому, например, посредством icap. Далее в службе wcf для пользователя oleg предусмотрена группа novk. Для остальных группа def. Настройки списков фильтрации в группах аналогичны Примеру 1.

35.3 Группы пользователей.

Группы пользователей позволяют задать для определенных пользователей свои настройки службы. Глобальные настройки службы можно задать только в корне настроек службы. Существует также одна команда, которую возможно задать как глобально, так и на уровне группы: **weighted-phrase-mode**, которая включает режим фильтрации контента, или же вообще отключает фильтрацию контента.

Группы имеют имя и приоритет. В группу под номером 1 попадают пользователи, которые не попали в остальные группы. Для читабельности конфигурации рекомендуется называть эту группу именем default, но это не обязательно.

Для создания группы введите:

```
(config—service—wcf)# group default
(config—service—wcf)# group vipusers
```

Для группы с номерами больше 1 необходимо задать правило попадания в группу, например:

```
(config—service—wcf—group—default)# match ip 10.0.0.1
(config—service—wcf—group—default)# match user Oleg
```

Правила отбора пользователей должны быть заданы с учетом включенных режимов аутентификации (см. Аутентификация клиентов службы).

Например, если задана команда **match ip**, то должна быть и команда **listen ip**; если задана команда **match user**, то должна быть и команда **listen digest** или **listen basic**.

35.4 Списки фильтрации

Служба осуществляет фильтрацию WEB-трафика посредством списков фильтрации.

35.4.1 Понятие списка

Список представляет собой набор элементов (далее - объектов).

35.4.2 Понятие фильтрации

Под фильтрацией могут пониматься две операции:

- **фильтрация** - при обнаружении объекта в данных WEB-запроса осуществляются либо запрет, либо разрешения прохождения запроса дальше
- **модификация** - при обнаружении объекта в данных WEB-запроса осуществляется модификация данных WEB-запроса

Теперь, зная понятия список и фильтрация, можно определить список фильтрации по **содержанию** и по **именованию**.

35.4.3 Содержание списка фильтрации

Под содержанием списка будем понимать все данные, которые содержит список. Список фильтрации с точки зрения содержания представляет собой набор объектов. Объекты могут быть двух типов:

- **обычный объект** - например, IP-адрес, интервал IP-адресов, домен, URL и др.
- **правило** - например RX1 –> RX2, где RX1/2 - регулярные выражения

35.4.4 Именование списка фильтрации

Список фильтрации с точки зрения именования определяется двумя параметрами: **тип** списка и **имя** списка. Таким образом формат задания списка через команды системы такой: ТИП_СПИСКА ИМЯ_СПИСКА. Тип списка влияет на содержание списка, однозначно задавая типы объектов в списке - обычные объекты или правила (см.Содержание списка фильтрации.).

Например тип списка mimelist или iplist всегда задает только списки обычных объектов, а тип списка regexpreplacelist задает списки правил.

Рассмотрим далее типы списков фильтрации.

35.4.4.1 Тип списка

Тип списка определяется типом объектов, например список IP-адресов, список доменов, список MIME-типов и др.

Рассмотрим типы списков:

- **fileextlist** - список расширений имен файлов
- **iplist** - список IP-адресов клиентов службы (источников запроса)
- **ipsitelist** - список IP-адресов (и их интервалов) сайтов
- **mimelist** - список MIME-типов
- **regexpboollist** - список регулярных выражений для поиска в WEB-запросе (в URL, HTTP-заголовках и др.)
- **regexpreplacelist** - список регулярных выражений для модификации WEB-запроса (URL, HTTP-заголовках и др.)
- **searchlist** - список слов, отслеживаемых в поисковых запросах
- **sitelist** - список доменов сайтов
- **urllist** - список URL сайтов
- **phraselist** - список фраз

35.4.4.2 Имя списка

Имя списка может быть двух типов:

- **стандартное имя**: список с данным именем будет использован без дополнительных действий, т.к. это имя используется в стандартном системном файле истории;

- **пользовательское имя**: список с данным именем не будет использован до тех пор, пока это имя не будет указано в пользовательском файле истории (см. Файл истории)

Файл истории описывает логику прохождения WEB-запроса по спискам фильтрации. В службе используются предопределенные стандартные файлы истории. Если используются стандартные файлы истории, то и имена списков необходимо использовать тоже стандартные, которые видны по автодополнению команды списка.

Далее мы будем рассматривать только стандартные имена списков, т.к. только для них существует уже описанная в службе логика в стандартном файле истории.

35.4.5 Имена стандартных списков фильтрации

Имена для стандартных списков часто имеют похожий формат: имя списка начинается с одного из трех ключевых слов, которые показывают что именно делать с объектами (IP-адреса, домены и т.д), которые описаны в данном списке. Рассмотрим эти ключевые слова:

- **banned** - черный список объектов, объекты из данного списка запрещены при обработке WEB-запроса;
- **exception** - белый список объектов (IP-адресов, доменов и т.д), объекты из данного списка разрешены при обработке WEB-запроса;
- **grey** - серый список объектов (IP-адресов, доменов и т.д), объекты данного типа разрешены, но WEB-запрос будет проверен далее по спискам фильтрации содержимого
- **authexception** - список объектов до этапа аутентификации
- **bannedssl** - черный список объектов (для HTTPS сайтов)
- **log** - список объектов, которые нужно только протоколировать
- **refererexception** - сайты, на которые ссылаются указанные объекты, будут в белом списке
- **exceptionfile** - белый список объектов, для которых возможно скачивание файлов
- **exceptionvirus** - белый список объектов, которые не нужно проверять на вирусы
- **nocheckcert** - список объектов, для которых не проверять SSL сертификат
- **embededreferer** - список объектов, на которые можно ссылаться
- **addheader/headermods/change/redirect/searchterms/sslreplace** - относятся к типу списка regexpreplacelist и связаны с модификацией элементов WEB-запроса: заголовка, URL, перенаправления, искомых слов
- **weighted** - относится к типу списка phraselist и представляют собой фразы с весом, что более подробно рассмотрено далее (см. Списки фраз).

Неописанные выше имена строятся похожим образом, например: exceptionheader, banneduseragent, и т.д.

35.4.6 Область действия списка

Списки можно определять как глобально в корне службы, так и на уровне групп пользователей.

35.4.6.1 Глобальные списки фильтрации

Данные списки применяются на этапе **до аутентификации** пользователя в службе, т.е. до момента отнесения пользователя к той или иной группе.

Эти списки задаются в корне конфигурации службы.

35.4.6.2 Групповые списки фильтрации

Данные списки применяются после этапа аутентификации пользователя в службе, т.е. после момента отнесения пользователя к той или иной группе.

Основная масса списков определяется именно на уровне группы.

35.5 Списки фраз

Из всех типов списков рассмотрим один, т.к. он требует пояснения. Это тип phraselist - списки правил для фраз.

35.5.1 Белый и черный список правил для фраз

Данные списки правил определяются следующими командами:

- phraselist banned - черный список правил
- phraselist exception - белый список правил

Для добавления нового правила в эти списки используйте команду:

```
add <P1> [P2] ... [P8]
```

Здесь Pn - это фраза, т.е. слово или строка, которая может быть заключена в кавычки.

Формат фразы:

- “PHRASE” : без пробелов в начале и в конце; совпадение по всем фразам, которые содержат на указанную фразу в любой своей части
- ”PHRASE”: с пробелом в начале и без пробела в конце; совпадение по всем фразам, которые начинаются на указанную фразу
- “PHRASE” : с пробелом в конце и без пробела в начале; совпадение по всем фразам, которые кончаются на указанную фразу
- ”PHRASE “: с пробелом в конце и в начале; совпадение по всем фразам, которые равны на указанной фразе (точное совпадение)

Если в команде add указано несколько разных фраз, то правило сработает, если будут найдены все указанные фразы на данной WEB-странице, т.е. используется AND-логика.

Удаление правила возможно по его номеру.

35.5.2 Список правил для фраз с весами

Данные списки правил определяются следующей командой:

- phraselist weighted

Для добавления нового правила в эти списки используйте команду:

```
add <W> <P1> [P2] ... [P8]
```

Здесь W - это вес правила, который может быть задан как положительное или отрицательной число или процент. При срабатывании правил для WEB-страинцы, данные числа суммируются и результат сравнивается со значением **limit naughtyness** (предел "грязности" страницы). Если результат суммирования весов сработавших правил превысит значение **limit naughtyness** (по-умолчанию, 100), то страница будет заблокирована.

В остальном формат правила аналогичен правилу для черного и белого списков.

Удаление правила возможно по его номеру.

35.6 Файл истории

Файл истории определяет логику и процесс прохождения WEB-запроса по спискам. В нем определены функции, описывающие поведение списков фильтрации. Имена функций соответствуют именам списков фильтрации. Файл истории написан на упрощенном языке.

Файлы истории существуют глобально для службы и для каждой группы.

35.6.1 Файлы истории для службы

В начале процесса прохождения WEB-запроса используется доаутентификационный файл истории. В нем описаны, например, функции для таких имен списков фильтрации, как: authexception, exceptionclient и др.

Чтобы изменить доаутентификационный глобальный файл истории:

story-pre-auth <FILE>

По-умолчанию: ro:/wcf/story/preauth.story

35.6.2 Файлы истории для группы

Файл истории для группы состоит из трех частей, т.е. других файлов истории, которые применяются последовательно:

- **базовый файл истории** - определяет общую базовую логику прохождения WEB-запроса и все функции стандартных списков фильтрации;
- **файл истории сайта** - это общий для всех групп файл истории, который включается в каждую группу; в нем можно переопределить стандартные функции из системного файла истории, или добавить новые функции для нестандартных списков фильтрации;
- **файл истории группы** - это такой же файл истории, как и предыдущий, только используется он для данной конкретной группы;

В файле истории сайта можно переопределить функции базового файла истории, а в файле истории группы можно переопределить функции файла истории сайта.

Чтобы изменить файл истории сайта (общегрупповой файл истории), выполните на уровне конфигурации службы:

story-site <FILE>

По-умолчанию: ro:/wcf/story/site.story

Чтобы изменить файл истории группы, выполните на уровне конфигурации группы:

story <FILE>

По-умолчанию: ro:/wcf/story/group.story

Для удобства пользователя стандартные файлы истории (доаутентификационный, общегрупповой и групповой) скопированы в пространство wcf:/story. Администратор может их заменить или отредактировать.

35.7 Настройка работы по ICAP

Существует возможность передавать в службу для анализа WEB-трафик, в том числе расшифрованный HTTPS-трафик, от службы proxy.

Для примера рассмотрим службу proxy и wcf на одной и той же системе Dionis-NX.

В конфигурации proxy необходимо указать службу wcf как ICAP-сервер, добавив следующие команды:

```
adapt masterx—shared—names X—ICAP—E2G
adapt send—client—ip
adapt send—username X—Client—Username
adapt—srv req icap request 127.0.0.1:1344/request bypass
adapt—srv resp icap response 127.0.0.1:1344/response bypass
```

В настройки службы wcf необходимо добавить ICAP-порт:

```
listen icap 1344
```

Браузеры клиентов в данной схеме будут работать через службу proxy.

35.8 Настройка работы с HTTPS в режиме MITM

В режиме HTTPS MITM непосредственно служба wcf будет заниматься расшифрованием HTTPS-трафика.

Если не настроить службу в данном режиме, то для HTTPS-сайтов будет невозможна фильтрация по содержимому и любым другим данным, которые являются зашифрованными для HTTPS-сайтов. Однако останется вторая возможность направлять расшифрованный службой proxy трафик по ICAP протоколу в службу wcf для анализа (см. Настройка работы по ICAP).

Для настройки работы службы в режиме HTTPS MITM необходимо создать приватный ключ и сертификат:

```
ssl pkey generate rsa bits 4096 cert:/private_cert.pem
ssl cert generate cert:/private_cert.pem 1000 cert:/my_rootCA.crt
ssl cert convert cert:/my_rootCA.crt der cert:/my_rootCA.der
ssl pkey generate rsa bits 4096 cert:/private_root.pem
```

Этими командами мы создаем корневой сертификат для службы wcf, его der-вариант для импорта в браузеры клиентов, а также приватные ключи.

Далее необходимо прописать созданные сертификаты и ключи в службе wcf и включить в ней режим ssl:

```
ssl enable
ssl path privkey—cert cert:/private_cert.pem
ssl path root—ca—cert cert:/my_rootCA.crt
ssl path root—ca—key cert:/private_root.pem
```

Далее необходимо установить файл my_rootCA.der в браузеры клиентов - например разместить где-либо в LAN-домене ссылку для скачивания на него.

После этого включим режим MITM в группе:

```
ssl man—in—middle
```

Этими командами мы включаем режим MITM для группы.

Если возникнут проблемы с проверкой сертификата, можно указать команду не проверять сертификат:

```
ssl no—check—cert
```

35.9 Настройка работы с HTTP/HTTPS в прозрачном режиме

В режиме прозрачного проксирования HTTP/HTTPS служба wcf не будет заниматься расшифровыванием HTTPS-трафика, в результате будет невозможна фильтрация по содержимому и любым другим данным, которые являются зашифрованными для HTTPS-сайтов. Однако преимущество данного режима состоит в том, что:

- останется возможность блокирования по именам доменов HTTPS;
- нет необходимости импортировать сертификаты в браузеры клиентов, а также как-либо иначе настраивать браузеры клиентов для работы с службой wcf.

В следующих примерах IP-адрес 192.168.33.214 является адресом шлюза по-умолчанию и адресом принятия запросов от клиентов службой wcf.

В службе wcf укажем порт, на котором слушать HTTPS-трафик, например:

```
listen ip 192.168.33.214 8080
listen https 8443
```

Далее необходимо настроить NAT-правила для перенаправления HTTP/HTTPS трафика в службу:

```
adm@DionisNX(config—nat—proxy)# do show
1 exclude in tcp dport 443 dst 192.168.33.214
2 exclude in tcp dport 80 dst 192.168.33.214
3 nat tcp dport 443 src 192.168.33.0/24 redirect port 8443
4 nat tcp dport 80 src 192.168.33.0/24 redirect port 8080
```

Также желательно ограничить запросы на порты службы:

```
(config)# ip access—list dropmyself
(config—acl—dropmyself)# deny tcp dport 8080 dst 192.168.1.254
(config—acl—dropmyself)# deny tcp dport 8443 dst 192.168.1.254
```

Далее необходимо применить созданные nat- и acl-правила:

```
adm@DionisNX(config—if—ethernet0)# do show
ip address 192.168.33.214/24
ip nat—group proxy
ip access—group dropmyself in
enable
```

В браузерах клиентов необходимо отключить использование proxy-сервера.

35.10 Связь со службой proxy

На службу wcf можно перенаправлять WEB-запрос клиента от службы proxy.

Для этого настройте в службе wcf, например basic-аутентификацию с приемом запросов на loopback-интерфейсе:

```
listen basic 127.0.0.1
```

Вместо basic-аутентификации можно использовать digest-аутентификацию.

Далее настройте в службе proxy:

```
cache peer parent 127.0.0.1 8080 login=*:password
```

В случае использования вместе с proxy-службой, не рекомендуется запускать wcf-службу на адресе, отличном от loopback-адреса (127.0.0.1), поскольку в таком случае возможно возникновение сетевых петель.

35.11 Информация о службе

Для просмотра логов службы введите:

show service wcf log [access|debug|stat]

Покажет лог доступа, отладочный лог, лог статистики. Если не указать тип лога - покажет лог службы.

Для включения отладочного лога:

log debug <all|icap|net|filter>

Для включения лога статистики:

log stat [INTERVAL]

Для включения лога доступа:

log access <denied|all|text>

Для включения лога оригинальных, незимененных службой, запросов клиента:

log request

Остальные подкоманды команды log также влияют на информацию в логе доступа. Например: log user-agent, log add-blocks и др.

36. Служба SNMP

Dionis-NX имеет службу SNMP.

Данная служба позволяет другим узлам получить SNMP-информацию о системе, а также отсылает другим узлам SNMP-нотификации о старте и остановке службы.

36.1 Общая настройка службы SNMP

Чтобы войти в режим настройки службы, следует выполнить команду:

```
(config)# service snmp
```

Настройка интерфейса и/или порта, который служба будет использовать для приема запросов и посылки ответов, нотификаций,zz например:

```
(config—service—snmp)# listen 192.168.0.1 udp
```

По умолчанию,используется UDP-порт 161 и любой локальный интерфейс, которому назначен IP-адрес.

Для работы с адресами IPv6 используется соответствующая команда **listen6**.

Служба snmp независимо от настроек производит опрос всех интерфейсов системы. Когда в системе большое количество интерфейсов, то это может существенно повлиять на загрузку ЦПУ. Если нет необходимости в сборе статистики о каких-то контретных типах интерфейсов, то их можно исключить из работы службы snmp:

```
(config—service—snmp)# excluded vpn
```

36.2 Настройка базовой SNMP информации

Настройка общей информации о системе. Например:

```
(config—service—snmp)# sysinfo location russia
(config—service—snmp)# sysinfo name router1
(config—service—snmp)# sysinfo contact admin@domain
```

Этими командами можно задать, соответственно, физическое местонахождение системы, имя системы и адрес электронной почты администратора системы.

36.3 Настройка SNMPv3

SNMPv3 поддерживает несколько моделей безопасности: User-based Security Model (USM) с использованием логина и пароля пользователя, и Transport Security Model (TSM) с использованием сертификатов пользователя.

Настройки SNMPv3 выделены в отдельный блок команд для перехода к которым необходимо в режиме настройки службы SNMP выполнить команду:

```
(config—service—snmp)# snmpv3
```

36.3.1 SNMPv3 USM

Данная модель безопасности улучшает формат сообщения SNMP, добавляя надлежащую проверку целостности и шифрование, таким образом позволяя передавать сообщения по незащищенным каналам. USM предполагает наличие пользователя SNMPv3 и предусматривает 3 уровня безопасности:

- noAuthNoPriv – данные передаются в открытом виде, конфиденциальность данных отсутствует;
- authNoPriv – аутентификация без шифрования;
- authPriv – аутентификация и шифрование. Максимальный уровень защищенности.

Ниже приведены команды для настройки различных пользователей SNMPv3. Например:

```
(config—service—snmp—v3)# user user1
(config—service—snmp—v3)# user user2 md5 BigAuthPass
(config—service—snmp—v3)# user user3 sha BigAuthPass des BigPrivPass
```

Рассмотрим приведенные в примере команды:

- первая команда: первый параметр команды (user1) - это обязательный параметр, задающий имя пользователя, которое будет использоваться в дальнейшем для настроек правил нотификаций и правил доступа. Соответствует уровню безопасности *noAuthNoPriv*;
- вторая команда: для пользователя user2 будет использоваться аутентификация по алгоритму md5 с использованием пароля BigAuthPass. Соответствует уровню безопасности *authNoPriv*;
- третья команда: для пользователя user3 будет использоваться аутентификация по алгоритму sha с использованием пароля BigAuthPass и шифрование по алгоритму des с использованием пароля BigPrivPass. Соответствует уровню безопасности *authPriv*.

36.3.2 SNMPv3 TSM

Данная модель безопасности позволяет передавать сообщения поверх защищенного TLS-соединения. TSM предполагает наличие пользователя SNMPv3, а также сертификатов и ключей в PEM формате.

Ниже приведены команды для настройки TSM модели SNMPv3:

```
(config—service—snmp—v3)# tsm ca cert:/root.crt
(config—service—snmp—v3)# tsm key cert:/agent.key
(config—service—snmp—v3)# tsm cert cert:/agent.crt
(config—service—snmp—v3)# user user4 tsm cert:/manager.crt sha256
```

Рассмотрим приведенные в примере команды:

- первая команда: задает корневой сертификат TSM (необязательная команда);
- вторая команда: задает секретный ключ агента;
- третья команда: задает сертификат агента;
- четвертая команда: задает имя и сертификат пользователя для модели TSM, которые будут использоваться в дальнейшем для настроек правил нотификаций и правил доступа. Также задается алгоритм расчета отпечатка сертификата.

36.3.3 EngineID для SNMPv3 пользователей

На этапе создания SNMPv3 пользователей можно для каждого из них дополнительно задать значение engine-id. Например:

```
(config—service—snmp—v3)# user user2 engine—id 0x0102030405 md5 BigAuthPass
```

Данное значение используется только для отправки SNMPv3 нотификаций (см. далее). Если параметр не задан, то будет использовано значение 0x0000000000.

36.4 Настройка правил доступа

Служба SNMP в DionisNX поддерживает как классическую модель доступа, так и модель основанную на видах - VACM (View Access Control Model).

36.4.1 Настройка классической модели

Настройка правил, по которым другим узлам разрешено получать информацию о системе по SNMP-протоколу. Например:

```
(config—service—snmp)# acl pas1
(config—service—snmp)# acl pas2 1.2.3.4
(config—service—snmp)# acl pas3 2.2.2.0/24
```

Рассмотрим приведенные в примере команды:

- первая команда: первый параметр команды (pas1) - это обязательный параметр, задающий пароль доступа, который должен использоваться узлом, желающим получить информацию о системе по протоколу SNMP. Адрес узла может быть любой, т.к. он не указан.
- вторая команда: только узел с адресом 1.2.3.4 и паролем доступа pas2 может получить информацию о системе по протоколу SNMP.
- третья команда: только узлы сети 2.2.2.0/24 и паролем доступа pas3 могут получить информацию о системе по протоколу SNMP.

Настройки правил доступа про протоколу SNMPv3 осуществляется в секции snmpv3. При этом предварительно должна быть настроена USM и/или TSM модель для заданного пользователя:

```
(config—service—snmp)# snmpv3
(config—service—snmp—v3)# acl user1
```

В данном примере пользователю user1 разрешено получать информацию о системе по протоколу SNMPv3.

Для работы с адресами IPv6 используется соответствующая команда **acl6**.

36.4.2 Настройка VACM

Для того, чтобы задействовать VACM-модель нужно в настройках классической модели доступа указать параметр **vacm** и задать имя модели. Для протокола SNMPv3 имя модели задавать не нужно (оно будет соответствовать имени пользователя). Напимер:

```
(config—service—snmp)# acl pas3 2.2.2.0/24 vacm myname
(config—service—snmp)# snmpv3
(config—service—snmp—v3)# acl user1 vacm
```

Рассмотрим приведенные в примере команды:

- первая команда: узлы сети 2.2.2.0/24 и паролем доступа pas3 могут получить информацию о системе по протоколу SNMP в соответствии с настроенной vacm моделью с именем myname;
- вторая команда: переход в режим конфигурации snmpv3;
- третья команда: пользователь user1 сможет получить информацию о системе в соответствии с настроенной vacm моделью.

Далее необходимо создать группу в которой описаны протоколы, по которым будет разрешено получать доступ:

```
(config—service—snmp)# vacm group my_group
(config—service—snmp—vacm—group—my_group) usm user1
(config—service—snmp—vacm—group—my_group) v2c myname
```

Рассмотрим приведенные в примере команды:

- первая команда: создает vacm группу с именем my_group;

- вторая команда: задает в группе my_group параметры доступа для пользователья user1. Будет использоваться USM модель протокола SNMPv3;
- третья команда: задает в группе my_group параметры доступа для модели myname. Будет использоваться протокол SNMPv2c.

Далее необходимо создать именованый список который определяет, какие OID будут доступны при получении информации о системе, а какие нет. Напимер:

```
(config—service—snmp)# vacm view not_all
(config—service—snmp—vacm—view—not_all)# included sysName.0
(config—service—snmp—vacm—view—not_all)# excluded sysUpTime.0
```

Рассмотрим команды подробнее:

- первая команда: создает список с именем not_all;
- вторая команда: включает в этот список sysName.0;
- третья команда: исключает из списка sysUpTime.0.

Завершающим этапом настройки является создание правил доступа, которые связывают списки OID с vacm группами, а также определяют уровень безопасности:

```
(config—service—snmp)# vacm access group my_group view not_all noauth
```

В данном примере создается правило по которому учасники группы **my_group** смогут получить информацию, описанную в списке **not_all**. Причем доступ будет осуществляться без аутентификации и шифрования, о чем говорит параметр **noauth**. Для протокола SNMPv3 также доступны: **auth** - только аутентификация (в этом случае должен быть создан USM-пользователь с аутентификацией) и **priv** - аутентификация и шифрование (в этом случае должен быть создан USM-пользователь с аутентификацией и шифрованием или TSM-пользователь).

Если используется протокол SNMPv3 и TSM-пользователь или USM-пользователь с аутентификацией (или с аутентификацией и шифрованием), а уровень безопасности в vacm правиле доступа указан **noauth**, то доступ будет возможен без аутентификации и шифрования.

36.5 Настройка правил нотификаций

Настройка правил, по которым другим узлам разрешено получать нотификации от службы SNMP. Например:

```
(config—service—snmp)# notify pas1 1.2.3.4
(config—service—snmp)# notify pas2 1.2.3.5:5555 v2
(config—service—snmp)# notify pas3 1.2.3.6:5556 v2c tcp
```

Рассмотрим третью команду в примере.

Первый обязательный параметр команды (pas3) - это пароль доступа, который должен быть установлен в конфигурации SNMP-клиента, который хочет получать нотификации.

Второй обязательный параметр команды (1.2.3.6) - это IP-адрес клиента, которому разрешено посылать нотификации.

Остальные параметры - это порт, версия нотификаций и транспортный протокол. По умолчанию,используется UDP-порт 162 и версия нотификаций v2c.

Для работы с адресами IPv6 используется соответствующая команда **notify6**.

Нотификации, посылаемые по умолчанию (не требуется настройка):

- NET-SNMP-AGENT-MIB::nsNotifyShutdown - посылается при выключении службы;
- SNMPv2-MIB::coldStart - посылается при включении службы;
- NET-SNMP-AGENT-MIB::nsNotifyRestart - посылается при перезапуске службы (например, во время настройки уже запущенной службы).
- FACTOR-DIONISNX-NOTIFICATION-MIB::fdnxSomeAdminLoggedIn - осуществлен вход первым администратором
- FACTOR-DIONISNX-NOTIFICATION-MIB::fdnxAllAdminLoggedOut - осуществлен выход последним администратором
- FACTOR-DIONISNX-NOTIFICATION-MIB::fdnxLoginRetriesExceeded - превышено количество попыток входа (вместе с нотификацией отправляется имя пользователя, под которым была осуществлена попытка входа)
- FACTOR-DIONISNX-NOTIFICATION-MIB::fdnxStartupConfigUpdated - стартовая конфигурация обновлена
- FACTOR-DIONISNX-CLUSTER-MIB::fdnxClusterStateChanged - изменилось состояние кластера

Дополнительно можно включить следующие типы нотификаций (команда указана после описания нотификации):

- SNMPv2-MIB::authenticationFailure - посылается при неуспешной аутентификации SNMP-клиента (например, когда он послал серверу неверный пароль).
- IF-MIB::linkDown/IF-MIB::linkUp - посылается при включении/отключении интерфейса.

Включение дополнительных нотификаций осуществляется командой:

```
(config—service—snmp)# trap auth
(config—service—snmp)# trap iface
```

Настройки правил нотификаций про протоколу SNMPv3 осуществляется в секции snmpv3. При этом предварительно должна быть настроена USM и/или TSM модель. Например:

```
(config—service—snmp)# snmpv3
(config—service—snmp—v3)# notify user4 10.0.0.1 dtlsudp inform.
```

В данном примере предполагается, что настроена модель TSM и соответствующий пользователь user4. На адрес 10.0.0.1 будет происходить отправка inform сообщений по защищенному каналу по протоколу udp (tls поверх udp).

36.6 Работа со службой

Для запуска службы выполните команду:

```
(config—service—snmp)# enable
```

Для остановки службы выполните команду:

```
(config—service—snmp)# disable
```

Для просмотра журналов службы выполните команду:

```
(config—service—snmp)# do show service snmp log
```

37. SSH

В рамках системы Dionis-NX протокол SSH не считается защищенным и, соответственно, его использования недостаточно для установления доверенного канала передачи данных. Для создания доверенных каналов передачи данных могут использоваться крипто-туннели.

37.1 Сервер SSH

В Dionis-NX реализована возможность удалённого доступа к командному интерфейсу по протоколу SSH.

Чтобы разрешить удалённый доступ к данному узлу, необходимо активировать службу SSH следующими командами (из режима конфигурации):

```
(config)# service ssh
(config—service—ssh)# enable
```

По умолчанию служба будет принимать SSH-соединения на всех интерфейсах, на IPv4-адресах. По-умолчанию использование IPv6 отключено. Есть возможность настроить службу для приёма SSH-соединений на одном локальном IP-адресе и/или изменить TCP порт по умолчанию (22). Для этого необходимо указать опцию «listen». Например:

```
(config—service—ssh)# listen 192.168.1.1
```

или

```
(config—service—ssh)# listen 0.0.0.0 2222
```

Кроте того есть возможность настроить службу для приёма SSH-соединений на локальных IP-адресах, которые находятся в VRF:

```
(config—service—ssh)# listen 192.168.1.1 vrf 1
```

Данная опция может быть очищена с помощью команды «no listen».

Для использования IPv6-адресов необходимо указать команду “listen6”:

```
(config—service—ssh)# listen6 fd55:1::ca60:ff:fe61:4177
```

Чтобы сервис ожидал соединений на любых IPv6-адресах, необходимо использовать следующую команду:

```
(config—service—ssh)# listen6 ::
```

Директива “listen6” может удалена из настроек с помощью команды “no listen6”.

Также можно отдельно задать порт для приема соединений:

```
(config—service—ssh)# port 2222
```

В этом случае указанный порт будет использоваться для всех слушающих IP-адресов, если в соответствующей команде «listen» порт не задан явно. Допустимо указание нескольких слушающих портов. Очистить опцию можно с помощью команды «no port»:

```
(config—service—ssh)# no port 2222
```

По умолчанию возможен доступ извне только к учётной записи «cli», которая обеспечивает доступ к командам непривилегированного режима, и для входа в привилегированный режим будет необходимо ввести команду «enable». Чтобы ускорить доступ к привилегированному режиму через учётную запись «adm», необходимо включить опцию:

```
(config—service—ssh)# permit—adm—login
```

Данная опция может быть очищена командой «no permit-adm-login».

По-умолчанию сервис SSH не использует DNS для получения имени удаленного хоста. Если необходимо получать имя удаленного хоста и затем проверять, что обратное преобразование имени в IP-адрес совпадает с исходным IP-адресом, то следует использовать директиву “use-dns”:

```
(config—service—ssh)# use—dns
```

Проверка имен может приводить к большим задержкам при подключении клиентов к SSH-серверу. Это происходит при некорректном или не полностью сконфигурированном DNS. На DNS может быть не настроена обратная зона для сопоставления имени IP-адресу. Для отключения проверки используется команда “no use-dns”.

Следует помнить, что если настройки службы редактируются при работающей (активированной) службе, необходимо перезапустить службу, чтобы настройки вступили в силу:

```
(config—service—ssh)# disable
(config—service—ssh)# enable
```

либо

```
(config—service—ssh)# reload
```

Чтобы остановить службу и удалить все настройки, нужно выполнить команду режима конфигурации:

```
(config)# no service ssh
```

37.1.1 Контроль доступа

Для контроля доступа к сервису ssh предусмотрены команды «allow» и «deny». Контроль осуществляется на основании имени пользователя и удаленного адреса, с которого происходит попытка соединения (user[@host]). В имени пользователя или удаленном адресе могут быть использованы шаблоны («*» - ноль или более символов, «?» - ровно один символ).

Команды «allow» и «deny» соответственно разрешают или запрещают доступ для указанных пользователей. Команда «deny» имеет более высокий приоритет, чем команда «allow». Команд «allow» и «deny» может быть несколько. Если используется команда «allow», то доступ будет предоставлен только указанным пользователям (и адресам). Всем остальным пользователям в доступе будет отказано. И наоборот, если используется команда «deny», то доступ будет запрещен указанным пользователям (и адресам). Всем остальным пользователям доступ будет разрешен.

Если команды «allow» и «deny» не используются - доступ по-умолчанию разрешен. Обратите внимание на то, что администратор adm имеет полный контроль над системой. Возможность удаленного доступа администратора является важным параметром безопасности и контролируется дополнительной командой режима конфигурирования сервиса ssh - «permit-adm-login» (команда описана выше). Для учетной записи администратора команды «allow» и «deny» лишь добавляют возможности по контролю доступа на основании удаленного адреса, с которого происходит попытка соединения.

Удаленный адрес может быть не указан. В этом случае будет учитываться только имя пользователя.

Следующая команда разрешит доступ оператору cli и запретит всем остальным (в данном случае adm):

```
(config—service—ssh)# allow cli
```

Следующая команда разрешит доступ оператору cli и администратору adm с удаленного адреса 192.168.2.3. Доступ с других адресов запрещен:

```
(config—service—ssh)# allow *@192.168.2.3
```

Следующая команда запретит доступ администратору adm из подсети 192.168.2.0/24. Доступ с других адресов разрешен. Доступ оператору cli разрешен с любых адресов:

```
(config—service—ssh)# deny adm@192.168.2.*
```

37.2 Клиент SSH

В Dionis-NX также реализован клиент SSH для удалённого доступа к другим узлам Dionis-NX. Команда доступна как из привилегированного, так и из непривилегированного режима. Формат команды:

```
> ssh <user> <host> [<port>]
```

Для узлов Dionis-NX в качестве <user> можно указывать учётные записи «cli» или «adm».

При обращении на удаленные узлы, информация об этих узлах заносится в список известных хостов (know-hosts). При повторном обращении на тот же удаленный узел, сравнивается сохраненный (при первом обращении) ключ удаленного хоста и текущий ключ удаленного хоста. Если ключи не совпадают, соединение считается небезопасным и связь не устанавливается. Это сделано для предотвращения подмены удаленного хоста. Для просмотра списка известных хостов используется следующая команда:

```
Router# show ssh known—hosts
```

Если администратор знает, что удаленный хост был легально заменен или изменился ключ удаленного хоста, он может удалить из списка известных хостов информацию о таком узле. Это позволит установить связь и сохранить новый ключ хоста в списке известных хостов:

```
Router# clear ssh known—hosts 192.168.1.33
```

также можно полностью очистить список известных хостов:

```
Router# clear ssh known—hosts all
```

37.3 Соединение с использованием ключей

Если соединение с удаленным узлом по протоколу SSH является частой операцией, неудобно каждый раз вводить пароль. Для установления соединения без использования паролей могут использоваться открытые ключи.

Если администратор, находясь на хосте А, хочет устанавливать соединение с хостом Б, то на хосте А он должен создать закрытый и открытый ключи:

```
A# ssh key generate
```

Затем открытый ключ должен быть отправлен на хост Б. В случае, если хост Б работает под управление системы Dionis-NX, администратор может выполнить команду:

```
A# ssh key export host adm 192.168.1.2
```

Предполагается, что хост Б имеет IP-адрес 192.168.1.2 и администратор желает устанавливать соединение используя учетную запись adm на удаленном хосте. Последним аргументом может быть указан удаленный порт. В данном случае порт не указан и, соответственно будет использован стандартный номер для SSH протокола - 22. После получения ключа, хост Б добавит этот ключ в список авторизованных ключей (authorized-keys). Все последующие соединения с хоста А на хост Б будут происходить без использования пароля.

Существует возможность полностью отключить использование паролей для авторизации. Для этого, находясь в режиме конфигурации службы SSH на хосте Б, воспользуйтесь командой:

```
(config—service—ssh)# no auth—passwd
```

Используйте команду auth-passwd, чтобы снова разрешить парольную авторизацию.

В случае, если хост Б работает под управлением операционной системы отличной от Dionis-NX, администратор может записать созданный открытый ключ в файл на хосте А:

```
A# ssh key export file open.key
```

Где open.key - произвольное имя файла, в который будет сохранен открытый ключ. После этого файл может быть скопирован на хост Б любым удобным способом и добавлен в список авторизованных ключей. Например в Linux-системах файл с авторизованными ключами находится в домашней директории пользователя и называется ~/.ssh/authorized_keys.

Если же на хосте А установлена система, отличная от Dionis-NX, администратор может скопировать файл с открытым ключом с хоста А на хост Б любым удобным способом, а затем добавить полученный ключ в список авторизованных ключей на хосте Б:

```
B# ssh key import file open.key
```

Где open.key - имя файла с открытым ключом.

Администратор может просмотреть список авторизованных ключей используя команду:

```
B# show ssh authorized—keys
```

Для более детального вывода, можно указать опцию «verbose».

```
B# show ssh authorized—keys verbose
```

Если какой-либо ключ больше не требуется, он может быть удален из списка авторизованных ключей:

```
B# clear ssh authorized—keys adm@A
```

Где «adm@A» идентифицирует ключ в списке авторизованных ключей и является последним полем при выводе на экран детального списка авторизованных ключей. Можно удалить сразу все ключи из списка авторизованных ключей:

```
B# clear ssh authorized—keys all
```

37.4 Передача файлов

Протокол SSH может использоваться для передачи файлов. Если администратор хочет получить файл с удаленного хоста, он может выполнить следующую команду:

```
Router# ssh get petrov 192.168.1.2 /tmp/test.txt
```

Где «petrov» - учетная запись на уделенном хосте, «192.168.1.2» - IP-адрес удаленного хоста, »/tmp/test.txt» - путь к файлу. Для отправки файла на удаленный хост, администратор может выполнить команду:

```
Router# ssh put test.txt petrov 192.168.1.2
```

Описанные команды соответствуют команде scp в Linux-системе. Файлы с Linux-системы (и других систем, поддерживающих протокол SSH) могут быть отправлены на хост с системой Dionis-NX.

38. Telnet

Система Dionis-NX имеет службу Telnet, реализующую сетевой протокол уровня приложений для создания текстового интерфейса по сети.

38.1 Настройка

Для входа в режим конфигурации службы следует выполнить команду:

```
(config)# service telnet
```

Для настройки сокета, на котором служба будет ожидать Telnet-соединение, выполните:

```
(config—service—telnet)# listen 192.168.0.1 1023
(config—service—telnet)# listen6 fe80::be5f:f4ff:fec4:7edd
```

Если порт не указан, по умолчанию используется порт 23. В данном случае мы предписываем службе принимать запросы на адресе 192.168.0.1 и использовать порт 1023, и на адресе fe80::be5f:f4ff:fec4:7edd и использовать порт 23.

Возможно указание множества сокетов для принятия соединений.

По умолчанию,если не указано ни одной опции listen и listen6, служба ожидает соединения на всех доступных IPv4 адресах и на порту 23.

Чтобы включить службу выполните

```
(config—service—telnet)# enable
```

Чтобы выключить службу выполните

```
(config—service—telnet)# disable
```

Если служба включена, для изменения ее опций выполните:

- измените нужную опцию
- выполните команду disable
- выполните команду enable

39. Служба DIWEB

В Dionis-NX реализована возможность удалённого доступа к визуальному Web-интерфейсу по протоколу HTTP. Сервис позволяет конфигурировать часть функций системы.

В текущей версии Dionis- NX работа через Web-интерфейс возможна только с помощью учетной записи “adm”. При этом учетная запись “adm” должна иметь статус “supervisor”, т.е. иметь полный доступ к возможностям системы.

Важно: для безопасной эксплуатации рекомендуется, после настройки изделия, выключать службу Diweb.

Чтобы разрешить удалённый доступ к данному узлу, необходима активировать службу diweb командами (из режима конфигурации):

```
(config)# service diweb
(config—service—web)# enable
```

По умолчанию служба будет принимать соединения на всех интерфейсах. Есть возможность настроить службу для приёма HTTP-соединений на одном локальном IP-адресе и/или изменить порт по умолчанию (80). Для этого необходимо указать опцию «listen». Например:

```
(config—service—web)# listen 192.168.1.1
```

или

```
(config—service—web)# listen 0.0.0.0 8080
```

Данная опция может быть очищена с помощью команды «no listen».

Аналогично команде “listen” работает команда “listen6” для IPv6 адресов. Например:

```
(config—service—web)# listen6 ::
```

или

```
(config—service—web)# listen6 :: 8080
```

Опция очищается с помощью команды «no listen6».

Доступно ssl-шифрование, включается командой “ssl”:

```
(config—service—web)# ssl
```

При этом будет использоваться пара ключ/самоподписанный сертификат, созданные в системе по умолчанию.

Если требуется указать определенную пару ключ/сертификат, необходимо выполнить:

```
(config—service—web)# ssl cert cert:/pair.pem
```

Опция очищается командой “no ssl”.

Следует помнить, что если настройки службы редактируются при работающей (активированной) службе, необходимо перезапустить службу, чтобы настройки вступили в силу:

```
(config—service—web)# disable
(config—service—web)# enable
```

Чтобы остановить службу и удалить все настройки, нужно выполнить команду режима конфигурации:

```
(config)# no service diweb
```

40. Служба TOX

Tox — протокол для децентрализованной текстовой, голосовой и видеосвязи в интернете на основе асимметричного шифрования.

Dionis может выступать в качестве рандеву-ноды (bootstrap-node) для tox-клиентов и использует для этого протокол DHT.

Минимальная настройка сервиса представляет собой генерацию ключевой пары и указание этой ключевой пары в конфигурации службы.

Пример минимальной конфигурации службы.

```
# service tox key generate mykey
(config—service—tox)# do show
keypair mykey
enable
```

Указать bootstrap-node для получения списка с такими же нодами можно командой:

```
bootstrap—node <ipv4/6_address|host>[A.B.C.D|A:B:...:H|Server_FQDN] port [0..65535] public—key
        [NAME]
```

Список нод можно найти на странице проекта wiki.tox.chat.

Порядок добавления нод:

1. Добавить публичный ключ:

   ```
   # service tox key add velusip
           7E5668E0EE09E19F320AD47902419331FFEE147BB3606769CFBE921A2A2FD34C
   ```

2. Просмотреть добавленный ключ:

   ```
   # show service tox key velusip
   velusip: 7E5668E0EE09E19F320AD47902419331FFEE147BB3606769CFBE921A2A2FD34C
   ```

3. Добавить ноду в конфигурацию сервиса.

   ```
   (config—service—tox)# bootstrap—node 144.217.167.73 port 33445 public—key velusip
   ```

4. Аналогичным образом можно добавить и другие ноды из списка.

   ```
   (config—service—tox)# do show
   keypair mykey
   bootstrap—node 144.217.167.73 port 33445 public—key velusip
   bootstrap—node 2604:a880:1:20::32f:1001 port 33445 public—key cody
   bootstrap—node tox.abilinski.com port 33445 public—key abilinski
   enable
   ```

Остальные возможные настройки сервиса:

ipv6	Использовать ipv6 для отправки сообщений
ipv4-fallback	Переходить на ipv4 в случае невозможности использования ipv6.
keypair [NAME]	Указать локальную ключевую пару.
lan-discovery	Попытаться автоматически определить bootstrap-node в локальной сети.
message-of-the-day [STR]	Сообщение дня для тех, кто его запросит, должно поместиться в 255 байт.
port [0..65535]	Udp-порт на котором будет работать нода(по умолчанию - 33445).
tcp-relay-port	Указать использование tcp-порта (дополнительно к udp-порту).

Команды для работы с ключами:

service tox key add [key_name] [hex]	Добавить публичный ключ службы tox (из HEX).
service tox key export [[*]|[skey]] [ofname] [private]	Экспортировать ключи службы tox.
service tox key generate [key_name]	Сгенерировать ключевую пару службы tox.
service tox key import [ifname]	Импортировать ключи службы tox.
service tox key remove [key_name]	Удалить ключ службы tox.

Для перезапуска службы можно использовать команду:

```
# service tox restart
```

Для просмотра лога службы можно использовать команду:

```
# show service tox log
```

Для просмотра информации о ключах, добавленных в службу, можно использовать команду:

```
# show service tox key
```

Пример полной конфигурации службы:

```
(config—service—tox)# do show
keypair mykey
port 33445
tcp—relay—port 443
ipv6
ipv4—fallback
lan—discovery
message—of—the—day ”Hello from Factor—TS!”
bootstrap—node 144.217.167.73 port 33445 public—key velusip
bootstrap—node 2604:a880:1:20::32f:1001 port 33445 public—key cody
bootstrap—node tox.abilinski.com port 33445 public—key abilinski
enable
```

41. Сервис XMPP

В Dionis-NX реализован сервис xmpp (Jabber)

Приведем пример минимально необходимых настроек для запуска сервиса xmpp:

```
(config)# service xmpp
(config—service—xmpp)# host 192.168.1.1
```

Таким образом мы настраиваем сервер принимать соединения на виртуальном хосте 192.168.1.1. При этом вместо IP-адреса может быть указано доменное имя.

Чтобы запустить сервер необходимо выполнить команду

```
(config—service—xmpp)# enable
```

Сервер будет ожидать соединения на порту 5222, «plain»-аутентификация и шифрование отключены.

Если потребуется поддержка «plain»-аутентификации, необходимо выполнить комманду в настройках виртуального хоста:

```
(config—service—xmpp—192.168.1.1)# allow—plain—auth
```

Для включения ssl/tls шифрования потребуется выполнить команду:

```
(config—service—xmpp—192.168.1.1)# ssl
```

Отключение отдельного виртуального хоста производится командой «off»:

```
(config—service—xmpp—192.168.1.1)# off
```

Разрешение регистрации пользователей включается командой «allow-client-reg»:

```
(config—service—xmpp—192.168.1.1)# allow—client—reg
```

Данные опции могут быть очищены с помощью команд «no allow-plain-auth», «no ssl», «on» и "no allow-client-reg" соответственно.

По умолчанию ssl/tls используют самоподписанный сертификат и ключ, автоматически созданные в системе. Для указания специфического сертификата и ключа для отдельного хоста или сервиса используется команда certs. Параметром к ней выступает путь к каталогу, в котором они размещаются.

```
(config—service—xmpp)# certs cert:/xmpp
```

Сертификаты и ключи загружаются автоматически и именуются по определенному правилу:

Certificate file	Key file
HOSTNAME.crt	HOSTNAME.key
HOSTNAME/fullchain.pem	HOSTNAME/privkey.pem
HOSTNAME.pem	HOSTNAME.pem
SERVICE.crt	SERVICE.key

Например, для сервиса https сертификат и ключ будут иметь имена файлов https.crt и https.key

соответственно. Если certs указывает на cert:/xmpp, файлы для сервиса https должны быть размещены как cert:/xmpp/https.crt и cert:/xmpp/https.key.

Дополнительными возможностями виртуального хоста является поддержка конференций (Multi-User Chat) и передача файлов по протоколу http/https. Данные настройки требуют указать вместо IP-адреса доменное имя для виртуального хоста сервера. Для включения поддержки конференций на виртуальном хосте fts.ru необходимо выполнить команды:

```
(config)# service xmpp
(config—service—xmpp)# host fts.ru
(config—service—xmpp—fts.ru)# muc
(config—service—xmpp)# enable
```

Для включения поддержки передачи файлов на виртуальном хосте fts.ru необходимо выполнить команды:

```
(config)# service xmpp
(config—service—xmpp)# host fts.ru
(config—service—xmpp—fts.ru)# http—upload
(config—service—xmpp)# enable
```

Необязательными дополнительными параметрами команды **http-upload** являются команды **expire-limit** (Время хранения файла на сервере. По истечению данного периода файл буде удален. По умолчанию период составляет 7 дней), **quita-limit** (Общий размер хранилища для файлов на сервере. По умолчанию 10Mb) и **size-limit** (Максимальный размер передаваемого файла. По умолчанию 1Mb).

Пример команды описывающей время хранения файлов на сервере равное 1 день, максимальный размер хранилища файлов 100Mb и максимальный размер передаваемого файла 500Kb:

```
(config—service—xmpp—fts.ru)# http—upload expire—limit 1 quita—limit 100mb size—limit 500kb
```

По умолчанию служба будет принимать соединения на всех интерфейсах. Есть возможность настроить службу для приёма соединений на нескольких локальных IP-адресах и/или изменить порт по умолчанию (5222). Для этого необходимо требуемое число раз указать опцию «listen» и/или «port». Например:

```
(config—service—xmpp)# listen 192.168.1.1
```

и/или

```
(config—service—xmpp)# port 5223
```

Для протокола IPv6 используйте вместо «listen» «listen6». Например:

```
(config—service—xmpp)# listen6 2001:db8::2
```

Все эти опции могут быть очищены с помощью команд «no listen 192.168.1.1», «no port 5223» и «no listen 2001:db8::2» соответственно.

Следует помнить, что если настройки службы редактируются при работающей (активированной) службе, необходимо перезапустить службу, чтобы настройки вступили в силу:

```
(config—service—xmpp)# disable
(config—service—xmpp)# enable
```

Чтобы остановить службу и удалить все настройки, нужно выполнить команду режима конфигурации:

```
(config)# no service xmpp
```

Для создания пользователя “user” в enable режиме требуется ввести комманду:

```
# service xmpp account create 192.168.1.1 user passwd
```

Для удаления пользователя “user” в enable режиме требуется ввести комманду:

```
# service xmpp account remove 192.168.1.1 user
```

Для изменения пароля пользователя “user” в enable режиме требуется ввести комманду:

```
# service xmpp account password 192.168.1.1 user newpasswd
```

42. Служба netperf

42.1 Настройка службы netperf режима configure

Позволяет запустить службу измерения пропускной способности сети.

Для измерения пропускной способности между узлами А и Б, на которых установлены изделия Dionis-NX:

- запустите данный сервис на узле А;
- используйте команду netperf режима enable на узле Б, в которой в качестве первого параметра задайте IP-адрес узла А.

Чтобы войти в режим конфигурации службы, следует использовать команду:

```
(config)# service netperf
```

Если необходимо, порт, на котором будет запущена служба, может быть настроен при помощи команды:

```
(config—service—netperf)# listen 12345
(config—service—netperf)# listen6
```

По умолчанию, если не указано ни одной опции listen и listen6, служба ожидает соединения на всех доступных IPv4 адресах и на порту 12865. В данном случае мы предписываем службе принимать запросы на адресах IPv4 и порту 12345 и на адресах IPv6 и порту 12865

Включить службу следует командой:

```
(config—service—netperf)# enable
```

Для выключения службы следует выполнить команду:

```
(config—service—netperf)# disable
```

42.2 Команда netperf режима enable

netperf <IP[:PORT] | IPv6[:PORT] | HOST[:PORT]> <TEST> [time TIME] [cycles NUM] [local-sock-size LSZ] [remote-sock-size RSZ] [message-size MSZ] [verbose <low|average|hi>] [nodelay <both|local|remote>] [source <IP|IPv6>]

Данная команда осуществляет тестирование пропускной способности сети. Имеет два обязательных параметра:

- IP или IPv6 или HOST - IPv4 или IPv6 адрес или имя узла, на котором запущена служба netperf;
- TEST - тип теста: tcp или udp.

Остальные параметры являются необязательными:

- PORT - задает порт, на котором запущена служба netperf (по умолчанию 12865);
- TIME - время одного цикла теста в секундах (по умолчанию: 10);
- NUM - число циклов теста (по умолчанию: 1);
- LSZ - размер локального буфера сокета (по умолчанию: 64000 байт);
- RSZ - размер удаленного буфера сокета (по умолчанию: 64000 байт);
- MSZ - размер сообщения;
- verbose - задает уровень подробности выдачи команды (низкий,средний или высокий);
- nodelay - включает на локальном ,удаленном или обоих сокетах опцию TCP_NODELAY, которая может ускорить передачу большого количества данных маленького размера (частые посылки);
- source - задает локальный IPv4 или IPv6 адрес сокета для соединения с узлом, на котором запущена служба netperf.

43. Служба IPERF

43.1 Настройки службы iperf режима configure

Позволяет запустить службу iperf для измерения пропускной способности сети.

Для измерения пропускной способности между узлами А и Б, на которых установлены изделия Dionis-NX:

- следует запустить данный сервис на узле А;
- затем необходимо выполнить команду iperf режима enable на узле Б; в этой команде в качестве первого параметра следует задать IP-адрес узла А.

Чтобы войти в режим конфигурации службы, следует выполнить команду:

```
(config)# service iperf
```

Следует задать тип теста и ,если необходимо, пару IP-адрес и порт, на котором будет запущена служба:

```
(config—service—netperf)# listen udp 10.0.0.1 12345
```

По умолчанию: listen tcp 0.0.0.0 5001

Для задания IPv6 адреса используется команда **listen6**:

```
(config—service—netperf)# listen6 tcp fe80::be5f:f4ff:fec4:7edd 12346
```

Другие варианты типов теста: udp (UDP-тест), udp-single(однопоточный UDP-тест).

Включить службу следует командой:

```
(config—service—netperf)# enable
```

Для выключения службы следует выполнить команду:

```
(config—service—netperf)# disable
```

43.2 Команда iperf режима enable

iperf <IP [PORT] | IPv6 [PORT]> <TEST> [SPEED] [time TIME] [cycles NUM] [threads NT] [size SOCKSZ] [tos <relaiablity|throughput|delay|TOS>] [window-size WIN] [mss MSS] [bidirectional BPORT <simultaneous|individual>] [source <IP|IPv6>]

Данная команда осуществляет тестирование пропускной способности сети. Имеет два обязательных параметра:

- IP - IPv4 или IPv6 адрес узла, на котором запущена служба iperf;

- TEST - тип теста: tcp или udp.

Остальные параметры являются необязательными:

- PORT - задает порт, на котором запущена служба iperf;
- SPEED - задает скорость передачи данных в службу iperf (по умочанию: 1Mbit/s, только для udp теста);
- TIME - время одного цикла теста в секундах (по умолчанию: 10);
- NUM - число циклов теста (по умолчанию: 1);
- NT - число потоков (по умолчанию: 1);
- SOCKSZ - размер буфера сокета (по умолчанию: 8Кб);
- tos - тип обслуживания: задается либо байтом TOS, либо relaiability - высокая надежность, throughput - высокая пропускная способность, delay - низкая задержка передачи IP-сегмента;
- WIN - размер TCP-окна (по умолчанию: 16Кб, только для tcp-теста);
- MSS - максимальный размер сегмента TCP (только для tcp-теста);
- bidirectional - задает двунаправленный тест - передача данных в обе стороны последовательно, BPORT - локальный порт для двунаправленного теста;
- source - задает локальный IPv4 или IPv6 адрес сокета для соединения с узлом, на котором запущена служба iperf.

44. Служба SLAGENT

Система Dionis-NX имеет службу slagent, реализующую возможность генерации эхо-пакетов полученных по протоколу UDP.

А также команды **udp-ping** и **udp6-ping** для генерации запросов, приема эхо-пакетов и проверки целостности и качества соединений по протоколу UDP.

44.1 Настройка службы slagent из режима configure

Чтобы войти в режим конфигурации службы, следует выполнить команду:

```
(config)# service slagent
```

Задайте пары адрес-порт, на которых система будет слушать UDP-трафик:

```
(config—service—slagent)# listen 10.0.0.1 1000
(config—service—slagent)# listen 10.0.0.1 1001
(config—service—slagent)# listen 10.0.0.2 1000
```

Для прослушивания UDP-трафика на IPv6 адресах используется команда **listen6**:

```
(config—service—slagent)# listen6 fe80::be5f:f4ff:fec4:7edd 1005
```

Можно задать максимум 1000 пар адрес-порт.

Для включения журнала службы следует выполнить команду:

```
(config—service—slagent)# log
```

Для включения службы следует выполнить команду:

```
(config—service—slagent)# enable
```

Для выключения службы следует выполнить команду:

```
(config—service—slagent)# disable
```

44.2 Герерация UDP-запросов

Для генерации UDP-запросов на IPv4 адреса используется команда udp-ping из режима enable.

udp-ping <IP> <PORT> [size <SZ>] [indefinite | repeat <R>]

Для генерации UDP-запросов на IPv6 адреса используется команда udp-ping6 из режима enable.

udp-ping6 <IPv6> <PORT> [size <SZ>] [indefinite | repeat <R>]

Данные команды имеют по два обязательных параметра:

- IP или IPv6 - IPv4 адрес (для udp-ping) или IPv6 адрес (для udp-ping6) узла на котором запущена служба slagent;
- PORT - задает порт, на котором запущена служба slagent;

Остальные параметры являются необязательными:

- size <SZ> - размер пакета;
- repeat <R> - количество запросов (по умолчанию 4 запроса)
- indefinite - бесконечное число запросов (Для прерывания генерации запросов необходимо нажать на клавиатуре Ctrl-C.)

45. Служба LLDP

Dionis-NX имеет службу LLDP, реализующую протокол канального уровня, который позволяет сетевым устройствам анонсировать в сеть информацию о себе и о своих возможностях, а также собирать эту информацию о соседних устройствах.

LLDP-информация может получаться и отправляться (анонсироваться) только с/на непосредственно подключенные к данной системе устройства (в дальнейшем - соседи), либо подсоединенные через концентратор или повторитель. Анонсируемая данной системой информация, будучи полученной соседями, не пересылается далее по сети. Переданные LLDP анонсы не требуют своего подтверждения или какой-либо другой реакции от получателей данных анонсов.

Таким образом, LLDP-служба позволяет данной системе:

- передавать LLDP-информацию о себе соседям;
- получать LLDP-информацию от соседей;
- сохранять и управлять полученной LLDP-информацией в LLDP MIB, которую можно получить удаленно, если включена служба snmp на системе.

LLDP-служба передает анонсы в виде пакетов, называемых LLDP PDU, состоящих из набора TLV элементов, каждый из которых содержит информацию определенного типа об устройстве, либо о сетевом порте, который передает данный анонс.

Служба LLDP позволяет настраивать различные параметры для отдельных физических интерфейсов (в дальнейшем - портов), либо для всех вместе.

45.1 Базовые настройки службы и настройки обязательных TLV

Описываемые в данном разделе обязательные TLV соответсвуют стандарту Mandatory Base TLVs—IEEE 802.1AB-2005.

Команды данного раздела позволяют задать такой обязательный TLV как TTL. Другие обязательные TLV формируются и передаются службой автоматически (Port ID TLV и Chasis ID TLV).

Чтобы войти в режим настройки службы, следует выполнить команду:

```
(config)# service lldp
```

Чтобы указать порт, который служба будет использовать для приема и посылки LLDP-фрэймов, следует выполнить команду:

```
(config—service—lldp)# listen <IFACE>
```

По умолчанию: использовать для LLDP все порты.

Следующая команда задает длительность задержки между посылками LLDP-фрэймов:

```
(config—service—lldp)# tx—interval <NSEC>
```

По умолчанию: 30 сек.

Следующая команда определяет параметр для расчета TTL отправленного LLDP-пакета по следующей формуле:

TTL = <значение tx-multiplier> *<значение tx-interval>

```
(config—service—lldp)# tx—multiplier <NSEC>
```

По умолчанию: 4.

Значение TTL в LLDP-пакете определяет промежуток времени, в течение которого информация, полученная в данном LLDP-пакете, остается актуальной.

Указать поддерживаемые, помимо LLDP, протоколы:

```
(config—service—lldp)# proto <all|cdp|fdp|sdp|edp>
```

Параметры команды:

- cdp - Cisco Discovery Protocol;
- fdp - Foundry Discovery Protocol;
- edp - Extreme Discovery Protocol;
- sdp - SynOptics Network Management Protocol;
- all - все перечисленные выше протоколы.

По умолчанию: не включать поддержку дополнительных протоколов.

Включить режим службы, при которой она только принимает LLDP-пакеты, однако не передает их, т.е. не анонсирует свою информацию:

```
(config—service—lldp)# read—only
```

По умолчанию: режим отключен.

Команда задает тип MAC-адреса отправителя в LLDP-фреймах, посылаемых на подчиненные bond-интерфейсы:

```
(config—service—lldp)# bond—slave—mac <real|zero|fixed|local>
```

Параметры:

- real - настоящий мак-адрес подчиненного интерфейса;
- zero - нулевой мак-адрес;
- fixed - фиксированный мак 00:60:08:69:97:ef;
- local - настоящий мак-адрес подчиненного интерфейса, с установленным специальным битом в мак-адресе; если данный бит уже взведен в настоящем мак-адресе - используется фиксированный мак-адрес 00:60:08:69:97:ef.

По умолчанию: bond-slave-mac local.

45.2 Настройка опциональных TLV (DOT1)

Описываемые в данном разделе опциональные TLV соответсвуют стандарту Optional Base TLVs—IEEE 802.1AB-2005.

Команды данного раздела позволяют задать следующие опциональные TLV:

- описание системы;
- описание порта;
- имя системы;
- IP-адрес управления.

Пример настройки базовой анонсируемой информации о системе:

```
(config—service—lldp)# sys hostname host1
(config—service—lldp)# sys description ”main core”
(config—service—lldp)# sys iface—descr
```

Первая команда задает имя текущего хоста для аносирования вместо реального системного имени хоста. Вторая команда задает описание системы вместо стандартного описания, в которое входит: версия и название ядра системы, имя хоста, дата сборки и тип процессорной архитектуры системы. Последняя команда указывает, что вместо заданного в системе описания интерфейса нужно анонсировать имя соседнего узла или их число.

Указать управляющий IP-адрес системы следует при помощи команды:

```
(config—service—lldp)# management—address <IP>
```

Это адрес, который используется для получения LLDP-информации с текущей системы.

По умолчанию: использовать первый найденный IP-адрес текущей системы.

45.3 Настройка опциональных TLV (DOT3)

Описываемые в данном разделе опциональные TLV соответствуют стандарту IEEE 802.3 Organizationally Specific TLVs (802.3 TLVs)—IEEE 802.1AB-2005 Annex G.

Следующая команда задает TLV, описывающий POE-MDI-опции для порта - параметры передачи энергии через MDI-портов.

Анонсирование PoE(Power over Ethernet)-информации для MDI-портов:

```
(config—service—lldp)# dot3 power < *|IFACE > <TYPE> powerpairs <PWP> [supported] [enabled]
        [paircontrol] [CLASS] [802.3at type <ATYPE> source <SRC> prio <PRIO> <PWREQ> <PWALC>]
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;

- TYPE - задает энергетический тип порта: источник энергии (pse), либо получатель энергии (pd);
- PWP - способ передаче энергии по витой паре:
  - spare - использовать 2 свободные витые пары кабеля Ethernet (те,что не используются для передачи данных);
  - signal - использовать витые пары кабеля, занятые в передаче данных;
- suppored - данный порт поддерживает передачу энергии через MDI;
- enabled - на данном порту включена передача энергии через MDI;
- paircontrol - контроль выбора витых пар для передачи энергии;
- CLASS - класс энергии (от 0 до 4);
- ATYPE - определяет типа стандарта 803.3at: type 1, либо type 2;
- source - определяет источник энергии для данного порта:
  - для TYPE=pd:
    * unknown - неизвестный источник энергии;
    * pse - источник энергии;
    * local - источник энергии - локальный;
    * both - источник энергии - локальный pse;
  - для TYPE=pse:
    * unknown - неизвестный источник энергии;
    * primary - первичный источник энергии;
    * backup - резеврный источник энергии (например,UPS);
- PRIO - приоритет источника энергии (неизвестный (unknown), критичный(critical), высокий(high) или низкий(low));
- PWREQ - требуемая мощность в миливаттах;
- PWALC - выделяемая мощность в миливаттах.

Справочная таблица для CLASS (мощность указана в Ваттах):

Класс мощности	Мощность для PD	Мощность от PSE
0	0.44-12.95	15.4
1	0.44-3.84	4.0
2	3.84-6.49	7.0
3	6.49-12.95	15.4
4	12.95-25.5	30

Приоритет PRIO определяет насколько данный порт важен для обеспечения его энергией: в случае приоритета critical у порта - даже при недостатке мощности у PSE будет сделано все (отключены от питания другие,менее приоритетные порты) для обеспечения мощностью данного порта.

45.4 Настройка расширения LLDP-MED

LLDP-MED - это расширение LLDP-протокола для оконечных медиа-устройства (Media Endpoint Devices, MED), используемое для LLDP-взаимодействия между сетевыми устройствами LAN (маршрутизаторы, концентраторы и др.) и оконечными медиа-устройствами, подсоединенными к ним, например IP-телефонами.

Настройка класса устройств, для которых будет анонсироваться LLDP-MED информация, осуществляется с помощью команды:

```
(config—service—lldp)# med transmit <class1|class2|class3|class4>
```

Классы устройств означают следующее:

- class1 - эти устройства поддерживают базовые возможности обнаружения по LLDP, анонсирование сетевых политик (VLAN ID, приоритеты 802.1p и DSCP), управление PoE; этот класс включает такие устройства как IP контроллеры вызовов и communication-related сервера;
- class2 - включает в себя class1 плюс возможности передачи медиаинформации; это такие устройства, как голосовые/медиа-шлюзы, устройства для конференц-связи,медиа-сервера;
- class3 - включает в себя class2 плюс идентификацию местоположения, номер экстренной связи(ELIN), поддержку коммутатора 2го уровня, управление информацией об устройстве; как правило, это IP-телефоны или софт-телефоны;
- class4 - прочие стевые устройства: хабы,мосты,сетевые карты,маршрутизаторы и другие устройства сетевого взаимодействия.

Анонсирование физического адреса местонахождения портов следует задавать при помощи команды:

```
(config—service—lldp)# med address < *|IFACE > <CC> city <CITY> street <STR> bld <BLD> app
        <APP>
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;
- CC - код страны;
- CITY - город;
- STR - улица;
- BLD - номер строения;
- APP - номер квартиры/комнаты в указанном строении BLD.

Анонсирование географических координат местонахождения портов следует задавать при помощи команды:

```
(config—service—lldp)# med coordinate < *|IFACE > latitude <LAT N|S> longtitude <LNG W|E> altitude
        <ALT> <DATUM>
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;
- LAT - широта (число с плавающей точкой), N - северная широта, S - южная широта;
- LNG - долгота (число с плавающей точкой), W - западная долгота, E - восточная долгота;
- ALT - высота над уровнем моря (метры, число с плавающей точкой);
- DATUM - датум (nad83, wgs84 или nad83-mllw).

Анонсирование номера экстренной связи следует задавать при помощи команды:

```
(config—service—lldp)# med elin < *|IFACE > <ELIN>
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;
- ELIN - строка,задающая номер экстренной связи, например 112 (для Европы), 911(для США).

Большинство IP-телефонов (или других IP-устройств для передачи голоса) имеют два интерфейса: один для соединения с сетью и другой для соединения с другим устройством, например компьютером, в результате чего компьютер может подсоединиться к сети через IP-телефон. Желательно различать информационный и голосовой трафик так, чтобы разные параметры QoS применялись для каждого типа трафика.

Следующая команда позволяет передать MED-устройству значения параметров VLAN ID и DSCP для выбранного типа трафика. В результате MED-устройство (например, IP-телефон) будет передавать голосовой трафик исходя из сконфигурированных значений, а обычный информационный трафик, идущий с данной системы через MED-устройство, будет передаваться, исходя из значений по умолчанию.

Анонсирование сетевой политики для портов следует задавать при помощи команды:

```
(config—service—lldp)# med policy < *|IFACE > <ATYPE> [unknown] [vlan <VID>] [dscp <DSCP>] prio
      <PRIO>
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;
- ATYPE - задает тип приложения:
  - voice - передача голоса/телефония;
  - voice-signaling - голосовая сигнализация;
  - guest-voice;
  - guest-voice-signaling;
  - softphone-voice - программные телефоны;
  - video-conferencing - видео-конференция;
  - streaming-video - потоковое видео;
  - video-signaling - видео-сигнализация;
- VID - анонсируемое VLAN ID, используемое для сетевой политики указанного типа приложения;
- DSCP - анонсируемое значение DSCP (от 0 до 63), используемое для сетевой политики указанного типа приложения;
- PRIO - значение CoS, используемое для указанного типа приложения;

- unknown - сетевая политика неизвестна для данного типа приложения.

С помощью следующей команды можно настроить параметры PoE. PoE - это механизм для передачи мощности сетевому устройству по тому же кабелю, который используется устройством для передачи сетевых данных. Понятия используемые в данной команде:

- Powered Device (PD) - устройства потребляющие энергию;
- Power Sourcing Equipment (PSE) - устройства, являющиеся источниками энергии.

Анонсирование PoE(Power over Ethernet)-параметров следует задавать при помощи команды:

```
(config—service—lldp)# med power < *|IFACE > <TYPE> source <SRC> prio <PRIO> <PWREQ>
```

Параметры:

- <*|IFACE> - задает все порты, либо конкретный порт системы;
- TYPE - задает энергетический тип порта: источник энергии (pse), либо получатель энергии (pd);
- source - определяет источник энергии для данного порта:
  - для TYPE=pd:
    - unknown - неизвестный источник энергии;
    - pse - источник энергии - pse;
    - local - источник энергии - локальный;
    - both - источник энергии - локальный pse;
  - для TYPE=pse:
    - unknown - неизвестный источник энергии;
    - primary - первичный источник энергии;
    - backup - резеврный источник энергии (например,UPS);
- PRIO - приоритет источника энергии (неизвестный (unknown), критичный(critical), высокий(high) или низкий(low));
- PWREQ - значение мощности в миливаттах, требуемое (для pd), либо доступное (для pse).

Приоритет PRIO определяет, насколько данный порт важен для обеспечения его энергией: в случае приоритета critical у порта - даже при недостатке энергии у PSE будет сделано все (отключены от питания другие,менее приоритетные порты) для обеспечения данного порта энергией.

Пример PD-устройств: WAP, VoIP-устройства, IP-камеры.

Пример PSE-устройств: коммутатор.

45.5 Работа со службой

Следующие команды задают передачу различной информации о службе. Введем ряд параметров, которые могут быть показаны администратору в результатах выдачи команд данного раздела:

- Chassis ID - идентификатор платформы, например, это может быть MAC-адрес;
- Port ID - порт, который послал LLDP PDU.

Для запуска службы следует выполнить команду:

```
(config—service—lldp)# enable
```

Для остановки службы следует выполнить команду:

```
(config—service—lldp)# disable
```

Для просмотра журналов службы следует выполнить команду:

```
(config—service—lldp)# do show service lldp log
```

Для просмотра статуса работы службы следует выполнить команду:

```
(config—service—lldp)# do show service lldp status
```

Для просмотра статистики работы службы следует выполнить команду:

```
(config—service—lldp)# do show service lldp statistics [summary] [IFACE]
```

Если указан порт - будет показана информацию по нему. Если указан параметр summary - будет показана краткая статистическая сводка по работе службы.

Для просмотра информации по физическим сетевым портам следует выполнить команду:

```
(config—service—lldp)# do show service lldp neighbors [verbose|summary] [hidden]  [IFACE]
```

Если указан порт - будет показана информацию по нему. Если указан параметр verbose - будет показана более подробная информация. Если указан параметр hidden - будет показана информация об удаленных интерфейсах. Если указан параметр summary - будет показана краткая статистическая сводка.

46. Служба предотвращения вторжений IDSM.

Dionis-NX имеет службу предупреждения и предотвращения вторжений idsm (Intrusion Detection System with Multiprocessing).

Служба IDSM - это система предотвращения сетевых вторжений (атак), основанная на правилах.

46.1 Режимы

Служба IDSM может работать в 2х режимах:

- ids - система обнаружения вторжений;
- ips - система обнаружения и предупреждения вторжений.

46.2 Путь пакета через службу idsm

Рассмотрим путь сетевого пакета через службу:

1. **Выборка пакетов**: осуществляется выбор сетевого пакета из сетевого интерфейса или получение его из подсистемы ядра netfilter/nfqueue через iptables; передача пакета службе idsm для анализа.
2. **Декодирование пакета**: проверка всех инкапсулированных пакетов на предмет аномалий соответствующих им сетевых протоколов
3. **Препроцессинг пакета**: определение цели и содержимого самого вложенного пакета; это может включать в себя пересортировку IP-фрагментов и пересборку TCP-сегментов для получения оригинального пакета (PDU), посланного сетевым узлом; полученный PDU анализируется и нормализуется для дальнейшей обработки службой.
4. **Обнаружение уязвимости**: если пакет попадает под условия заголовка правила (совпадение IP адресов, портов и протокола), то осуществляется поиск по образцу, который может быть задан в правиле; если образец найден, то анализируются остальные условия правила.
5. **Создание события и реакции для пакета**: может включать в себя блокирование, отбрасывание пакета и другие действия с пакетом, определяемые действием правила, под которое попал данный пакет; также сюда относится посылка ответного сообщения или иной реакции на событие, например TCP Reset.
6. **Журналирование события**: запись,если необходимо, результатов проверки пакета

Рассмотрим далее основные особенности и основные команды настройки перечисленных выше подсистем службы.

46.3 Настройка подсистема выборки пакетов

Реализована через отдельную библиотеку daq, которая используется службой idsm.

Цель подсистемы выборки пакетов (Data Acqustion, DAQ) - взять пакет из сетевой подсистемы ядра и передать его на обработку другим подсистемам службы.

daq mode <nfq|nfq6|afpacket>

Команда задает режим выборки:

- nfq - через подсистему NFQUEUE для IPv4 трафика
- nfq6 - - через подсистему NFQUEUE для IPv6 трафика
- afpacket - считывание пакета непосредственно с одного сетевого интерфейса и передача его в другой сетевой интерфейс (возможно, тот же самый)

По умолчанию: nfq.

daq afpacket bufsize <VAL>

Данная команда позволяет настроить размер буфера <VAL> (в Мб), выделяемого для подсистемы выборки пакетов.

Тип: глобальный. Режим: ids. *По умолчанию: 128 Мб*

daq snaplen <VAL>

Данная команда определяет, сколько байт из каждого пакета анализируется службой.

Тип: глобальный. Режим: ids. *По умолчанию: 1514 для режима afpacket; 65535 для режима nfq*

daq nfq queue len <VAL>

Данная команда задает размер очереди пакетов.

Тип: глобальный. Режим: ips. *По умолчанию: 1024*

Другие команды задают различные опции каждого из режимов выборки.

Для режима выборки nfq/nfq6 можно также указать правила отбора трафика. Если их не указать, то будет отбираться весь трафик.

Для этого используйте команды **ipsrule <perlmit|deny>** и **ipsrule6 <perlmit|deny>**.

Их формат аналогичен многим другим командами системы для правил отбора трафика, например permit в ip access-list.

В режиме выборки afpacket интерфейсы, заданные командами iface или iface-pair, автоматически устанавливаются в неразборчивый режим (promisc-режим), т.е. интерфейс будет принимать все пакеты, даже не предназначенные системе (с неизвестными MAC-адресами).

Кроме того к подсистеме отбора трафика относится команда: **packets bpf**.

Она позволяет создать Barkley-фильтр для отбора трафика. Он выполнится до правил ipsrule/ipsrule6 на входе трафика в интерфейс и после правил ipsrule на выходе трафика из интерфейса.

46.4 Настройки подсистемы декодирования

После того, как пакет захвачен системой, он направляется в подсистему декодирования. Этапы декодирования:

- декодирование канального уровня поддерживает следующие протоколы: ETH, PPP, MPLS, ARP, GTP и др.;
- декодирование сетевого и транспортного уровня: IPv4, IPv6, ICMPv4, ICMPv6, GRE, ESP, IGMP, TCP, UDP и др.

На каждом этапе декодирования происходит проверка соответствующего протокола на предмет ошибок/аномалий и ,если они выявлены, то пакет может быть отброшен и будет выдано соответствующее предупреждение в журнал - сработает встроенное правило (с gid >=100).

Для настройки подсистемы декодирования используйте следующие команды:

- команды раздела **decoder**: decoder esp, decoder mpls и др.
- команды раздела **packets**: packets daq-tracking, packets vlan-tracking - настройка процесса учета IP фрагментов и соединений
- команды раздела **network** : например network cksum, network layers, network max-ip-layers и др.
- команды раздела **threads**: threads total - задает число потоков выборки и анализа

46.5 Настройки подсистемы препроцессинга

После того, как пакет декодирован и не были обнаружены никакие ошибки или аномалии соответствующих ему протоколов, начинается процесс формирования оригинального пакета (т.к. пакеты могут приходить не по порядку, быть фрагментированными и сегментированными).

Для настройки подсистемы препроцессинга используйте следующие команды:

- команды раздела **hosts**: hosts tracker - можно задать политику IP-дефрагментации и TCP-пересборки пакетов для указанной сети
- команды раздела **inspector** : inspector stream, arp, smtp, http и др.

Настройка инспектора позволяет указать особенности обработки трафика протокола, за который отвечает данный инспектор. Например команды inspector http - описывают различные особенности нормализации HTTP-сообщения, установки различных лимитов, условий выдачи предупреждений в результате обнаруженных аномалий протокола и др.

Помимо инспекторов протоколов существуют два особенных типа инспекторов: wizard и binder. Остановимся на них подробнее.

46.5.1 Инспектор binder.

Инспектор binder используется для привязки сетевого соединения к инспектору протокола, например трафик SSH или весь трафик на порту 22 будет обслуживаться инспектором SSH. Привязка осуществляется посредством создания правила, в котором можно указать такие условия проверки, как порты, IP-адреса, протокол, а также **имя сервиса**. Если указано несколько условий, то применяется AND-логика - все эти условия должны быть удовлетворены для срабатывания правила. Под **именем сервиса** понимается имя, которое связывает правила инспектора wizard и инспектора binder (см.ниже описание инспектора wizard).

В конце инспектора binder всегда указано правило перехода на инспектор wizard, таким образом, если в правилах инспектора binder не удалось определить, какой именно инспектор применить к соединению, то идет просмотр правил инспектора wizard, в котором определяется **имя сервиса** для текущего соединения. В инспекторе wizard посредством независящих от портов правил идентификации определяется сервис для данного соединения. После этого осуществляется повторный проход по правилам инспектора binder с уже известной информацией о сервисе для соединения. Более подробно об инспекторе wizard см. в следующем подразделе.

Инспектор binder конфигурируется командой match внутри каждого из инспекторов. Кроме того, автоматически при создании инспектора будет добавлено правило в инспектор binder в виде, например, для инспектора ssh:

```
{ when = { service = ”ssh” }, use = { type = ”ssh” } }
```

Если затем добавить команды match внутри инспектора ssh для отбора трафика, то таблица правил инспектора binder может стать следующей:

```
adm@DionisNX(config—service—idsm) do show service idsm binder
Binder:
{ when = { ports = ”22” }, use = { type = ”ssh” } }
{ when = { ports = ”123”, service = ”ssh” }, use = { type = ”ssh” } }
{ when = { service = ”ssh” }, use = { type = ”ssh” } }
{ use = { type = ”wizard” } }
```

Рассмотрим эту таблицу подробнее:

- первое правило таблицы соответствует команде **match port 22** в инспекторе ssh, т.е. трафик на порту 22 направляется в инспектор ssh;
- второе правило таблицы соответствует команде **match port 123 service ssh** в инспекторе ssh, т.е. трафик на порту 123 сервиса ssh (определенного инспектором wizard как сервис ssh) направляется в инспектор ssh;
- третье правило таблицы соответствует автоматической команде, добавляемой в таблицу при выполнении команды **inspector ssh**, т.е. при создании инспектора ssh; автоматические команды будут **всегда** последними в таблице и могут сработать, если предыдущие правила не подошли для данного пакета или соединения;

- последнее правило предписывает обратиться к инспектору wizard для идентификации сервиса и повторного прохождения по таблице правил binder.

Посмотреть таблицу правил инспектора binder можно командой **show service idsm binder** при включенной службе idsm.

46.5.2 Инспектор wizard.

Инспектор wizard используется для независящий от портов идентификации сервиса для трафика. После успешной идентификации сервиса служба будет знать имя сервиса для текущего сетевого пакета или соединения и обратиться к правилам инспектора binder для определения, какой именно инспектор протокола использовать.

Для задания правила в инспекторе wizard используются две команды:

- spell : задает символьный образец для поиска в данных пакета, а также другие условия: направление пакета (к клиенту или к серверу) , транспортный протокол (tcp или udp), указание на инициатора обмена сообщениями
- hex : аналогично spell, но образец является не символьным, а последовательностью hex-байтов в формате AA:[BB:].

В системе по-умолчанию присутствует встроенная таблица правил инспектора wizard для определения некоторых сервисов. Посмотреть таблицу правил инспектора wizard можно командой **show service idsm wizard** при включенной службе idsm.

Пользователь в конфигурации инспектора wizard может добавить свои правила. Если правило для данного сервиса уже существует в системной таблице wizard, то оно будет заменено пользовательским, если оно не существует, то будет добавлено в системную таблицу wizard.

Для примера рассмотрим конфигурацию:

```
inspector wizard
  hex ftp udp toclient 12:34:56:ab:cd:ef
  spell dnp3 tcp toserver THIS_IS_DNP3 toclient THAT_IS_DNP3 server—first
  spell ssh tcp toserver THIS_IS_SSH
```

Эти команды будут в таблице wizard в следующем виде

```
Spells:
{ proto = ”tcp”, client_first = false, service = ”dnp3”, to_server = { ”THIS_IS_DNP3” }, to_client = {
        ”THAT_IS_DNP3” } }
{ service = ”ssh”, to_server = { ”THIS_IS_SSH” }, proto = ”tcp” }
Hexes:
{ service = ”ftp”, to_client = { ”|12 34 56 ab cd ef|” }, proto = ”udp” }
```

Эти команды определяют:

- 1я команда: если от сервера к клиенту передается строка по UDP-протоколу в виде последовательности байтов 0x12,0x34,0x56,0xab,0xcd,0xef, то данное соединение будет принято за трафик сервиса ftp;

- 2я команда: если от сервера к клиенту передается строка THAT_IS_DNP3 по TCP-протоколу, от клиента к серверу передается строка THIS_IS_DNP3 по TCP-протоколу, причем инициатором соединения является сервер, то данное соединение будет принято за трафик сервиса dnp3;

- 3я команда: если от клиента к серверу передается строка THIS_IS_SSH по TCP-протоколу,то данное соединение будет принято за трафик сервиса ssh;

46.6 Настройка подсистемы обнаружения

После прохода пакета через подсистему препроцессинга, через выбранный для него инспектор, в случае если для пакета не выявлено нарушений, то он далее попадает в подсистему обнаружения, которая основана на правилах обнаружения и других смежных настройках.

Для настройки подсистемы обнаружения используйте следующие команды:

- некоторые команды раздела **alerts**: alerts order, alerts detection-filter-memcap, alerts stateful
- команда **class**: задает класс атаки для использования в пользовательских правилах

- команды раздела **detection**

- команды **ipvar/portvar**: для задания именованных переменных для IP-адресов и портов, эти переменные затем могут быть использованы в заголовке правила

- команда **limit prepackets**
- команда **norule**: отключение правила по gid/sid
- команды раздела **search**: search method, search no-any-any и др.; для настройки механизма поиска по образцу

- команды раздела **ruleset**: настройка системных или пользовательских правил
- команды разделов active, rewrite, react, reject: подсистема активного реагирования

Рассмотрим далее основные сущности подсистемы: правила и переменные.

46.6.1 Переменные

В службе есть понятие именованных переменных, которые могут быть двух типов:

- ipvar - именованная переменная, хранящая IP-адреса;

- portvar - именованная переменная, хранящая номера портов.

Эти переменные нужны для удобства и используются в правилах для указания IP-адресов или портов.

Существуют предопределенные наборы системных переменных обоих типов, имеющие некоторые стандартные значения, например:

```
FTP_PORTS =  21 2100 3535
```

Для просмотра текущего набора заданных и системных переменных используйте команды **show service idsm ipvars** и **show service idsm portvars**.

Для создания переменной выполните команду **portvar** или **ipvar**. При задании переменной можно использовать знак "!" для отрицания следующего за ним значения, например **portvar VAR !22** означает переменная VAR состоящая из всех портов, кроме 22 порта.

Рассмотрим формат переменных на примере ipvar. Для portvar формат команды аналогичен.

[PRIO] ipvar <NAME> {IPVAR|[!]IP[/MSK]|any,8}

Параметры команды:

- PRIO - номер ,под которым следует вставить переменную в списке переменных;
- NAME - имя переменной;
- IPVAR - другая переменная типа ipvar;
- IP - адрес IP;
- MSK - маска подсети;
- any - любой IP-адрес;
- ! - отрицание следующего параметра.

Имя переменной any - зарезервировано и не может быть переопределено. Переменная ipvar any задает все IP-адреса.

Например: ipvar net1 10.0.0.0/24 !10.0.0.1 - задает переменную net1, хранящую набор IP-адресов из сети 10.0.0.0/24 кроме адреса 10.0.0.1

Для изменения ранее заданной переменной в команде изменения следует использовать приоритет (PRIO) , равный приоритету изменяемой переменной.

46.6.2 Правила обнаружения

Основа подсистемы обнаружения - это правила обнаружения.

Правило обнаружения (далее - правило) состоит из:

- **заголовок**: какие именно сетевые пакеты, идентифицируемые по двум тройкам протокол-адрес-порт источника и назначения, следует отслеживать;
- **тело**: что именно в данных пакетах следует проверять;

- **действие**: что именно следует делать с пакетом в случае выполнения условий, определяемых заголовком и телом правила.

Пример правила обнаружения: alert tcp $EXTERNAL_NET any -> $HOME_NET 6000 (msg:“X11 xopen”; flow:established; content:“|00 0B 00 00 00 00 00 00 00 00|”,fast_pattern,nocase; sid:1225;)

В данном правиле заголовок это tcp $EXTERNAL_NET any -> $HOME_NET 6000, тело правила - это опции правила, указанные в скобках, действие правила - alert, означает предупреждать (через лог), при срабатывании правила.

Правила идентифицируются в системе следующим триплетом чисел: gid:sid:rev, где gid - задает номер подсистемы, которой принадлежит правило, sid - номер правила среди правил данной подсистемы, rev - ревизия правила (обычно не используется).

Если в правиле не указан GID, он по умолчанию равен 1. Это означает,что предупреждения, выдаваемые правилом с gid=1 принадлежат к подсистеме правил, без точного указания, к каким именно правилам.

Правила службы могут быть трех типов:

- **встроенные**: описывают события подсистем службы (например декодирование пакета, различные инспекторы и т.д); их можно посмотреть по команде show service idsm rules all builtin ; имеют gid >= 100.

- **системные**: относятся к подсистеме обнаружения и уже включены в дистрибутив вместе с службой idsm; их можно посмотреть по команде show service idsm rules detection/detection-loaded NAM; имеют gid = 1.

- **пользовательские**: относятся к подсистеме обнаружения и уже включены в дистрибутив вместе с службой idsm; их можно посмотреть по команде show service idsm rules detection; имеют gid = 8000.

Настраивать можно только системные и пользовательские правила. Встроенные правила всегда активны.

Существует понятие группы правил. Группа правил - это набор правил, сгрупированных по какому-то смысловому признаку, например группа правила icmp, группа правил windows. Это сделано для удобства.

46.6.3 Настройка системных правил обнаружения.

Для включения группы системных правил выполните команду **ruleset** с указанием имени группы правил.

После этого служба включит данную группу правил в свою конфигурацию. Причем группа будет включена в том виде, в котором она присутствует в дистрибутиве, т.е. в ней могут быть какие-то правила отключены, а какие-то включены. Это же верно и после обновления правил.

Если войти в группу правил, то открывается возможность использования следующих команд:

on <all| <GID> <SID> >

По этой команде можно включить все или указанное правило данной группы. Правила идентифицируется по номерам GID/SID, которые можно узнать по команде show service idsm rules.

off <all|<GID> <SID»

по этой команде можно выключить все или указанное правило данной группы. Правила идентифицируется по номерам GID/SID, которые можно узнать по команде show service idsm rules.

action <GID> <SID> <TYPE>

Эта команда меняет действие встроенного правила под номером GID/SID на действие, определяемое параметром TYPE:

- alert : записать в журнал пакет и предупреждение;
- log : записать в журнал пакет;
- pass : разрешить прохождение пакета;
- drop : записать пакет в журнал и одновременно запретить его;
- reset : записать пакет в журнал и одновременно запретить его; дополнительно послать “TCP reset” (для TCP) или “ICMP port unreachable” (для UDP);
- block : то же самое, что drop, но блокировать не только данный пакет, но и все последующие пакеты потока.

Перечислим группы системных правил обнаружения, входящие в дистрибутив:

- ruleset app-detect - правила обнаруживающие определенные приложения, работающие в сети (например, Gizmo - аналог Skype). При отсутствии политики, запрещающей использования ПО из списка, указанного в данном наборе правил, можно отключать весь набор.

- ruleset browser-chrome, ruleset browser-firefox, ruleset browser-ie, ruleset browser-other, ruleset browser-plugins, ruleset browser-webkit - реагируют на уязвимости соответствующих браузеров - переполнения адресной строки браузера, попытки отправить отчет об ошибках и т.д. Группа browser-other - в основном содержит ошибки связанные с браузером Opera, а webkit - Safari. Если в организации отсутствуют какие-то из этих браузеров, то соответствующие наборы правил можно закомментировать.

- ruleset content-replace - возможность заменять данные внутри пакета, попадающего под правило

- ruleset deleted – содержит устаревшие правила.

- ruleset exploit-kit - сигнатуры известных эксплоитов. Предупреждения генерируются в случае, если данными эксплоитами пытались воспользоваться, хотя сами эксплоиты к этому времени могли быть закрыты патчами или уязвимого ПО вообще может быть не установлено в системе.

- ruleset file-executable, ruleset file-flash, ruleset file-identify, ruleset file-image, ruleset file-java, ruleset file-multimedia, ruleset file-office, ruleset file-other, ruleset file-pdf - предупреждения о наличии в пересылаемых по сети файлах,содержащих потенциально “опасные” данные (например, макросы в офисных документах).

- ruleset indicator-compromise - фиксирует “опасные” запросы к удаленным ресурсам (попытка обращения к конфигурационным файлам веб-сайтов, вызывать cmd на удаленном узле и т.п.).
- ruleset indicator-obfuscation - обнаруживает в трафике обфусцированный JS код. Как правило его используют для защиты информации от автоматического копирования, но иногда с помощью него могут попытаться скрыть опасный код. Если такого трафика генерируется мало, то в принципе можно попытаться деобфусцировать полученные данные.
- ruleset indicator-scan - обнаруживают попытки сканирования сети. Содержат сигнатуры некоторых конкретных сетевых сканеров.
- ruleset indicator-shellcode - обнаруживают в пересылаемых по сети пакетах шеллкод, что вероятнее всего легитимный трафик. По умолчанию - выключены, при включении значительно снижают производительность сервера IDSM.
- ruleset malware-backdoor - реагирует на запрос соединения с удаленным компьютером, инициализированный вредоносным ПО.
- ruleset malware-cnc - обнаруживает центр управления множеством компьютеров. Но теперь уже реагирует на вредоносное ПО, ожидающее подключения новых ботов, регистрирующее их в своей базе, следящее за их состоянием и выдающее им команды, выбранные владельцем ботнета из списка всех возможных команд для бота.
- ruleset malware-other - фиксирует работу вредоносного ПО, не подпадающего под описание предыдущих двух пунктов.
- ruleset malware-tools - обнаруживают работу хакерского инструментария на узле сети.
- ruleset netbios - обнаруживают деятельность некоторых сетевых червей, атакующих машины под управлением Windows. Часто выдают ложные срабатывания (особенно если дело касается правил управления общим доступом и правил, предупреждающих о доступе к SMBи NetBIOS). Если сенсор Snort’а наблюдает только за интернет-трафиком, или NetBIOSтрафик не выходит за пределы сети (и не входит в ее пределы), то рекомендуется отключить этот набор правил.
- ruleset os-linux - содержит правила, которые ищут уязвимости в операционных системах на базе Linux. Не для браузеров или любое другое программное обеспечение на нем, а против самой ОС.
- ruleset os-mobile - содержит правила, которые ищут уязвимости в мобильных операционных системах.
- ruleset os-solaris - содержит правила, которые ищут уязвимости в операционных системах на базе Solaris.
- ruleset os-windows - содержит правила, которые ищут уязвимости в операционных системах на базе Windows.
- ruleset os-other - содержит правила, которые ищут уязвимости в операционных системах которые не описаны в приведенном выше списке.
- ruleset policy-multimedia, ruleset policy-other, ruleset policy-social, ruleset policy-spam - реагируют на активность, которая может быть запрещена политикой безопасности в некоторых организациях (например, анонимный вход по ftp, запуск java-апплетов плееров для проигрывания видео, доступ к gmail и т.п.)..

- ruleset protocol-dns - реагируют на атаки направленные на dns-сервера.
- ruleset protocol-finger - - этот набор содержит правила, касающиеся известных атак на службу finger (которая по умолчанию запускается во многих unix-подобных ОС). Если таких узлов в сети нет, то эти правила можно отключить.
- ruleset protocol-ftp - срабатывают при обнаружении атак на ftp-сервера.
- ruleset protocol-icmp - фиксируют попытки пропинговать узлы сети с использованием некоторого хакерского инструментария, так же учитывает любые пинги, которые могут в огромном количестве генерироваться вполне легитимным ПО.
- ruleset protocol-imap - детектируют попытки атак на IMAP-сервера.
- ruleset protocol-nntp - содержат сигнатуры атак на службы времени.
- ruleset protocol-other - другие протоколы
- ruleset detection protocol-pop регистрируют потенциальные атаки на соответствующие почтовые службы на осонове pop2-протокола.
- ruleset protocol-rpc - регистрируют атаки на службы удаленного вызова процедур. Если такой трафик не поступает из внешней сети или во внешнюю сеть, то набор правил рекомендуется отключить.
- ruleset protocol-scada - фиксируют атаки по протоколу ModBus
- ruleset protocol-services - фиксируют команды удаленного доступа к системе (rlogin, rsh, rexec). Если они используются в Вашей сети легитимно, то этот набор правил стоит отключить.
- ruleset protocol-snmp - фиксирует активность SNMP протокола (удаленное управление сетевым оборудованием)
- ruleset protocol-telnet - предупреждают об опасном трафике, пересылаемом во время telnet-сессии.
- ruleset protocol-tftp - сигнатуры потенциальных атак на службы tftp.
- ruleset protocol-voip - регистрируют ошибки и потенциальные атаки на средства голосового общения по сети.
- ruleset pua-adware - регистрируют ошибки и потенциальные атаки с использованием «Потенциально нежелательных приложений»(pua) , рекламные или шпионское ПО.
- ruleset pua-p2p - детектирует активность пиринговых программ, нарушающих законодательство (в частности, авторское право).
- ruleset pua-toolbars - правила касаются тулбаров (панелей инструментов, встраиваемых в браузер), нарушающих права (например, отправляющих статистику запросов на удаленный сервер без спроса пользователя).
- ruleset pua-other - регистрируют ошибки и потенциальные атаки с использованием pua-приложений (см.выше в pua-adware), которые не описаны в приведенном выше списке.

- ruleset server-apache - обнаруживают атаки на веб-сервера Apache.
- ruleset server-iis - обнаруживают атаки на веб-сервера IIS.
- ruleset server-mail - регистрируют попытки известных атак на почтовые сервера организации.
- ruleset server-mssql - регистрируют попытки атак на сервера баз данных под управлением Microsoft SQL Server.
- ruleset server-mysql - регистрируют попытки атак на сервера баз данных под управлением MySQL.
- ruleset server-oracle - регистрируют известные атаки на сервера баз данных под управлением Oracle.
- ruleset server-other - обнаруживают уязвимости или атаки против серверов, которые не описаны в приведенном выше списке.
- ruleset server-samba - регистрируют основные признаки уязвимости или атаки на серверы Samba.
- ruleset server-webapp - регистрируют основные признаки уязвимостей или атак на веб-приложения на серверах.
- ruleset sql - Эта категория содержит правила, которые обнаруживают инъекции SQL или наличие других уязвимостей против SQL, как серверы.
- ruleset x11 - фиксируют потенциальные атаки использующие уязвимости графического интерфейса UNIX-подобных ОС.

Для удобства включения и отключения всех групп правил существует команда:

ruleset detection-all

46.6.4 Настройка пользовательских правил обнаружения.

Пользовательские правила, как и системные, состоят из групп правил. Для создания пользовательской группы правил используйте команду **ruleset manual**. Внутри группы правил можно создавать отдельные правила.

Таким образом, в дополнение к командам включения и выключения правил (см. Настройка системных правил обнаружения), добавляется команда создания правила.

[NUM] rule <NAME>

По этой команде можно создать пользовательское правило под именем NAME, войти в режим его конфигурации и поместить его под номером NUM.

По умолчанию создается такое правило:

```
alert icmp any any —> any any ( gid:8000; sid:1; )
```

46.6.5 Настройка пользовательского правила

Правило состоит из:

- заголовка : определяет, какой трафик подпадает под анализ данным правилом; заголовок состоит из следующих параметров:
  - action : действие (что будет сделано с пакетом, если будут выполнены условия, описанные опциями данного правила).
  - proto : тип протокола, может быть tcp,udp,ip,icmp;
  - src/dst : IP-адреса источника и назначения пакета;
  - sport/dport : порты источника/назначения пакета;
  - dir : направление трафика: от источника к назначению, либо в обе стороны;
- опций : под этим термином понимаются различные проверки, выполняемые над данными, содержащимися в пакете:
  - опции правила для описания правила;
  - опции правила для проверки содержимого пакета;
  - остальные опции правила: для проверки заголовка пакета и др.

Введем понятия,необходимые для понимания правил:

- **курсор обнаружения** - это смещение в пакете после последнего найденного содержимого или перемещение туда,куда указывает команда byte-jump/byte-test (см.ниже); т.е. это указатель на данные в пакете, где будет продолжено обнаружение в соответствии с опциями правила, которые должны следовать после команды установления курсора обнаружения;
- для некоторых опций правила важен порядок их нахождения в списке опций правила, т.е. **приоритет**; опции с приоритетом анализируются в этих правилах в порядке, определяемом приоритетом;
- в некоторых командах используется параметр LOGIC, задающий оператор сравнения; этот оператор может принимать значения: more,less,not,equal,more-equal,less-equal (больше,меньше,не равно,равно,больше либо равно,меньше либо равно).

46.6.6 Управление пользовательским правилом

on

Эта команда включает текущее пользовательское правило (по умолчанию правило и так включено).

off

Эта команда выключает текущее пользовательское правило.

46.6.7 Пользовательское правило: заголовок.

action <TYPE>

Эта команда задает действие правила:

- alert : записать в журнал пакет и предупреждение;
- log : записать в журнал пакет;
- pass : разрешить прохождение пакета;
- drop : записать пакет в журнал и одновременно запретить данный пакет;
- reject : записать пакет в журнал и одновременно запретить данный пакет; дополнительно послать “TCP reset” (для TCP) или “ICMP port unreachable” (для UDP);
- sdrop : запретить данный пакет.

По умолчанию: alert

proto <tcp|udp|ip|icmp|http>

Эта команда задает тип протокола. Параметр может быть следующим: tcp,udp,ip,icmp,http.

По умолчанию: tcp

src <{[!]IP[/MSK]|IPVAR,4}>

Эта команда задает адреса источника пакета. Только те пакеты, IP-адрес источника которых принадлежит указанным адресам, будут обрабатываться данным правилом.

По умолчанию: any

dst <{[!]IP[/MSK]|IPVAR,4}>

Эта команда задает адреса назначения пакета. Только те пакеты, IP-адрес назначения которых принадлежит указанным адресам, будут обрабатываться данным правилом.

По умолчанию: any

sport <{[!]PORT1[:PORT2],4}>

Эта команда задает порт источника пакета. Только те пакеты, порт источника которых принадлежит указанным портам, будут обрабатываться данным правилом.

По умолчанию: any

dport <{[!]PORT1[:PORT2],4}>

Эта команда задает порт назначения пакета. Только те пакеты, порт назначения которых принадлежит указанным портам, будут обрабатываться данным правилом.

По умолчанию: any

bidirectional

Эта команда задает двунаправленный трафик: адреса и порты,указанные командами src/sport и dst/dport, могут быть адресами и портами источника и адресами и портами назначения.

46.6.8 Пользовательское правило: базовые опции.

general message <STRING>

Эта команда задает описание правила в виде текстовой строки.

general reference <bugtraq|cve|nessus|arachnids|mcafee|osvdb|URL>

Задает ID атаки, которую отслеживает правило, в той или иной базе данных уязвимостей. Также возможно указание непосредственно URL. Возможные базы данных:

- bugtraq;
- cve;
- nessus;
- arachnids;
- mcafee;
- osvdb.

general gid <VAL>

Эта команда задает идентификатор группы, к которой принадлежит правило.

Параметр VAL принимает значения от 1 до 128.

Реальное значение gid, идущее в правило: VAL+8000. Это необходимо учитывать при использовании идентификатора в других командах: нужно прибавлять 8000 к значению gid, указанному в данной команде.

general sid <VAL>

Эта команда задает идентификатор правила.

Реальное значение sid, идущее в правило: VAL+1000000. Это необходимо учитывать при использовании идентификатора в других командах: нужно прибавлять 1000000 к значению sid, указанному в данной команде.

general class <CLASS>

Эта команда задает класс правила. Параметр может принимать только те значения, которые выдаются по автодополнению команды.

46.6.9 Пользовательское правило: опции проверки содержимого.

В некоторых опциях данного типа важен их приоритет, т.к. опции с приоритетом в правиле анализируются последовательно. Например,если опция стоит под номером 1, она будет анализироваться первой и, возможно, установит курсор обнаружения в новое значение.

Сначала будут рассмотрены команды, для которых можно задавать приоритет (PRIO). После них будут описаны команды без приоритета.

[PRIO] payload content <PAT> [PARAMS]

Эта команда позволяет правилу искать заданный образец в содержимом (данных) пакета.

Формат образца PAT: текстовая строка и/или бинарные данные. Пример использования:

```
payload content abc*12fef*def
```

Образец представляет собой тестовую строку «abc», за которой следует набор байтов 0x12,0xfe,0xf0 и далее текстовая строка «def». Особенности формата образца:

- последовательность символов является бинарными данными,если она состоит из символов 0..9 ,a..f, заключенных между 2мя символами «*»;
- все остальные последовательности символов являются текстом;
- если число цифр бинарных данных (не включая ограничивающие символы «*») нечетное, то результирующая hex-последовательность дополняется справа нулем (как например 0xf0 в вышеприведенном примере).

Образец PAT - единственный обязательный параметр команды payload content. Остальные параметры PARAMS - необязательные и нужны для изменения особенностей поиска по указанному образцу. Рассмотрим далее возможные параметры PARAMS команды payload content вместе с примерами.

46.6.9.0.1 nocase Это значение параметра задает нечувствительный к регистру поиск; «payload content abc nocase» будет искать, например, такие строки как abc, ABC, abC и др.

46.6.9.0.2 depth <VAL|BEVAL> Это значение параметра задает размер данных, начиная с начала данных пакета, в которых следует вести поиск по образцу.

Дополнительные параметры:

- VAL - число от 1 до 65535;
- BEVAL - имя заданного в данном правиле byte-extract-параметра.

По умолчанию: все данные пакета.

46.6.9.0.3 offset <VAL|BEVAL> Это значение параметра задает смещение от начала данных пакета, которое будет применено при определении начальной позиции в данных, с которой начнется поиск по образцу.

Дополнительные параметры:

- VAL - число от -65535 до 65535;
- BEVAL - имя заданного в данном правиле byte-extract-параметра

По умолчанию: offset 0.

46.6.9.0.4 match-offset min <VAL1|BEVAL1> max <VAL2|BEVAL2> Аргументы для max и min:

- VAL - число от -65535 до 65535;
- BEVAL - имя заданного в данном правиле byte-extract параметра.

Параметр min задает количество байт, которые нужно пропустить после курсора обнаружения, для начала поиска образца. Параметр max задает максимальное количество байт после курсора обнаружения, среди которых искать образец.

Т.е. данной командой мы задаем интервал поиска образца, относительно смещения последнего найденного образца. Поэтому данную команду обычно следует использовать после другой команды payload content.

Рассмотрим пример:

```
1 payload content ABC
2 payload content DEF match—offset min 10 max 20
```

Данные команды ищут по образцу /ABC.{10}DEF/, причем DEF ищется в первых 20 байтах после ABC,но не далее (т.к. курсор обнаружения после 1й команды payload content уже сдвинут на образец ABC, если он конечно найден).

46.6.9.0.5 fast-pattern [<OFF>,<LEN>] Этот параметр предписывает активировать быстрый поиск по образцу - позволяет ускорить поиск образца. Дополнительные параметры:

- OFF,LEN - задает смещение и длину части образца для поиска. Полезно, если образец слишком длинный, чтобы сохранить память в механизме быстрого поиска.

[PRIO] payload isdataat [!]<VAL> [relat] [rawbytes]

Этот параметр предписывает проверять, есть ли (нет ли,если указан знак »!») данные по указанному смещению VAL от начала данных пакета, или от конца последнего найденного образца (если указан relat).

Например, по команда:

```
payload content PASS isdataat 50 relat
```

будет проверено, есть ли еще какие-то данные на протяжение 50 байт после слова PASS. Если данные есть, выдается предупреждение.

Параметр rawbytes - искать в ненормализованных препроцессорами данных.

[PRIO] payload regex [no]<REGEX>[newline|nocase]

Этот параметр предписывает создать регулярное выражение для поиска по образцу. Дополнительные параметры:

- no - поиск инвертируется;
- newline - в метасмивол ».» попадают также символы переноса строк;
- nocase - нечувствительный к регистру поиск.

Крайне желательно,чтобы перед данной командой была команда payload content, которая отсеивала бы неподходящие варианты перед передачей данных на анализ с помощью регулярного выражения (что более медленно).

[PRIO] payload base64-decode [size <VAL>] [offset <VAL> [relative]

Этот параметр предписывает разрешить декодировать base64-данные и осуществлять в них поиск по образцу.

Дополнительные параметры:

- size - размер base64-данных для декодирования;
- offset - смещение начала base64-данных относительно начала данных пакета (если не задан relative);
- relative - параметр offset задает смещение base64-данных относительно курсора обнаружения.

[PRIO] payload cursor <TYPE>

Этот параметр устанавливает курсор обнаружения. Варианты установки курсора определяются параметром TYPE:

- file - данные передаваемого файла;
- packet - нормализованые данные пакета;
- rawpacket - ненормализованые данные пакета;
- sip-header - заголовок SIP;
- sip-body - данные сообщения SIP;
- dce-stub-data - stub данные DCE/RPC;
- modbus-data - поле данных Modbus;
- dnp3-data - начало DNP3 пересобранного (инспектор dnp3) фрагмента уровня приложения;
- http-client-body - HTTP-запрос клиента (сообщения PUT, POST)
- http-cookie - нормализованые Cookie и Set-Cookie заголовки HTTP-запроса клиента и HTTP-ответа сервера
- http-raw-cookie - ненормализованые Cookie и Set-Cookie заголовки HTTP-запроса клиента и HTTP-ответа сервера
- http-header - нормализованые HTTP-заголовки
- http-raw-header - ненормализованые HTTP-заголовки
- http-method - поле Method HTTP-запроса

- http-uri - нормализованные данные поля URI HTTP-запроса
- http-raw-uri - ненормализованный данные поля URI HTTP-запроса
- http-stat-code - код поля Status HTTP-ответа сервера
- http-stat-msg - сообщение поля Status HTTP-ответа сервера
- http-true-ip - данные заголовков X-Forwarded-For или True-Client-IP (т.е. настоящий IP-адрес клиента)
- http-trailer - нормализованные данные HTTP-трейлера (это данные, которые могут быть после основного тела HTTP сообщения)
- http-raw-trailer - ненормализованные данные HTTP-трейлера (это данные, которые могут быть после основного тела HTTP сообщения)
- http-raw-body - ненормализованное тело сообщения HTTP (запрос или ответ)
- http-version - 1я строка HTTP сообщения (там, где указана версия протокола HTTP)
- http-raw-request - 1я строка HTTP сообщения запроса
- http-raw-status - 1я строка HTTP сообщения ответа (там, где указан статус HTTP ответа)
- http-param - данные указанного HTTP параметра
- http2-decoded-header - декдоированный HTTP/2 заголовок
- http2-frame-header - декдоированный HTTP/2 заголовок фрейма

Будучи установленным, курсор будет действовать на все опции правила, заданные после него.

Команда устанавливает положение курсора обнаружения. Курсо

[PRIO] payload asn1 [bisttring-overflow][double-overflow][OLEN][absolute-offset <VAL>|relative-offset <VAL>]

Эта команда включает декодирование ASN1 и задает обнаружение атак, связанных с ASN1-кодированием.

Параметры:

- double-overflow - обнаружение двойного ASCII-кодирования с превышением размера буфера;
- bitstring-overflow - обнаружение атаки, связанной с неверным кодированием битовых строк;
- OLEN - максимальный размер типа ASN1;
- absolute-offset - абсолютное смещение от начала пакета, где находятся ASN1-данные;
- relative-offset - смещение относительно курсора обнаружения.

[PRIO] payload byte test <[no] OP> <VAL> <OFF> <NUM> [relative] [dce] [END] [string <hex|dec|oct>]

Цели этой команды:

- сравнение байтового поля из пакета с указанным значением;
- конвертация байтовых строк в байтовое поле и сравнение его с указанным значением.

Параметры команды:

- OP - оператор: more/less/equal/and/or (больше/меньше/равно/и/или); модификатор no - делает отрицание оператора, т.е.: euqal no - проверка на «не равно»;
- VAL - значение, с которым сравнивается байтовое поле из пакета;
- OFF - смещение в данных пакета, определяющее начало байтового поля;
- NUM - число байт байтового поля (от 1 до 10);
- relative - смещение относительно курсора обнаружения;
- dce - DCE/RPC-препроцессор будет определять порядок байтового поля пакета;
- END - порядок байт: little-endian или big-endian;
- string : задает тип байтовой строки в пакете, которая конвертируется в байтовую последовательность: hex(16-ричная),dec(10-тичная),oct(8-ричная)

[PRIO] payload byte jump <OFF> <NUM> [relative] [string <hex|dec|oct>] [END] [dce] [from-beginning] [align] [post-offset <POFF>] [mul <MUL>]

Эта команда позволяет перемещать курсор обнаружения на значение ,считанное из пакета по заданному смещению. Обычно это удобно для протоколов, в которых по определенным смещениям указаны длины различных полей пакета.

Параметры:

- OFF - смещение в данных пакета, определяющее начало байтового поля;
- NUM - число байт байтового поля (от 1 до 10);
- relative - смещение относительно курсора обнаружения;
- string : задает тип байтовой строки в пакете, которая конвертируется в байтовую последовательность: hex(16-ричная),dec(10-тичная),oct(8-ричная);
- dce - DCE/RPC-препроцессор будет определять порядок байтового поля пакета;
- END - порядок байт: little-endian или big-endian;
- from-begginig - следует перемещать курсор обнаружения, начиная с начала пакета, а не с текущей позиции курсора;
- align - следует выравнивать число конвертированных байт на 32-битную границу;
- POFF - пропуск вперед или назад после отработки остальных опций команды;
- MUL - число, на которое умножается число взятых/сконвертированных байт.

[PRIO] payload byte extract <OFF> <NUM> <VAR> [relative] [string <hex|dec|oct>] [END] [dce] [from-beginning] [align] [post-offset <POFF>] [mul <MUL>]

Эта команда позволяет прочитать определенное число байт из пакета и присвоить полученное значение переменной, которую можно далее использовать в правиле.

Параметры:

- OFF - смещение в данных пакета, определяющее позицию начала считывания байт;
- NUM - число байт для считывания (от 1 до 10);
- VAR - имя переменной, которой следует присвоить считанное значение;
- relative - смещение относительно курсора обнаружения;
- string : задает тип байтовой строки в пакете, которая конвертируется в байтовую последовательность: hex(16-ричная),dec(10-тичная),oct(8-ричная);
- dce - DCE/RPC-препроцессор будет определять порядок байтового поля пакета;
- END - порядок байт: little-endian или big-endian;
- from-begginig - передвигать курсор обнаружения относительно начала пакета, а не с текущей позиции курсора
- align - выравнивать число конвертированных байт на 32-битную границу;
- POFF - пропуск вперед или назад после отработки остальных опций команды;
- MUL - число, на которое умножается число взятых/сконвертированных байт.

[PRIO] payload byte math <MATH_OPER> <VAR> <VAL> <OFF> <NUM> [relative] [dce] [ENDIAN] [MSK]

Цели этой команды:

- сравнение байтового поля из пакета с указанным значением;
- конвертация байтовых строк в байтовое поле и сравнение его с указанным значением.

Параметры команды:

- MATH_OPER - математическая операция над извлекаемым значением и значением VAL;
- VAR - имя переменной, которой следует присвоить результат операции;
- VAL - значение, которое используется в математической операции вместе с извелченным значением;
- OFF - смещение в данных пакета, определяющее начало байтового поля;
- NUM - число байт байтового поля (от 1 до 10);
- relative - смещение относительно курсора обнаружения;
- dce - DCE/RPC-препроцессор будет определять порядок байтового поля пакета;
- END - порядок байт: little-endian или big-endian;
- MSK - значение для операции AND над извлеченным значением перед операцией

payload cvs

По этой команде осуществляется детектирование атак на CVS.

payload bufferlen <IDSM_RANGE> >

Эта команда задает ограничения на длину текущего буфера обнаружения.

payload ssl-version {VER,5} [no]

Эта команда задает отслеживание версии SSL, согласованной двумя точками SSL-соединения.

Правило срабатывает,если замечена хотя бы одна из версий (OR-логика).

Если применено несколько данных команд, между ними используется AND-логика.

Параметры:

- VER : принимает значения: ssl2,ssl3,tls10,tls11,tls12 для версий SSLv2,SSLv3,TLS1.0,TLS1.1,TLS1.2, соответственно;
- no - оператор отрицания; правило будет проверять, что не используется ни одна из указанных версий.

payload ssl-state {VER,5} [no]

Эта команда задает отслеживание состояния SSL-соединения во время процесса приветствия и обмена ключом.

Правило срабатывает, если замечено хотя бы одно из состояний (OR-логика).

Если применено несколько данных команд, между ними используется AND-логика (правило сработает, если соединение достигло всех указанных множеств состояний).

Параметры:

- STATE : принимает значения: client-hello, server-hello, client-keyx, server-keyx, unknown для состояний Clinet Hello (клиент отослал сообщение Client Hello), Server Hello (сервер ответил на сообщение Client Hello сообщением Server Hello), Client Key Exchange, Server Key Exchange и неизвестное состояние;
- no - оператор отрицания; правило будет проверять, что не было достигнуто ни одно из указанных состояний.

payload dce-opnum {<OP|OP1-OP2>,4} [no] По этой команде осуществляется поиск одного из указанных номеров операций (OP) DCE/RPC. Возможно указание интервала номеров операций OP1-OP2.

Возможно указание оператора отрицания no - правило сработает,если указанные номера не были обнаружены.

payload priv-pattern <credit-card|us-ssn|us-ssn-nodash|REGEX> <COUNT>

эта команда задает тип отслеживаемых личных данных.

Параметры:

- credit-card - номера кредитных карт;
- us-ssn - номера социального обеспечения США;
- us-ssn-nodash - номера социального обеспечения США без дефисов;

- REGEX - собственный образец поиска;
- COUNT - число обнаружений личных данных до генерации предупреждения.

payload sip-method {METHOD,4} [no]

Эта команда задает тип отслеживаемых SIP-методов запроса. Поддерживаются следующие методы: invite,cancel,ack,bye,register,options,refer,subscribe,date,join,info,message,notify,prack.

Необязательный параметр no - это оператор отрицания.

payload sip-status-code {CODE|CODE1-CODE2,4}

Эта команда задает тип отслеживаемых SIP-кодов ответа (числа от 100 до 999, или интервалы таких чисел).

payload sip-header {CODE|CODE1-CODE2,4}

Эта команда задает тип отслеживаемых SIP-кодов ответа (числа от 100 до 999, или интервалы таких чисел).

payload enip command <IDSM_RANGE>

Эта команда предписывает проверять на указанную команду протокола CIP/ENIP.

payload enip request

Эта команда предписывает искать запрос протокола CIP/ENIP.

payload enip response

Эта команда предписывает искать ответ протокола CIP/ENIP.

payload gtp version <VAL>

Эта команда предписывает проверять на указанную версию протокола GTP.

payload gtp info <VAL>

Эта команда предписывает проверять на наличие указанного информационного элемента GTP.

payload gtp type {VAL,4}

Эта команда предписывает проверять на наличие указанных типов сообщений GTP.

payload modbus func <VAL>

Эта команда предписывает проверять на наличие указанного значения Function code в заголовке Modbus-протокола.

payload modbus unit <VAL>

Эта команда предписывает проверять на наличие указанного значения Unit ID в заголовке Modbus-протокола.

payload dnp3 flags {VAL,16}

Эта команда предписывает проверять на наличие указанных флагов Internal Indicators в DNP3.

Будучи заданными в одной команде, между флагами применяется операция ИЛИ.

Будучи заданными в разных командах, между множествами флагов этих команд применяется операция И.

payload dnp3 func <VAL>

Эта команда предписывает проверять на наличие указанного значения Function code в заголовке DNP3-протокола.

payload dnp3 obj <GROUP> <OVAR>

Эта команда предписывает проверять на наличие указанных объектных заголовков в DNP3-протоколе.

Параметры:

- GROUP - группа объекта;
- VAR - объект.

46.6.10 Пользовательское правило: заголовочные опции.

Сначала рассмотрим команды, для которых можно задавать приоритет (PRIO). После них будут описаны команды без приоритета.

[PRIO] header stream <reassembly <enable|disable> > <server|client|both> [noalert] [fastpath]

Эта команда включает или выключает пересборку TCP-пакетов для трафика, попадающего в правило. Параметры:

- enable/disable - включить/выключить пересборку TCP пакетов;
- server/client/both - тип трафика (серверный,клиентский или оба типа);
- fastpath - игнорировать остальные соединения;
- noalert - не выдавать предупреждения.

Должен быть включен инспектор stream.

header frag-offset <IDSM_RANGE>

Эта команда задает ограничение на значение смещения фрагмента в IP-заголовке.

header ttl <IDSM_RANGE>

Эта команда задает ограничение на значение TTL в IP-заголовке.

header tos <IDSM_RANGE>

Эта команда задает ограничение на значение TOS в IP-заголовке.

header id <IDSM_RANGE>

Эта команда проверяет значение ID в IP-заголовке.

header ipopts <IPOPTS>

Эта команда проверяет на наличие IP-опции в IP-заголовке. Возможные опции:

- rr - запись маршрута;

- eol - конец списка;
- nop - нет опции;
- ts - временная метка;
- sec - IP-безопасность;
- esec - расширенная IP-безопасность;
- lsrr - гибкая маршрутизация от источника;
- lsrre - расширенная гибкая маршрутизация от источника;
- ssrr - жесткая маршрутизация от источника;
- satid - идентификатор потока;
- any - любые опции.

header frag-flags <{more-frags|dont-frag|rsrv-bit,3}> [more|any|not]

Эта команда предписывает проверять, установлены ли биты фрагментации или зарезервированный бит IP-заголовка.

Параметры:

- more-frags - бит More Fragments (будут еще фрагменты);
- dont-frag - бит Don’t Fragment (не фрагментировать);
- rsrv-bit - зарезервированный бит;
- more - обнаружение срабатывает, если установлены, как минимум, указанные биты;
- any - обнаружение срабатывает, если установлены любые из указанных битов;
- not - обнаружение срабатывает, если указанные биты не установлены.

header tcp-flags <{BITS,8}> [more|any|not]

Эта команда предписывает проверять, установлены ли биты фрагментации или зарезервированный бит IP-заголовка.

Параметры:

- BITS - флаги TCP-пакета: fin,syn,rst,psh,ack,urg,cwr,ece,none (нет флагов);
- more - обнаружение срабатывает, если установлены, как минимум, указанные флаги;
- any - обнаружение срабатывает, если установлены любые из указанных флагов;
- not - обнаружение срабатывает, если указанные флаги не установлены.

header data-size <IDSM_RANGE>

Эта команда задает обнаружение пакетов с данными заданной длины.

header seq <IDSM_RANGE>

Эта команда предписывает проверять значение номера последовательности TCP-пакета (TCP Sequence number).

header ack <IDSM_RANGE>

Эта команда предписывает проверять значение номера уведомления TCP-пакета (TCP Acknowledge number).

header window-size <IDSM_RANGE>

Эта команда предписывает проверять значение размера окна TCP-пакета.

header icmp-code <IDSM_RANGE>

Эта команда предписывает проверять значение кода ICMP-ответа.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-type <IDSM_RANGE>

Эта команда предписывает проверять значение типа сообщения ICMP.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-id <IDSM_RANGE>

Эта команда предписывает проверять значение идентификатора сообщения ICMP.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-seq <IDSM_RANGE>

Эта команда предписывает проверять значение номера последовательности сообщения ICMP.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header rpc <VAL1> [version <VAL2|any>] [procedure <VAL3|any>]

Эта команда предписывает проверять значение номера приложения,версии и процедуры RPC для SunRPC-запросов.

Параметры:

- VAL1 - номер приложения RPC;
- VAL2 - номер версии RPC;
- VAL3 - номер процедуры RPC.

header ip-proto [not|more|less] <VAL>

Эта команда предписывает проверять значение типа протокола,вложенного в IP. Параметры:

- LOGIC - задает оператор сравнения со значением VAL: more,less,not;
- VAL0,VAL1 - интервал значений.

Данная опция может быть задана только при ip-типе протокола в правиле.

header stream-size <client|server|both|either> <LOGIC> <VAL>

Эта команда предписывает рассматривать только сессии заданного размера. Параметры:

- LOGIC: задает оператор сравнения со значением VAL: more,less,not,equal,more-equal,less-equal;
- client,server,both,either - тип трафика: от клиента, от сервера, оба типа, любой из типов;
- VAL - размер данных.

Должен быть включен инспектор stream.

header flow [dir <to-server|from-server>] [state <established|not-esatblished|stateless>] [stream <no-stream|only-stream>] [frag <no-frag|only-frag>]

Эта команда задает, какой именно трафик наблюдать, исходя из следующих его свойств: направление, состояние, фрагментация, пересборка.

Параметры:

- dir - направление трафика - запросы клиентов или ответы серверов (from-server);
- state - состояние TCP-соединений: установленные, неустановленные, независимо от состояния;
- stream - no-stream - не учитывать пересобранные пакеты;
- only-stream - учитывать только пересобранные пакеты;
- frag - no-frag - не учитывать фрагментированные пакеты;
- only-frag - учитывать только фрагментированные пакеты.

header flowbits <OP> <BITS|any|all> <BITMAP>

Эта команда позволяет настроить отслеживание сессий для транспортных протоколов. Для TCP-сессий данная опция позволяет правилу отслеживать состояние протоколов уровня приложений.

Параметры:

- BITMAP - имя битовой карты, над которой производится операция OP над битами BITS;
- BITS - биты битовой карты: цифры от 1 до 8; any - любой бит; all - все биты; между битами могут быть операции AND (&) или OR(|);
- OP - операция с битовой картой:
  - set - установить указанные биты BITS в битовой карте BITMAP;
  - unset - сбросить указанные биты BITS в битовой карте BITMAP;
  - setx - установить указанные биты BITS в битовой карте BITMAP и сбросить все остальные;
  - toogle - установить указанные биты BITS в битовой карте BITMAP и сбросить все остальные;
  - isset - проверить, установлены ли биты BITS в битовой карте BITMAP; может быть применен модификатор or для битов BITS, вместо модификатора AND по умолчанию;
  - isnotset - проверить, не установлены ли биты BITS в битовой карте BITMAP; может быть применен модификатор or для битов BITS,вместо умалчиваемого AND;
  - noalert - не предупреждать о выполнениях команды set,unset,toggle в правиле;
  - reset - сбросить все биты указанной группы.

46.6.11 Пользовательское правило: опции пост-обнаружения.

Данные опции будут применены в случае срабатывания остальных опций и заголовка правила.

postdet detection-filter <src|dst> <COUNT/NSEC>

Данной опцией мы определяем частоту, которая должна быть превышена, чтобы правило создало предупреждение.

Частота определяется как COUNT/NSEC - число срабатываний (COUNT) правила за указанное число секунд(NSEC). Подсчет ведется по уникальным IP-адресам источника (задан параметр src) или назначения (задан параметр dst) атаки.

postdet session <printable | binary | all>

Позволяет экстрагировать и запротоколировать пользовательские данные из TCP-сессии. Данных протоколируются в одном из 3х указанных форматов:

- printable - записываются только данные, которые могли бы быть введены или прочитаны пользователем
- binary - данные в бинарном виде (как есть)
- all - все не-printable данные заменяются их hex-представлением.

postdet tag <host-src|host-dst|session> [packets <PNUM>] [seconds <SNUM>] [bytes <BNUM>]

Данная опция задает режим теггирования пакетов. Теггирование пакетов - это процесс, когда служба помечает пакеты для дальнейшего протоколирования их, если выполнены некоторые критерии.

Перед дальнейшим описанием опции условимся называть пакет, который привел к срабатыванию правила, активатором.

Существует два режима теггирования - стандартный и расширенный:

- в стандартном режиме мы можем протоколировать только пакеты после активатора
- в расширенном режиме мы можем протоколировать пакеты как после, так и до активатора

Рассмотрим алгоритм стандартного теггирования:

1. служба получает пакет-активатор, приводящий к срабатыванию правила
2. активатор протоколируется
3. исходя из опции postdet tag служба начинает помечать пакеты следующие за активатором
4. помеченные пакеты протоколируются, если они попадают под критерий поиска

Рассмотрим алгоритм расширенного теггирования:

1. служба сохраняет в специальном буфере каждый IP-пакет, размер буфера (в пакетах) задается командой limit prepackets; в этом буфере все пакеты являются априори помеченными,т.к. мы не знаем что за правило сработает
2. служба получает пакет-активатор, приводящий к срабатыванию правила

3. активатор протоколируется

4. исходя из опции postdet tag служба начинает помечать пакеты следующие за активатором

5. выбираются пакеты из буфера и если они попадают под критерий поиска, то они протоколируются

6. также стандартные помеченные пакеты протоколируются, если они попадают под критерий поиска

Рассмотрим параметры опции postdet tag:

- host-src - поиск среди помеченных пакетов осуществляется по IP-адресу источника пакета-активатора
- host-dst - поиск среди помеченных пакетов осуществляется по IP-адресу назначения пакета-активатора

- session - поиск среди помеченных пакетов осуществляется по сессии, соответсвующей пакету-активатору

- seconds - помечаем все пакеты после пакета-активатора в течение указанного числа секунд

- bytes - помечаем все пакеты после пакета-активатора, пока размер трафика, составляемого этими пакетами, не превысило указанное число байт

- packets - задает число пакетов, которые нужно пометить; формат PNUM: N[-M], т.е.:
  - N - включает стандартное тегирование, где N - число пакетов после активатора
  - N-M - включает расширенное теггирование, где N - число пакетов до активатора,M - число пакетов после активатора

Как описано выше, расширенный режим теггирования включается двумя опциями:

- limit prepackets - глобальная опция, задающая максимальное число хранимых пакетов в буфере (для каждого пакета предоставляется 64КБ); например : limit prepackets 100
- параметр packets N-M опции postdet tag; например: postdet tag host packets 5-2 src

46.6.12 Управление правилами в изделии Dionis-SMP

Модуль СОВ Dionis-NX анализирует трафик согласно заданным правилам обнаружения вторжений (сигнатурам). Изделие Dionis-SMP позволяет изменять заданные правила для устройств Dionis-NX. Выбор профиля устройства и загрузка правил:

1. Настроить правила (подробно описано в разделе «Управление правилами на устройствах» настоящего руководства)

2. Выбрать устройство в таблице на странице «Управление»–>«Устройства», щелкнув на соответствующую строку

3. Выбрать в раскрывающимся списке профиль для загрузки правил

4. Нажать кнопку «Загрузить»

Устройство будет использовать отправленный ему набор правил для обнаружения атак и выполнять определенные пользователем действия.

46.6.13 Подсистема активного реагирования

После срабатывания правила возможно принятие особых, дополнительных действий с атакующей машиной и/или атакующим пакетом.

Это осуществляется подсистемой активного реагирования (AP).

Для настройки подсистемы AP используйте следующие команды:

- команды раздела reject: reject reset, reject unreachable - посылать ICMP Unreach и/или TCP Reset пакеты для сброса сессии

- команды раздела react: послать HTML-страницу с сообщением о запрете доступа и затем сбросить сессию через TCP Reset

- команды раздела rewrite: замена содержимого пакета правилами типа rewrite с опцией replace

- команды раздела active: описывают общие настройки подсистемы AP

Таким образом, существует 3 типа активного реагирования - reject, react и rewrite. Каждый из типов включается одноименной командой. Кроме того существует одноименный тип правил, с помощью которого данный тип AP используется. Для типа AP rewrite помимо правила rewrite должна использоваться опция payload replace в правиле, которая задает строку, на которую следует заменить найденный образец. Образец ищется уже известной нам опцией payload content.

Более подробно примеры использования подсистемы AP смотри в Примерах (3 и 4 пример).

46.7 Настройка подсистемы событий

После прохождения пакета через описанные выше подсистемы создается событие, описывающее реакцию службы на данный пакет.

Для настройки подсистемы событий используйте следующие команды:

- некоторые команды раздела **alerts**: alerts tunnel-verdicts, alerts event-filter-memcap, alerts rate-filter-memcap, alerts logref
- команда suppress: подавление события после срабатывания правила, в зависимости от частоты срабатывания
- команды раздела event

Для примера рассмотрим команды раздела event.

event filter setup <GID> <SID> <limit|threshold|both> <src|dst> <COUNT/SEC>

Эта команда предназначена для уменьшения числа ложных предупреждений.

Для каждой пары GID,SID может быть задана только одна команда. Параметры:

- GID,SID - идентификаторы подсистемы и правила, для которых применяются данные параметры; особые комбинации :
  - gid != 0, sid = 0 - команда применяется для всех правил группы gid;
  - gid = 0 , sid = 0 - команда применяется для всех правил;
  - gid = 0, sid ! = 0 - неразрешенная комбинация;
- limit - предупреждать о первых COUNT-событиях за интервал времени SEC; затем игнорировать события для оставшейся части интервала SEC;
- threshold - предупреждать о каждом COUNT-событии за интервал времени SEC;
- both - предупреждать один раз после обнаружения COUNT-числа событий в течение интервала SEC; затем игнорировать события для оставшейся части интервала SEC;
- src - счетчик событий COUNT ведется по числу уникальных IP-адресов источника;
- dst - счетчик событий COUNT ведется по числу уникальных IP-адресов назначения;
- COUNT/SEC - число событий за интервал времени SEC(в секундах).

event queue <max|log|order|process-all> >

Эта команда задает параметры очереди событий. Параметры:

- max - размер очереди событий; задает максимальное число событий, запоминаемых для каждого пакета/сессии;
- log - задает максимальное число событий, регистрируемых в журнале для каждого пакета/сессии;
- order - задает тип сортировки событий в очереди: priority - по важности, content-length - по длине параметра опции content-правила;

- process-all - использовать все найденные группы действий, если не включено - только 1ю найденную группу действий.

rate filter <GID> <SID> <by-src|by-dst|by-rule> <NUM> <INT> <ACT> <TO> [NET]

Эта команда задает фильтр событий по частоте. Это позволяет предотвращать атаки, основанные на частоте обнаружения некоторого события. Пользователь может сконфигурировать службу так, что при достижении заданной частоты срабатывания некоторого правила его действие меняется на заданное на указанный промежуток времени.

Параметры:

- GID,SID - идентификаторы правила, для которых применяется данная команда;
- by-src - счетчик событий COUNT; ведется по числу уникальных IP-адресов источника;
- by-dst - счетчик событий COUNT; ведется по числу уникальных IP-адресов назначения;
- by-rule - счетчик событий COUNT; ведется по числу срабатывания правила;
- NUM - количество событий за интервал времени SEC;
- INT - интервал времени (сек.), за который считается счетчик событий NUM; если он равен 0 , то NUM - это общее число событий, причем для GID не равного 135 не будет производится возврат к старому действию по истечении интервала TO;
- ACT - новое действие для правила, которое будет им использоваться в результате срабатывания данного фильтра частоты;
- TO - время в секундах, через которое правило вернет себе старое действие вместо нового действия ACT, которое начало использоваться в результате срабатывания данного фильтра частоты; если равно 0, то не будет производиться возврат к старому действию;
- NET - сеть источника (если задан параметр src) или назначения (если задан параметр dst), к которым применяется фильтр

46.8 Настройка подсистемы журналирования

После прохождения пакета через описанные выше подсистемы имеется возможность записать в лог (журнал) событие связанное с пакетом.

Для настройки подсистемы журналирования используйте команды раздела log.

log <local> [dump] [verbose]

Эта команда включает ведение журнала в локальной файловой системе. В режиме verbose - печать пакета начиная от Layer-2. В режиме dump - печать в лог также и полезной нагрузки (данных) пакета.

Этот тип лога осуществляется непосредственно самой службой.

log <syslog> <SRV_IP:[PORT]> <SENSOR> <udp|tcp> [IP_FROM] [dump]

Эта команда включает ведение журнала на удаленный syslog-сервер.

Параметры:

- SRV_IP - IP-адрес syslog-сервера
- PORT - порт syslog-сервера, если не указан - 1468 для TCP- и 514 для UDP-передачи.
- SENSOR - имя сенсора
- IP_FROM - IP-адрес от которого слать атаки на syslog-сервер

- udp/tcp - включаем режим передачи данных UDP или TCP
- dump - пересылать также содержимое дейтаграммы

Этот тип лога осуществляется отдельным процессом, называемым стандартный логгер (Standart logger).

log <BD> <SRV> [user <USR>] [password <PAS>] [sensor <SENS>] [db <DB>] [brief]

Эта команда включает режим пересылки информации о предупреждениях на удаленный сервер в базу данных snortdb.

Параметры:

- BD : типа базы данных postgresql или mysql;
- SRV : адрес удаленного сервера базы данных и, возможно, порт;
- USR : пользователь (по умолчанию - snort);
- PAS : пароль (по умолчанию - pass);
- SENS : имя сенсора (по умолчанию - имя узла текущей системы);
- DB : имя сенсора (по умолчанию - snortdb);
- brief : краткая информация (по умолчанию - полная);

- dump : пересылать также содержимое дейтаграммы.

Этот тип лога осуществляется отдельным процессом, называемым стандартный логгер (Standart logger).

log diamant {<SUS_IP>[:PORT],8} [sensor <SNAM>] [id <ID>] [size <SZ>] [asize <ASZ>] [bsize <BSZ>] [logs <NLOGS>] [metaserver] [dump] [ID]

Эта команда включает режим журналирования в разделяемую память и отсылки предупреждений на удаленные сервер(а) Diamant.

Параметры команды:

- SUS_IP:[PORT] - IP-адрес и порт Diamant-сервера (по умолчанию порт 4150) или метасервера (по умолчанию порт 4161), откуда получить список Diamant-серверов;

- SNAM - название сенсора (sensor1 по умолчанию);

- ID - идентификатор сенсора (1 по умолчанию);

- SZ - размер разделяемой памяти для хранения предупреждений (1024 КБ по умолчанию);

- ASZ - максимальный размер всех файлов атак (1024 КБ по умолчанию)

- BSZ - размер внутреннего буфера логгера для хранения предупреждений (256 КБ по умолчанию)

- NLOGS - максимальное количество файлов атак, хранимых на диске (1024 по умолчанию);

- dump - передавать данные пакета, на который выдано предупреждение.
- metaserver - использовать метасервер вместо Диамант-сервера

Этот тип лога осуществляется внешним процессом, называемым Диамант логгер (Diamant logger).

Если указан тип лога diamant, то другие типы лога, кроме local, не будут активны. Больший приоритет имеет тип лога diamant, чем типы syslog, postgres, mysql.

46.9 Настройка подсистемы счетчиков

Для каждой подсистемы службы, включая инспекторы протоколов, существуют счетчики, которые подсчитывают число различных событий, связанных с подсистемой или инспектором. Будем далее называть модулем подсистему службы,которая функционально и по смыслу отличается от других подсистем, а также имеет набор счетчиков.

Служба может протоколировать в отдельные лог-файлы значения счетчиков разных модулей. Это позволяет увидеть узкие места работоспособности службы и понять особенности трафика, который через нее проходит.

perfmon <base|flow|flow-ip|cpu>

Задает тип счетчика для протоколирования:

- base - протоколировать предопределенный набор базовых счетчиков и включает возможность использования отдельных счетчиков командой perfmon stat
- cpu - протоколировать счетчики использования ЦПУ и потоков службы
- flow - протоколировать счетчики использования протоколов L3/L4
- flow-ip - протоколировать счетчики IP-адресов и сетей

perfmon stat <MOD> [CNT]

Команда отключает базовый набор счетчиков (задаваемый командой perfmon base) и включает указанный счетчик CNT модуля MOD. Если MOD равен all, будут включены все счетчики всех модулей. Если CNT не указан, то будут включены все счетчики указанного модуля MOD.

perfmon options flow-ip-memcap <N>

Задает максимальный объем памяти в байтах для отслеживания счетчиков IP-адресов (см. команду perfmon flow-ip)

perfmon options max-ports <N>

Задает максимальное количество отслеживаемых портов при включении команды perfmon flow-ip. По умолчанию: 1024.

perfmon options seconds <N>

Задает интервал записи информации о счетчиках в лог. По умолчанию: 60 сек.

perfmon options packets <N>

Задает минимальное количество пакетов, которые должны пройти через службу, для записи информации о счетчиках в лог. По умолчанию: 10000.

46.10 Начальная настройка

Служба IDSM требует для своего функционирования достаточно много оперативной памяти. Для полноценного использования необходимо хотя бы 500Мб оперативной памяти. Это зависит, помимо настройки, от числа используемых потоков (задается командой **threads total**) и числа включенных правил (задается командой **ruleset*).

Для настройки службы войдите в ее режим конфигурации, выполнив команду: **service idsm**

Служба не требует никаких настроек для запуска. По умолчанию, она запустится без описанных выше подсистем в режиме выборки трафика **daq mode nfq** и в режиме службы **mode ids**.

Для полноценной работы службы следует задать такие настройки:

- режим выборки трафика и режим работы службы: команды **daq mode** и **mode**
- если режим выборки трафика daq mode nfq: необходимо установить правила отбора трафика командой ipsrule или ipsrule6 (для daq mode nfq6) и, возможно, интерфейс, который будет прослушиваться, командой **iface**
- если режим выборки трафика daq mode afp: необходимо задать пару интерфейсов командой **iface-pair** для создания моста, что необходимо для режима daq mode afp
- количество потоков выборки и обработки трафика: команда **threads total** , по умолчанию количество потоков берется исходя из числа ядер процессора (threads total 0)

- включить необходимые инспекторы протоколов: как минимум необходим inspector stream,stream-ip, stream-tcp отвечающий за анализ TCP-трафика и пересборку TCP-сегментов.
- включить необходимые группы системных правил: **ruleset detection** или **ruleset detection-all** для включения всех системных групп правил;

- журналирование, например, самое простое: log local.

Дадим пояснение к некоторым командам.

mode <ids|ips>

Данная команда определяет ,какой из двух режимов работы службы будет использован: режим обнаружения (ids) или предотвращения вторжений (ips).

Отличия режимов:

- в режиме ips: работают такие действия правил как log,pass,alert, drop, block, reset; причем drop, block и reset могут блокировать пакет.
- в режиме ids: работают такие действия правил как alert, все остальные сводятся к alert (кроме pass).

iface-pair <IFACE1> <IFACE2>

Определение интерфейса, который необходимо использовать для анализа и фильтрации трафика (для режима daq mode afp).

Трафик из первого интерфейса копируется на второй и наоборот. Те пакеты, которые отбрасываются - не копируются. В качестве IFACE1/2 могут выступать, например, интерфейс защищаемой сети и интерфейс внешней сети (Интернет) - источник атак (угроз безопасности).

Интерфейсы IFACE1, IFACE2 не должны использоваться кем-либо, в том числе в интерфейсе типа bridge, т.к. служба сама создает мост между этими интерфейсами.

В случае, если на интерфейс IFACE1 и/или IFACE2 приходит VLAN-трафик например с VLAN_ID = 5, необходимо также создать VLAN-интерфейсы с VLAN_ID = 5, и указать их в iface-pair команде, вместо реальных ethernet-интерфейсов. Например:

```
interface ethernet 0.5
 enable

interface ethernet 1.5
 enable

service idsm
 daq mode afp
 iface—pair ethernet 1.5 ethernet 0.5
 ...
 enable
```

Для daq-режима afpacket в режиме ips необходимо отключить перегрузку пакетов (offload режим интерфейса) для интерфейсов-получателей, т.е. таких, которые указаны первыми в командах iface-pair. Если этого не сделать, то возможно получение фрэймов размером большим, чем daq snaplen, что приведет к усечению данных при чтении из интерфейса-получателя и в результате - к неполной передаче данных при записи усеченного фрэйма в интерфейс-приемник. Для этого в настройках интерфейсов, которые прописаны первыми в командах iface-pair, необходимо задать:

```
offload lro off
offload gro off
```

Если это не будет сделано, то при запуске службы будет выведена ошибка.

46.11 Форматы журналов

Рассмотрим далее форматы журналов службы IDSM.

46.11.1 Формат журнала log syslog.

Данные пересылаемые на syslog-сервер передаются в текстовом виде в определенном формате. Рассмотрим его далее на примере различных атак.

46.11.1.1 UDP атака в режиме dump

Сообщение об UDP-атаке в случае,если в службе IDSM задана команда log syslog с опцией dump:

[ALERT]: [sens1]] || 2017—04—13 06:47:53.290+003 5 [8001:1000001:1] Snort Alert [8001:1000001:1] || trojan—activity || 17 192.168.33.10 192.168.33.63 4 20 0 33 19527 2 0 64 30616 || 38237 7 13 25347 || 33 450000214C47400040112AEBC0A8210AC0A8213F955D0007000D630368656C6C6F

Описание события ИДС:

1. ALERT - тип предупреждения:
- ALERT - предупреждение без блокировки
- DROP - предупреждение с блокировкой пакета (режим mode ips, drop-правило)
- TAG - тагированый пакет (пакет после атаки)
- PTAG - претагированый пакет (пакет до атаки)
2. sens1 - имя сенсора указанное в команде log syslog
3. 2017-04-13 06:47:53.290+003 - дата атаки, формат ГГГГ:ММ:ДД ЧЧ:ММ:СС.МИЛИСЕК+ТАЙМЗОНА, таймзона задается командой timezone config-режима
4. 5 - приоритет атаки (то, что задается general severity в правиле)
5. [8001:1000001:1] - идетификатор атаки, GID:SID:REV, где REV- ревизия правила
6. Snort Alert [8001:1000001:1] - эта строка пишется вместо сообщения из правила, задаваемое в general message опции правила, т.к. оно не передается в режиме log syslog; сообщение ищется логгером по sid/gid среди стандартных правил, т.к. наше правило не стандартное, а ручное, то вместо сообщения пишется указанная строка.
7. trojan-activity - классификация предупреждения, то что задается командой general class, по умолчанию - non-suspicios.

Далее идет описание IP-заголовка:

8. 17 - номер вышележащего протокола,например протокола транспортного уровня, в данном случае UDP
9. 192.168.33.10 - IP-адрес источника атаки
10. 192.168.33.63 - IP-адрес назначения атаки
11. 4 - версия IP протокола
12. 20 - длина IP заголовка в байтах
13. 0 - ToS -тип обслуживания
14. 33 - длина всей IP-датаграмы, включая заголовок
15. 19527 - идентификатор IP-пакета
16. 2 - флаги IP
17. 0 - смещение фрагмента

18. 64 - время жизни IP-датаграмы

19. 30616 - контрольная сумма IP-датаграмы

Далее идет описание вышележащего протокола, в данном случае UDP:
20. 38237 - порт источника атаки
21. 7 - порт назначения атаки
22. 13 - длина UDP-пакета включая заголовок
23. 25347 - контрольная сумма UDP-пакета
24. 33 - см. п.14 (длина IP-датаграммы)
25. 450000214C47400040112AEBC0A8210AC0A8213F955D0007000D630368656C6C6F - содержимое IP датаграмы.

46.11.1.2 UDP атака без режима dump

Сообщение об UDP-атаке в случае,если в службе IDSM задана команда log syslog без опции dump:

[ALERT] Snort Alert [8001:1000001:1] [Classification: A Network Trojan was Detected] [Priority: 5]: <ipsnfq> {UDP} 192.168.33.10:47293 —> 192.168.33.63:7

Поля:

1. ALERT - аналогично режиму dump

2. Snort Alert [8001:1000001:1] - аналогично режиму dump

3. Classification: A Network Trojan was Detected - более развернутое описание класса атаки; это то, что в режиме dump было trojan-activity

4. Priority: 5 - приоритет, аналогично режиму dump

5. ipsnfq - режим захвата трафика службы idsm

6. UDP - буквенное название протокола, вместо 17 в режиме dump

7. 192.168.33.10:47293 -> 192.168.33.63:7 - адреса и порты источника и назначения атаки.

46.11.1.3 ICMP атака в режиме dump

Сообщение об ICMP-атаке в случае,если в службе IDSM задана команда log syslog с опцией dump:

[ALERT]: [sens1]] || 2017—04—13 07:49:44.584+003 5 [8001:1000001:1] Snort Alert [8001:1000001:1] || trojan—activity || 1 192.168.33.10 192.168.33.63 4 20 0 84 24352 2 0 64 6127 || 8 0 39947 27318 1 || 84 450000545F204000400117EFC0A8210AC0A8213F08009C0B6AB60001906EF758 000000009FA20B0000000000101112131415161718191A1B1C1D1E1F202122232425262728292A2B2C2D 2E2F3031323334353637

Отличие от UDP с опцией dump только в полях самого вышележащего протокола, поэтому опишем только поля ICMP-заголовка:

1. 8 - тип ICMP - эхо запрос
2. 0 - код ICMP
3. 39947 - контр сумма ICMP
4. 27318 - идентификатор эхо-запроса ICMP
5. 1 - номер последовательности эхо-запроса ICMP

46.11.1.4 ICMP атака без режима dump

Аналогично UDP атаке без режима dump.

46.11.1.5 TCP атака в режиме dump

Сообщение о TCP-атаке в случае,если в службе IDSM задана команда log syslog с опцией dump:

[ALERT]: [sens1]] || 2017—04—13 08:06:44.700+003 5 [8001:1000001:1] Snort Alert [8001:1000001:1] || trojan—activity || 6 192.168.33.10 192.168.33.63 4 20 0 60 41249 2 0 64 54784 || 40246 777 624173857 0 10 0 2 29200 42304 0 || 60 4500003CA12140004006D600C0A8210AC0A8213F9D3603092534232100000000A0027210A5400000020405B40402080A028082000000000001030307

Отличие от UDP с опцией dump только в полях самого вышележащего протокола, поэтому опишем только поля TCP-заголовка:

1. 40246 - порт источника атаки
2. 777 - порт назначения атаки
3. 624173857 - порядковый номер SEQ
4. 0 - номер подтверждения ACK
5. 10 - размер заголовка TCP в 4-байтных октетах
6. 0 - зарезервированное поле
7. 2 - флаги TCP
8. 29200 - размер окна
9. 42304 - контрольная сумма
10. 0 - указатель важности (порядковый номер октета, которым заканчиваются важные данные), если установлен флаг URG в поле п.9.

46.11.1.6 TCP атака без режима dump

Аналогично UDP атаке без режима dump.

46.12 Просмотр информации службы

show service idsm status

Эта команда выводит следующую информацию:

1. статус службы и логгеров: включен или выключен;
2. PID процесса службы и логгеров, если эти процессы включены
3. версию правил в формате DAT1 [DAT2], где DAT1 - дата последнего изменения директории etc в архиве правил (используется как версия правил), DAT2 - дата формирования правил в дистрибутиве (это либо дата копирования правил в дистрибутив при сборке системы, либо дата обновления правил).

Пример:

```
Service idsm     is enabled and running (9076)   (OK)
Standart logger  is disabled and not running     (OK)
Diamant logger   is disabled and not running     (OK)
Rules vesion: 2018—11—21.15:23:22 [2019—02—19.11:10:16]
```

show service idsm statistics

Эта команда показывает статистику сервиса.

show service idsm log [alerts|stat-base|stat-cpu|stat-flow|stat-flowip]

Эта команда показывает следующие журналы:

- alerts - предупреждения службы;
- stat-base - лог базовых счетчиков (команды perfmon base/perfmon stat);
- stat-cpu - лог cpu-счетчиков (команда perfmon cpu);
- stat-flow - лог cpu-счетчиков (команда perfmon flow);
- stat-flowip - лог cpu-счетчиков (команда perfmon flow-ip);

Если тип журналов не задан - показывает журнал службы.

show service idsm ipvars

Эта команда показывает переменные ipvar.

show service idsm portvars [VIEW]

Эта команда показывает переменные portvar.

show service idsm rules <disabled | enabled | all | msg <MSG> | id <GID> [SID] > [builtin | detection | detection-loaded | manual]

Эта команда показывает или осуществляет поиск среди правил.

Параметры:

- disabled - показать только отключенные правила;
- enabled - показать только включенные правила;
- all - показать все правила (и включенные, и отключенные);

- msg - искать в опциях msg правил образец MSG;
- id - показать правила с указанным GID и, возможно, SID;

- builtin - при показе/поиске рассматривать только группу встроенных правил;

- detection - при показе/поиске рассматривать только группу системных правил обнаружения, хранящихся на диске (без изменений, сделанных в конфигурации службы);

- detection-loaded - при показе/поиске рассматривать только группу системных правил обнаружения, которые реально загружены в работающую службу (с изменениями, сделанными в конфигурации службы);

- manual - при показе или поиске рассматривать только группу пользовательских правил;

При выдаче в начале правила будет показан его порядковый номер.

show service idsm defs

Эта команда показывает умалчиваемые значения для параметров службы и ее подсистем.

show service idsm binder

Эта команда показывает таблицу правил инспектора binder. Только для работающей службы.

show service idsm wizard

Эта команда показывает таблицу правил инспектора wizard. Только для работающей службы.

46.13 Обновление правил

Обновление правил службы возможно двух типов:

- локальное - с помощью флэш-накопителя или share: партиции
- удаленное - посредством команды от портала управления Dionis-SMP

При локальном обновлении правил их можно взять с ресурса **ftp://base.factor-ts.ru/2.0-x** .

service idsm rules update <ARCH> [light] [default-config] [default-tables] [file-magic]

По этой команде выполняется обновление встроенных системных правил из указанного архива (ARCH). Тип архива должен быть gz или bz2.

Существует **два режима** обновления: обычный и облегченный.

Для обоих режимов возможно задание опциональных параметров:

- default-config - если указано, то при обновлении будет также обновлен файл snort.lua из архива; в нем содержаться некоторые значения по-умолчанию и он используется для получения других значений по-умолчанию из файла snort_defaults.lua; обычно не следует обновлять данный файл, т.к. бывает, что он связан с исполняемым файлом службы и ,будучи обновлен, служба не сможет его использовать; необходимо четкое понимание того, что новый файл возможно использовать с текущей версией службы idsm (ее можно проверить в логе службы idsm по строке 'Snort++');

- default-tables - если указано, то при обновлении будет также обновлен файл snort_defaults.lua из архива; в нем содержаться некоторые значения по-умолчанию основных настроек службы; обычно не следует обновлять данный файл, т.к. бывает, что он связан с исполняемым файлом службы и ,будучи обновлен, служба не сможет его использовать; необходимо четкое понимание того, что новый файл возможно использовать с текущей версией службы idsm (ее можно проверить в логе службы idsm по строке 'Snort++');

- file-magic - если указано, то при обновлении будет также обновлен файл file_magic.lua из архива; в нем содержаться сигнатуры для определения типа файла; обычно не следует обновлять данный файл, т.к. бывает, что он связан с исполняемым файлом службы и ,будучи обновлен, служба не сможет его использовать; необходимо четкое понимание того, что новый файл возможно использовать с текущей версией службы idsm (ее можно проверить в логе службы idsm по строке 'Snort++'); проверить, можно ли обновить данный файл можно слелующим образом: название таблицы в данном файле (обычно это file_magic таблица) также должно быть в выводе команды 'show service idsm defs'.

Если ни один из указанных выше 3х параметров не указан, то будут обновлены исключительно правила обнаружения. В подавляющем большинстве случаев при обновлении следует использовать команду "service idsm rules update" без каких-либо опциональных параметров.

Далее рассмотрим отличия режимов обновления правил:

- **обычный** - не указан параметр light; его алгоритм следующий:
  - проверка наличия в архиве директории **rules**
  - проверка наличия в директории **etc** файлов: snort.lua, snort_default.lua, fail_magic.lua (которые могут быть пустыми, если изначально отсутствуют); проверка осуществляется, только если указан соответствующий необязательный параметр default-config, default-tables или file-magic соответственно; если же данные параметры не указаны, то эта проверка не осуществляется;
  - полная замена текущих файлов правил и сопутствующих файлов на файлы из архива, т.е. происходит полная **замена**; в случае если не указаны параметры default-config, default-tables или file-magic, то соответствующие файлы (snort.lua/snort_defaults.lua и file_magic.lua) не будут обновлены и будут использованы их текущий версии (из дистрибутива);

- **облегченный** - указан параметр light; его алгоритм следующий:
  - наличие хотя бы 1 файла в любой из директорий **etc** или **rules** архива
  - добавление к текущим файлам правил и сопутствующим файлам файлов из архива, т.е. происходит **добавление** и, возможно, **замена** при совпадении имен файлов.

Таким образом, **отличие режимов** состоит в том, что в режиме light вы можете добавить какие-то файлы правил к текущем, а в режиме обычном предоставляемый архив становится новой полной базой правил. В системе может существовать сразу два обновления, но только разных режимов, например, сначала вы обновляете в обычном режиме, затем обновляете в light-режиме. В результате к базе правил от обычного обновления добавятся файлы правил от облегченного обновления.

Соответственно, если вы скачиваете архив правил из Интернет, например с официального сайта snort, то для некоторых архивов (например, Talos правила), возможно, вам придется **перепаковать** архив для создания в нем той файловой иерархии, которая описана выше в описании режимов обновления.

При обновлении производится расчет версии правил. Версия имеет следующий формат: VERSION [RVERSION], где VERSION - это любая строка длиной до 20 символов, обычно это будет дата создания правил в формате YYYY-MM-DD.HH:MM:SS; RVERSION - дата и время добавления файлов из архива правил в систему в таком же формате как и VERSION.

Расчет **версии правил** при обновлении следующий:

- если в архиве есть файл версии с именем version, то его содержимое, обрезанное до 20 символов, будет соответствовать VERSION; если содержимое файла пустое - в качестве VERSION возьмется дата и время последней модификации этого фала;
- если в архиве нет файла версии с именем version, то в качестве VERSION возьмется самая последняя дата и время из всех найденных в архиве файлов в директориях etc и rules;
- в качестве RVERSION возьмется дата и время добавления файлов из архива правил в систему.
- полученная на предыдущих шагах версия в будет отображаться в формате VERSION [RVERSION] в строке **“Rules vesion:”** при выдаче команды **show service idsm status**

После успешного обновления делается проверка конфигурации по-умолчанию, полученных в архиве обновления, а именно: проверка файлов snort.lua, snort_default.lua и file_magic.lua из архива обновления, в случае если они были обновлены. Если при этой проверке не находятся определенных таблиц, об этом сообщается **предупреждением** так:

- нет таблицы “default_references” - сообщение Warning: No default settings for ‘general reference’ rule option
- нет таблицы “default_classifications” - сообщение Warning: No default settings for ‘general class’ rule option
- нет таблицы “default_wizard” - сообщение Warning: No default settings for port-indpenedent protocol
- нет таблицы “file_magic” - сообщение Warning: No default settings for some popular file types

- нет таблицы “binder” - сообщение Warning: No default settings for mapping services/ports/protos to inspectors (inspectors)
- нет таблицы “default_med_port_scan” - сообщение Warning: No default settings for ‘sense med’ command (inspector portscan)
- нет таблицы “default_low_port_scan” - сообщение Warning: No default settings for ‘sense low’ command (inspector portscan)
- нет таблицы “default_hi_port_scan” - сообщение Warning: No default settings for ‘sense hi’ command (inspector portscan)
- нет таблицы “gtp_v0_msg” - сообщение Warning: No default settings for GTP messages version 0 (inspector gtp)
- нет таблицы “gtp_v1_msg” - сообщение Warning: No default settings for GTP messages version 1 (inspector gtp)
- нет таблицы “gtp_v2_msg” - сообщение Warning: No default settings for GTP messages version 2 (inspector gtp)
- нет таблицы “gtp_v0_info” - сообщение Warning: No default settings for GTP infos version 0 (inspector gtp)
- нет таблицы “gtp_v1_info” - сообщение Warning: No default settings for GTP infos version 1 (inspector gtp)
- нет таблицы “gtp_v2_info” - сообщение Warning: No default settings for GTP infos version 2 (inspector gtp)

В случае успешного выполнения обновления правил будет выведено сообщение (пример):

```
Info: Rules successfully updated to version 2015—11—24.02:01:09 [2018—02—01.01:58:35]
```

service idsm rules restore

Эта команда выполняет отмену обновления (как обычного, так и облегченного) к правилам, которые были изначально в данном дистрибутиве.

В случае успешного выполнения отката обновления правил будет выведено сообщение (пример):

```
Info: Rules successfully restored to version 2018—01—22.23:21:19 [2018—01—22.23:29:48]
```

updatecheck <SRV> [TO]

Данная команда осуществляет проверку наличия более свежих правил (только std-правила), чем те, что в системе уже установлены. Проверка осуществляется чтением файла <SRV>/version на указанном сервере. В этом файле должна быть записана дата и время (далее дата) архива обновлений, хранящегося на том же сервере. Проверка осуществляется сравнением даты из файла <SRV>/version с датой

установленных в системе правил службы IDSM. Формат даты в этом файле должен совпадать с форматом даты, выдаваемой по команде show service idsm status. Формат следующий: YYYY-MM-DD.HH:MM:SS. Например: 2015-08-04.12:00:00.

Параметры команды -

- SRV - это URL директории на сервере,где хранятся архивы правил.
- TO - период проверки (в часах, от 1 до 23).

В качестве SRV можно указать следующий URL: ftp://base.factor-ts.ru (или в виде IP-адреса: 83.220.32.93). На данном ресурсе поддерживаются актуальные обновления.

Результат выполнения команды - сообщение в лог системы, которое можно посмотреть по команде **show log sysyem**. В случае наличия свежих правил в логе будет строка “Remote rules version is newer than local one: NEED UPDATE”.

При наличии свежих правил на сервере для их обновления в службе необходимо:

1. загрузить архив правил с сервера командой copy (для FTP-сервера), либо ssh get (для SSH-сервера); имя архива правил имеет формат ANAME.tar.gz, либо ANAME.tar.bz2.
2. загрузить файл с эталонной контрольной суммой архива правил; имя этого файла имеет формат CNAME.chk, где CNAME=ANAME (из п.1).
3. подсчитать контрольную сумму архива командой gostsum (см.Основы работы с интерфейсом командной строки: Команды работы с файлами)
4. сравнить подсчитанную контрольную сумму с эталонной контрольной суммой данного архива, которая указана в файле из п.2.
5. если проверка в п.4 успешна (контрольные суммы одинаковы), выполнить процедуру обновления правил командой service idsm rules upgrade PATH, где PATH - путь к файлу архива правил.

46.14 Примеры конфигураций

46.14.1 Пример 1. Обнаружение ICMP.

Обнаружение ICMP-Echo-сообщения (пинг) из сети 192.168.33.0/24 .

```
1 rule icmp
 action drop
 proto icmp
 src 192.168.33.0/24
```

46.14.2 Пример 2. Блокировка подбора SSH-пароля

Правило делает невозможным попытки SSH-соединения с одного и того же IP-адреса чаще чем 3 раза за 10 секунд:

```
 1 rule ssh
  action drop
  proto tcp
  dport 22
  header tcp—flags syn
  postdet detection—filter src 3/10
```

46.14.3 Пример 3. Сброс сессии.

Для простоты атакуемая машина и idsm-шлюз будут одной и той же машиной.

Соединяемся с атакующей машины по ssh на idsm-шлюз и запускаем ping 127.0.0.1, т.е. делаем TCP-сессию с трафиком.

На атакующей машине запускаем tcpdump.

Заходим на idsm-шлюз и настраиваем службу idsm:

```
adm@srv(config—service—idsm)# do show
iface ethernet 2
mode ips
log local
inspector stream
inspector stream—tcp
inspector stream—ip
active device ip
reject reset source
reject unreachable all
ruleset manual m
 1 rule r
  action reject
  proto tcp
  src 192.168.33.214
  general message test
enable
```

Команды типа active - настраивают подсистему Active Response (Активное реагирование), которая в свою очередь состоит из трех других подсистем - reject, react и rewrite.

Будем использовать активное реагирование типа **reject** - сброс сессии.

Разберем команды:

- active device ip: указываем, что посылка пакетов при активном реагировании будет происходить по IP-стеку с учетом маршрутизации; но можно также указать конкретный интерфейс, тогда не будет учитываться маршрутизация;

- reject reset source - задаем, куда посылать пакет TCP Reset - на адрес источника пакета, также можно указать dest - на адрес назначения пакета или both - в обе стороны;

- reject unreachable all - задаем, какие именно ICMP Unreach пакеты будут посылаться; в данном случае - Unreach Host, Network и Port, но можно указать один из указанных вариантов;

- action reject - в правиле задаем действие **reject** - сбросить сессию при обнаружении указанного в правиле пакета

После включения службы idsm наблюдаем сброс сессии и ICMP Unreach пакеты на атакующей машине, где ранее мы запустили tcpdump:

```
14:05:17.657678 IP 192.168.33.87.22 > 192.168.33.214.60226: Flags [R.], seq 3978, ack 2450, win 0,
        length 0
14:05:17.657722 IP 192.168.33.87 > 192.168.33.214: ICMP net 192.168.33.87 unreachable, length 36
14:05:17.657922 IP 192.168.33.87 > 192.168.33.214: ICMP host 192.168.33.87 unreachable, length 36
14:05:17.657948 IP 192.168.33.87 > 192.168.33.214: ICMP 192.168.33.87 tcp port 22 unreachable,
        length 36
```

Также наблюдаем разрыв ssh соединения с атакующей машины на idsm-шлюз.

На idsm-шлюзе наблюдаем в логе атак атаку, говорящую о том, что сессия была сброшена (reset):

```
19/05/18—21:18:42.836813 [reset] [**] [8000:1:0] ”test” [**] [Priority: 0] {TCP}
        192.168.33.214:60226 —> 192.168.33.87:22
```

Если вместо правила reject написать правило **react**, то сначала будет послана на источник пакета HTML страница с содержанием

```
”You are attempting to access a forbidden site. Consult your system administrator for details.”
```

И только после этого сессия будет сброшена через TCP Reset. Данную страницу можно будет увидеть на атакующей машине, только если сбрасываемая сессия, попадающая под правило, была HTTP-сессией. Также сообщение внутри страницы можно заменить командой **msg** в подсистеме **react**.

46.14.4 Пример 4. Подмена данных в пакете.

Рассмотрим применение правила типа **rewrite** для подмены данных в пакете.

Настроим idsm-шлюз - службы idsm и diweb.

```
service idsm
 iface ethernet 2
 mode ips
 log local
 rewrite
 ruleset manual m
  1 rule r
    action rewrite
```

```
   proto tcp
   dst 192.168.33.10
   sport 80
   general message replaced
   1 payload content ip6
   2 payload replace XX6
enable
```

Пояснения по конфигурации - команда rewrite включает подсистему активного реагирования **rewrite**. В результате мы можем использовать в правиле действие правил **rewrite** и команду замены контента **payload replace**: находим в данных HTTP-пакета слово ip6 и заменяем его на XX6.

Далее заходим через браузер с машины атакующего с адресом 192.168.33.10 на адрес diweb-сервера на idsm-шлюзе: http://192.168.33.87.

Видим, что в некоторых местах вместо ip6 слово XX6.

На idsm-шлюзе смотрим лог атак:

```
19/05/19—01:32:46.456762 [allow] [**] [8000:1:0] ”replace” [**] [Priority: 0] {TCP}
        192.168.33.87:80 —> 192.168.33.10:45324
```

Видим что сработало правило замены, действие стоит allow, т.к. пакет не блокируется, а происходит замена данных в нем.

46.15 Установка базы данных MySql

Допустим мы настройках службы мы задали команду для записи алертов в базу MySql: log mysql sensor sens1 10.0.0.1 db snort user snort password 1.

На сервере 10.0.0.1 установим базу данных MySql и выполним следующие настройки.

- скачаем файл команд для настройки таблиц БД с сайта barnyard2 (например,выполним: wget https://raw.githubusercontent.com/firnsy/barnyard2/master/schemas/create_mysql)
- Войдем в косоль управления MySql: mysql -u root mysql -p
- создаем БД snort: create database snort;
- переходим в созданную БД: use snort;
- выполняем команды из скаченного в п.1 файла create_mysql: source create_mysql;
- создаем пользователя snort, который получать доступ к БД с любых удаленных узлов (значок %), даем ему пароль ‘1’: CREATE USER ‘snort’@‘%’ IDENTIFIED BY ‘1’;
- даем все привилегии этому пользователю на таблицы БД snort: grant all privileges on snort.* to ‘snort’@‘%’ with grant option;
- применяем привилегии: flush privileges;

Рассмотрим команды для работы с базой данных:

- mysql -u root mysql -p - вход в оболочку mysql

- show databases; - список баз данных
- use snort - подключиться к базе snort
- show tables - список всех таблиц текущей базы данных
- select * from data; - вывод содержимого таблицы data
- select count(1) from data; - показать количество записей в таблице data
- \q - выход из оболочки mysql

46.16 Установка базы данных Postgres

В локальной сети, в которой находится Dionis-NX со службой IDSM, необходимо установить базу данных Postgres. В данную базу данных служба IDSM Diosni-NX будет передавать информацию о предупреждениях.

В данной инструкции предполагается, что используется пользователь snort и база данных snortdb:

- sudo su -
- apt-get install postgresql
- passwd postgres : задайте пароль пользователю postgres
- su postgres : войдите под пользователем postgres
- createuser -P snort : создайте пользователя базы данных; пароль пользователя: pass

> Enter password for new role: *****
> Enter it again: *****
> Shall the new role be a superuser? (y/n) n
> Shall the new role be allowed to create databases? (y/n) n
> Shall the new role be allowed to create more new roles? (y/n) n

- createdb -O snort snortdb : создаем базу данных snortdb
- Проверьте содержимое файла /etc/postgresql/9.1/main/pg_hba.conf , в нем должны быть такие строки (сеть 192.168.33.0/24 замените на сеть, в которой находится служба IDSM):

```
local   all             postgres                                peer
local   all             all                                     trust
host    all             all             127.0.0.1/32            trust
host    all             all             192.168.33.0/24         md5
```

- Скачайте файл https://raw.github.com/firnsy/barnyard2/master/schemas/create_postgresql ; приведем его содержимое здесь:

```
CREATE TABLE schema ( vseq INT4 NOT NULL, ctime TIMESTAMP with time zone NOT NULL, PRIMARY
        KEY (vseq));
INSERT INTO schema (vseq, ctime) VALUES ('107', now());
CREATE TABLE signature ( sig_id SERIAL NOT NULL, sig_name TEXT NOT NULL, sig_class_id INT8,
        sig_priority INT8, sig_rev INT8, sig_sid INT8, sig_gid INT8, PRIMARY KEY (sig_id));
CREATE INDEX sig_name_idx ON signature (sig_name);
CREATE INDEX sig_class_idx ON signature (sig_class_id);
CREATE TABLE sig_reference (sig_id INT4 NOT NULL, ref_seq INT4 NOT NULL, ref_id INT4 NOT NULL,
        PRIMARY KEY(sig_id, ref_seq));
CREATE TABLE reference ( ref_id SERIAL, ref_system_id INT4 NOT NULL, ref_tag TEXT NOT NULL,
        PRIMARY KEY (ref_id));
CREATE TABLE reference_system ( ref_system_id SERIAL, ref_system_name TEXT, PRIMARY KEY
        (ref_system_id));
CREATE TABLE sig_class ( sig_class_id SERIAL, sig_class_name TEXT NOT NULL, PRIMARY KEY
        (sig_class_id) );
CREATE INDEX sig_class_name_idx ON sig_class (sig_class_name);
CREATE TABLE event ( sid INT4 NOT NULL, cid INT8 NOT NULL, signature INT4 NOT NULL, timestamp
        timestamp with time zone NOT NULL, PRIMARY KEY (sid,cid));
CREATE INDEX signature_idx ON event (signature);
CREATE INDEX timestamp_idx ON event (timestamp);
CREATE TABLE sensor ( sid SERIAL, hostname TEXT, interface TEXT, filter TEXT, detail INT2, encoding
        INT2, last_cid INT8 NOT NULL, PRIMARY KEY (sid));
CREATE TABLE iphdr ( sid INT4 NOT NULL, cid INT8 NOT NULL, ip_src INT8 NOT NULL, ip_dst INT8 NOT
        NULL, ip_ver INT2, ip_hlen INT2, ip_tos INT2, ip_len INT4, ip_id INT4, ip_flags INT2, ip_off INT4,
        ip_ttl INT2, ip_proto INT2 NOT NULL, ip_csum INT4, PRIMARY KEY (sid,cid));
CREATE INDEX ip_src_idx ON iphdr (ip_src);
CREATE INDEX ip_dst_idx ON iphdr (ip_dst);
CREATE TABLE tcphdr( sid INT4 NOT NULL, cid INT8 NOT NULL, tcp_sport INT4 NOT NULL, tcp_dport
        INT4 NOT NULL, tcp_seq INT8, tcp_ack INT8, tcp_off INT2, tcp_res INT2, tcp_flags INT2 NOT
        NULL, tcp_win INT4, tcp_csum INT4, tcp_urp INT4, PRIMARY KEY (sid,cid));
CREATE INDEX tcp_sport_idx ON tcphdr (tcp_sport);
CREATE INDEX tcp_dport_idx ON tcphdr (tcp_dport);
CREATE INDEX tcp_flags_idx ON tcphdr (tcp_flags);
CREATE TABLE udphdr( sid INT4 NOT NULL, cid INT8 NOT NULL, udp_sport INT4 NOT NULL, udp_dport
        INT4 NOT NULL, udp_len INT4, udp_csum INT4, PRIMARY KEY (sid,cid));
CREATE INDEX udp_sport_idx ON udphdr (udp_sport);
CREATE INDEX udp_dport_idx ON udphdr (udp_dport);
CREATE TABLE icmphdr( sid INT4 NOT NULL, cid INT8 NOT NULL, icmp_type INT2 NOT NULL, icmp_code
        INT2 NOT NULL, icmp_csum INT4, icmp_id INT4, icmp_seq INT4, PRIMARY KEY (sid,cid));
CREATE INDEX icmp_type_idx ON icmphdr (icmp_type);
CREATE TABLE opt ( sid INT4 NOT NULL, cid INT8 NOT NULL, optid INT2 NOT NULL, opt_proto INT2 NOT
        NULL, opt_code INT2 NOT NULL, opt_len INT4, opt_data TEXT, PRIMARY KEY (sid,cid,optid));
CREATE TABLE data ( sid INT4 NOT NULL, cid INT8 NOT NULL, data_payload TEXT, PRIMARY KEY
        (sid,cid));
CREATE TABLE encoding(encoding_type INT2 NOT NULL, encoding_text TEXT NOT NULL, PRIMARY KEY
        (encoding_type));
INSERT INTO encoding (encoding_type, encoding_text) VALUES (0, 'hex');
```

```
INSERT INTO encoding (encoding_type, encoding_text) VALUES (1, 'base64');
INSERT INTO encoding (encoding_type, encoding_text) VALUES (2, 'ascii');
CREATE TABLE detail (detail_type INT2 NOT NULL, detail_text TEXT NOT NULL, PRIMARY KEY
        (detail_type));
INSERT INTO detail (detail_type, detail_text) VALUES (0, 'fast');
INSERT INTO detail (detail_type, detail_text) VALUES (1, 'full');
```

- cat create_postgresql | psql snortdb snort : создаете таблицы базы данных для службы IDSM
- psql snortdb snort -c «grant all privileges on database snortdb to snort;»
- psql snortdb snort -c «GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO snort;»
- проверьте что все хорошо: psql snortdb snort -c «\dp» должна выдать следующие права на все таблицы: snort=arwdDxt/snort, где snort -имя пользователя
- В файле /etc/postgresql/9.1/main/postgresql.conf добавьте(измените) следующие строки:
  - listen_addresses = '192.168.33.160,127.0.0.1' - адрес на котором слушать запросы к БД; адрес 192.168.33.160 замените на адрес, который доступен узлу службы IDSM.
- /etc/init.d/postgresql restart : перезапустите postgresql

Рассмотрим команды для работы с базой данных:

- sudo -u postgres psql - вход в оболочку postgresql
- \l - список баз данных
- \c snortdb - подключиться к базе snortdb
- \dt - список всех таблиц текущей базы данных
- select * from data; - вывод содержимого таблицы data
- select count(1) from data; - показать количество записей в таблице data
- \q - - выход из оболочки postgresql

46.17 Установка клиента просмотра базы данных BASE

Бывает удобно использовать web-клиент для работы с базой данных предупреждений. Например, существует такой клиент,как BASE. Рассмотрим шаги по его установке.

- apt-get install apache2 libapache2-mod-php5 php5-gd php5-pgsql libphp-adodb
- добавьте в файл /etc/php5/apache2/php.ini:

```
extension=pgsql.so
  extension=gd.so
```

- скачайте последнюю версию BASE с http://sourceforge.net/projects/secureideas/files/BASE/
- распакуйте файлы архива BASE в директорию /var/www/base
- установите права chown -R www-data:www-data /var/www/base
- cd /var/www/base
- cp base_conf.php.dist base_conf.php
- отредактируйте файл так чтобы следующие параметры были такими:

```
    $BASE_Language = 'russian';
$Use_Auth_System = 0;
$BASE_urlpath = '/base';
$DBlib_path = '/usr/share/php/adobd';
$DBtype = 'postgres';
$alert_dbname   = 'snortdb';
$alert_host     = '192.168.33.160';
$alert_port     = '5433'; порт, указанный в файле /etc/postgresql/9.1/main/postgresql.conf
$alert_user     = 'snort';
$alert_password = 'pass';
$archive_exists   = 0;
```

- /etc/init.d/apache2 restart
- Откройте в браузере http://localhost/base/base_db_setup.php
- Нажмите кнопку Create BASE AG
- Далее заходить нужно через http://localhost/base/index.php
- если нужно добавьте пользователей через Administrtion и включите заново систему аутентификации в /var/www/base/base_main.php: Use_Auth_System = 1;

46.18 Условные обозначения

При описании команд используются следующие обозначения:

- <VAR> - обязательный параметр VAR;
- [VAR] - необязательный параметр VAR;
- ! - отрицание смысловой нагрузки сущности, следующей за восклицательным знаком, например !10.0.0.0/24 - все сети кроме 10.0.0.0/24;
- VAR1:VAR2 - интервал чисел, например 10:123 - от 10 до 123 включительно;
- {VAR,N} - можно указать до N параметров, формат которых определен VAR, например {IPVAR|[!]IP[/MSK],8} может определять следующий набор параметров: «myvar2 10.0.0.1 !123.44.44.55»;

- <IPVAR> - задает имя определенной ранее ipvar-переменной;
- <PORTVAR> - задает имя определенной ранее portvar-переменной;
- {VAR1,…,VARn} - как минимум, один из параметров VAR1,…,VARn.

47. MAILER - служба пересылки почтовых сообщений

47.1 Введение

Dionis-NX имеет возможность отправки сообщений на указанный e-mail с помощью службы пересылки почтовых сообщений - MAILER

47.2 Настройка службы пересылки почтовых сообщений

Для настройки службы пересылки почтовых сообщений используется команда: service mailer из режима configure

```
adm@DionisNX(config)# service mailer
adm@DionisNX(config—service—mailer)#
```

Настройка службы происходит в два этапа:

1. Настройка учетных записей службы;
2. Прочая настройка.

Настройка учетных записей службы.

Учетная запись представляет собой настройки профиля для подключения к smtp-серверу

Для создания новой учетной записи или для редактирования существующей учетной записи необходимо выполнить команду account <Имя>

```
adm@DionisNX(config—service—mailer)# account gmail
adm@DionisNX(config—service—mailer—gmail)#
```

Команды доступные для настройки учетной записи.

команда	параметр
auth <on/off>	Включить или отключить аутентификацию
user <username>	Имя пользователя для аутентификации
password <password>	Пароль для аутентификации
tls <on/off>	Включить или отключить TLS/SSL
tls certcheck <on/off>	Включить или отключить проверку TLS/SSL сертификатов
from <envelope-from>	Адрес отправителя
host <hostname>	SMTP сервер
port <port>	Порт SMTP сервера
retry <Num>	Количество попыток подключения (необязательный параметр)
timeout <Num>	Тайм-аут подключения в секундах (необязательный параметр)

Прочая настройка.

Для завершения настройки службы необходимо выбрать учетную запись по умолчанию из списка уже настроенных учетных записей. Для этого необходимо выполнить команду default-acc <account-name>

```
adm@DionisNX(config—service—mailer)# default—acc gmail
```

47.2.1 Примеры настроек учетных записей для различных smtp-серверов

gmail

```
adm@DionisNX(config—service—mailer)# account gmail
adm@DionisNX(config—service—mailer—gmail)# host smtp.gmail.com
adm@DionisNX(config—service—mailer—gmail)# user username@gmail.com
adm@DionisNX(config—service—mailer—gmail)# password Secret
adm@DionisNX(config—service—mailer—gmail)# port 587
adm@DionisNX(config—service—mailer—gmail)# auth on
adm@DionisNX(config—service—mailer—gmail)# tls on
adm@DionisNX(config—service—mailer—gmail)# tls certcheck off
adm@DionisNX(config—service—mailer—gmail)# from username@gmail.com
```

mail

```
adm@DionisNX(config—service—mailer)# account mail
adm@DionisNX(config—service—mailer—mail)# host smtp.mail.ru
adm@DionisNX(config—service—mailer—mail)# user username@mail.ru
adm@DionisNX(config—service—mailer—mail)# password Secret
adm@DionisNX(config—service—mailer—mail)# port 25
adm@DionisNX(config—service—mailer—mail)# auth on
adm@DionisNX(config—service—mailer—mail)# tls on
adm@DionisNX(config—service—mailer—mail)# tls certcheck off
adm@DionisNX(config—service—mailer—mail)# from username@mail.ru
```

47.3 Отправка сообщения или файла с помощью службы MAILER

Для отправки сообщения на e-mail-адрес в enable-режиме необходимо выполнить следующую команду:

```
adm@DionisNX# mailer test@mail.com [message <message>]
```

Все параметры команды mailer:

команда	параметр
via <account>	Имя настроенной учетной записи. (Необязательный параметр. Если не указан, произойдет отправка с учетной записи, установленной по умолчанию)
<addr1,addr2…>	Список E-mail адресов для рассылки (Обязательный параметр)
message <Text>	Текст сообщения (Необязательный параметр)
subject <Text>	Тема сообщения (Необязательный параметр)
attach <file|running-config|startup-config|default-config>	Вложение (Необязательный параметр)

47.4 Настройка службы watcher для отправки сообщений с помощью службы mailer

Dionis-NX имеет возможность отслеживания некоторых системных событий и отправку e-mail сообщений с соответствующими оповещениями.

Для этого в настройке службы watcher необходимо создать react mail и для него указать:

- аккаунт службы mailer в параметре **account**;
- e-mail адрес получателя в параметре **send-to**;
- тему сообщения в параметре **subject**.

Далее созданный react mail можно связать с одним из watch, указав в параметре **link-react**.

Более подробную информацию по настройке смотрите в разделе службы watcher.

48. Служба автоконфигурирования IPv6

Возможности по автоматическому конфигурированию IPv6 достаточно широки. Для реализации этих возможностей в локальной сети должен присутствовать маршрутизатор, который отвечает на запросы автоконфигурации от хостов локальной сети. Такой маршрутизатор реализует протокол NDP (Neighbor Discovery Protocol). Метод автоконфигурирования называется SLAAC (StateLess Address AutoConfiguration).

Специальный RA-сервис (Router Advertisement service), работающий на маршрутизаторе, ожидает сетевых запросов RS (Router Solicitation) от хостов и отвечает на них сообщениями RA (Router Advertisement). Также в сеть периодически рассылаются сообщения RA без получения запроса. Сообщения RA содержат информацию, которую хосты используют для конфигурирования своих интерфейсов. В частности, передается префикс сетевого адреса, MTU, информация о маршрутизаторе по-умолчанию.

48.1 Секция настройки сервиса RA

Для настройки сервиса RA в Dionis-NX предусмотрена отдельная секция в режиме конфигурирования.

```
adm@DionisNX(config)# service ra
```

Для включения сервиса RA используется команда:

```
adm@DionisNX(cfg—srv—ra)# enable
```

Для отключения сервиса RA используется команда:

```
adm@DionisNX(cfg—srv—ra)# disable
```

Для применения измененных настроек предусмотрена команда:

```
adm@DionisNX(cfg—srv—ra)# reload
```

Любые настройки сервиса RA связаны с конкретным сетевым интерфейсом. На маршрутизаторе может присутствовать сразу несколько интерфейсов и к каждому из них может быть подключена отдельная локальная IPv6 сеть. Из каждой локальной IPv6 сети могут приходить запросы RS на свой интерфейс. Т.е. настраивая параметры конкретного интерфейса на маршрутизаторе, изменяются настройки для подключенной к этому интерфейсу локальной сети.

Чтобы зайти в секцию настроек для конкретного интерфейса используется команда:

```
adm@DionisNX(cfg—srv—ra)# iface ethernet 0
```

Все настройки о которых пойдет речь в дальнейшем вводятся внутри секции, относящейся к конкретному интерфейсу.

48.2 Общие настройки интерфейса

Чтобы сервис периодически рассылал сообщения RA, а также отвечал на запросы RS на указанном интерфейсе, необходимо включить его в активное состояние.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv send on
```

Обратная команда:

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv send off
```

Без флага “adv send” секция настройки интерфейса может хранить установленные значения, но функционирование сервиса RA на данном интерфейсе будет остановлено.

Чтобы сервис самостоятельно не рассылал групповые (multicast) сообщения RA, а только отвечал на запросы RS от конкретных хостов, используется специальная опция.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# unicast—only
```

Сообщения RA будет отправлено адресно, тому хосту, который прислал запрос RS.

Максимальный интервал времени между отправкой RA сообщения по инициативе маршрутизатора задается опцией “max-adv-interval” и измеряется в секундах. Может изменяться в пределах от 4 сек до 1800 сек. Значение по-умолчанию 600 сек.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# max—adv—interval 600
```

Минимальный интервал времени между отправкой RA сообщения по инициативе маршрутизатора задается опцией “min-adv-interval” и измеряется в секундах. Может изменяться в пределах от 3 сек до 75% от значения “max-adv-interval”. Значение по-умолчанию 33% от “max-adv-interval”.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# min—adv—interval 500
```

Минимальный интервал времени между отправкой группового RA сообщения в ответ на запрос задается опцией “min-sol-delay” и измеряется в секундах. Значение по-умолчанию 3 сек.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# min—sol—delay 3
```

Флаг “adv managed-flag” указывает хостам использовать ли управляемый (stateful) протокол, например DHCPv6, для автоконфигурирования адреса в дополнение к автоконфигурированию адресов с помощью сервиса RA (stateless).

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv managed—flag on
```

Флаг “adv other-flag” указывает хостам использовать ли управляемый (stateful) протокол, например DHCPv6, для автоконфигурирования не-адресной информации.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv other—flag on
```

Когда MTU (Maximum Transfer Unit) в сети не зафиксирован жестко, используется опция “adv mtu”, чтобы гарантировать единое значение MTU для всех хостов в сети. Может изменяться в пределах от 1280 до максимально допустимого значения MTU для данного типа соединения.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv mtu 1500
```

Опция “adv reachable-time” определяет время, в миллисекундах, в течении которого хост считает соседний хост доступным после получения подтверждения доступности. Используется алгоритмом выявления недоступности соседнего хоста (Neighbor Unreachability Detection). Значение не должно превышать 3600000 миллисекунд (1 час).

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv reachable—time 2000000
```

Опция “adv retrans-timer” задает время, в миллисекундах, между посылкой запросов соседних хостов (Neighbor Solicitation). Используется для разрешения адресов и для алгоритма выявления недоступности соседнего хоста (Neighbor Unreachability Detection).

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv retrans—timer 20000
```

Опция “adv hop-limit” позволяет задать начальное значения поля “Hop Limit” (TTL в IPv4) в заголовке исходящих IPv6 пакетов. Значение по-умолчанию 64.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv hop—limit 64
```

Опция “adv default-lifetime” определяет время в течении которого маршрутизатор считается маршрутизатором по-умолчанию. Определяется в секундах. Значение “0” означает, что маршрутизатор не является маршрутизатором по-умолчанию и не должен появляться в списке маршрутизаторов по-умолчанию. Значение поля может изменяться от значения “max-adv-interval” до 9000 секунд. Значение по-умолчанию - утроенное значение “max-adv-interval”.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv default—lifetime 8000
```

Опция “adv default-pref” определяет насколько маршрутизатор по-умолчанию предпочтителен. Может иметь значения “low”, “medium”, “high”. По-умолчанию используется “medium”, т.е. среднее значение.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv default—pref low
```

Опция “adv ll-addr” определяет будет ли адрес канального уровня (link-layer) исходящего интерфейса включен в сообщение RA.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# adv ll—addr on
```

48.3 Префикс

Префикс, полученный от маршрутизатора в сообщении RA, хост использует для формирования глобального адреса своего интерфейса. Стандарт описывает, что заданный на маршрутизаторе префикс должен иметь маску “/64”, т.е. 64 значимых бит (из 128 в полном IPv6 адресе). Хост формирует IPv6 адрес, используя префикс в качестве старшей части и MAC-адрес интерфейса (в специальной форме EUI-64) в качестве младшей части.

Префикс задается следующей командой:

```
adm@DionisNX(cfg—srv—ra—ethernet0)# prefix 2001:db8:0:1::/64
```

При вводе такой команды администратор входит в секцию, описывающую настройки для данного префикса. Все далее описанные в разделе команды относятся к этой секции.

Флаг "adv autonomous" определяет может ли данный префикс использоваться для автономного конфигурирования адреса согласно RFC 4862. Значение по-умолчанию - "on".

```
adm@DionisNX(cfg—srv—ra—ethernet0—2001:db8:0:1::/64)# adv autonomous on
```

Флаг "adv on-link" определяет может ли префикс быть признаком принадлежности хостов с таким префиксом к одной локальной сети. В общем случае в сетях IPv6 одинаковый префикс в IP-адресах хостов не означает, что хост будет доступен напрямую. Иногда хосты с одинаковым префиксом вынуждены использовать маршрутизатор для обмена трафиком. Значение по-умолчанию - "on".

```
adm@DionisNX(cfg—srv—ra—ethernet0—2001:db8:0:1::/64)# adv on—link on
```

Опция "adv valid-lifetime" определяет время (в секундах), в течении которого префикс считается актуальным для определения локальности сети (on-link). Т.е. в течении этого времени единый префикс может считаться признаком того, что хост с таким же префиксом будет доступен по прямому подключению. Может изменяться в пределах от 2 часов до бесконечности (специальное значение "infinity"). Значение по-умолчанию - 86400 сек (1 день).

```
adm@DionisNX(cfg—srv—ra—ethernet0—2001:db8:0:1::/64)# adv valid—lifetime infinity
```

Опция "adv pref-lifetime" определяет время (в секундах) в течении которого адреса, сформированные на основания префикса, считаются предпочтительными. Специальное значение "infinity" - означает бесконечное время. Значение по-умолчанию 14400 секунд (4 часа).

```
adm@DionisNX(cfg—srv—ra—ethernet0—2001:db8:0:1::/64)# adv pref—lifetime 14400
```

Флаг "deprecate-prefix" указывает, что, при выключении маршрутизатора, сервис объявит префикс недействительным. Эта опция используется только в тех случаях, когда только один маршрутизатор экспортирует такой префикс. Иначе хосты будут считать адреса, построенные на основании этого префикса недействительными, хотя существуют другие маршрутизаторы экспортирующий данный префикс.

```
adm@DionisNX(cfg—srv—ra—ethernet0—2001:db8:0:1::/64)# deprecate—prefix on
```

48.4 Маршруты

Сервис RA на конкретном интерфейсе может экспортировать маршруты для передачи хостам. Маршрут задается следующей командой.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# route 2002:db8:0:2::/64
```

После подобной команды администратор попадает в секцию конфигурирования маршрута.

Опция "adv route-lifetime" определяет время (в секундах), в течении которого маршрут считается актуальным. Если маршрут должен быть актуальным всегда, используется специальное значение "infinity". Значение по-умолчанию - утроенное значение "max-adv-interval".

```
adm@DionisNX(cfg—srv—ra—ethernet0—2002:db8:0:2::/64)# adv route—lifetime infinity
```

Опция “adv route-pref” определяет степень предпочтения. Может иметь значения “low”, “medium”, “high”. По-умолчанию используется “medium”, т.е. среднее значение.

```
adm@DionisNX(cfg—srv—ra—ethernet0—2002:db8:0:2::/64)# adv route—pref low
```

Флаг “remove-route” указывает, что, при выключении маршрутизатора, сервис объявит маршрут недействительным. Значение по умолчанию - “on” (включено).

```
adm@DionisNX(cfg—srv—ra—ethernet0—2002:db8:0:2::/64)# remove—route on
```

48.5 Клиентские хосты

По-умолчанию сервис RA рассылает сообщения RA на групповой (multicast) адрес, поэтому любой клиентский хост в сети получит это сообщение. Однако список клиентских хостов, которым будет рассылаться сообщения RA и на RS-запросы которых сервис будет отвечать, можно задать явно. В этом случае сообщения RA будут отправляться на индивидуальные (unicast) адреса хостов, а RS-запросы от других хостов будут игнорироваться.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# client 2346::1
adm@DionisNX(cfg—srv—ra—ethernet0)# client 2346::2
```

В данном случае разрешена работа с двумя клиентскими хостами.

48.6 Рекурсивный DNS-сервер

Для автоконфигурирования списка DNS-серверов на клиентских хостах существует специальная опция сообщения RA.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# rdnss
```

Внутри секции “rdnss” можно указать список DNS-серверов.

```
adm@DionisNX(cfg—srv—ra—ethernet0—rdnss)# server 2346::1
adm@DionisNX(cfg—srv—ra—ethernet0—rdnss)# server 2347::2
```

Опция “adv rdnss-lifetime” определяет время (в секундах), в течении которого список серверов считается актуальным. Если список должен быть актуальным всегда, используется специальное значение “infinity”. Специальное значение “0” означает, что список неактуален. Ненулевое значение опции не может быть меньше чем “max-adv-interval”. Значение по-умолчанию - удвоенное значение “max-adv-interval”.

```
adm@DionisNX(cfg—srv—ra—ethernet0—rdnss)# adv rdnss—lifetime infinity
```

Флаг “flush-rdnss” указывает, что, при выключении маршрутизатора, сервис объявит список DNS-серверов недействительным. Значение по умолчанию - “on” (включено).

```
adm@DionisNX(cfg—srv—ra—ethernet0—rdnss)# flush—rdnss
```

48.7 Список поиска DNS

Для автоконфигурирования списка поиска DNS (DNS search list) на клиентских хостах существует специальная опция сообщения RA.

```
adm@DionisNX(cfg—srv—ra—ethernet0)# dnssl
```

Внутри секции “dnssl” можно указать список суффиксов.

```
adm@DionisNX(cfg—srv—ra—ethernet0—dnssl)# suffix my—company.ru
adm@DionisNX(cfg—srv—ra—ethernet0—dnssl)# suffix other.com
```

Опция “adv dnssl-lifetime” определяет время (в секундах), в течении которого список суффиксов считается актуальным. Если список должен быть актуальным всегда, используется специальное значение “infinity”. Специальное значение “0” означает, что список неактуален. Ненулевое значение опции не может быть меньше чем “max-adv-interval”. Значение по-умолчанию - удвоенное значение “max-adv-interval”.

```
adm@DionisNX(cfg—srv—ra—ethernet0—dnssl)# adv dnssl—lifetime infinity
```

Флаг “flush-dnssl” указывает, что, при выключении маршрутизатора, сервис объявит список DNS суффиксов недействительным. Значение по умолчанию - “on” (включено).

```
adm@DionisNX(cfg—srv—ra—ethernet0—dnssl)# flush—dnssl
```

49. Служба WATCHER

49.1 Введение

В DionisNX существует служба WATCHER, которая может использоваться для мониторинга состояния системы, а также применяться в качестве планировщика задач.

Конфигурация службы разделена на два набора модулей:

- Модули группы **watch**, которые применяются для отслеживания различных событий системы, а также для настройки планировщика задач;
- Модули группы **react**, которые применяются для выполнения определенного действия в системе.

Чтобы при наступлении отслеживаемого события выполнить заданные действия, модули группы **watch** должны быть связаны с модулями группы **react** (см. далее).

Вход в режим конфигурации службы выполняется командой:

```
Router(config)# service watcher
Router(config—service—watcher)#
```

Для того, чтобы служба watcher стала активной, нужно выполнить команду **enable** из режима конфигурации службы:

```
Router(config—service—watcher)# enable
```

Для отключения службы, используйте команду **disable**:

```
Router(config—service—watcher)# disable
```

Для того, чтобы полностью удалить конфигурацию службы, воспользуйтесь командой:

```
Router(config)# no service watcher
```

49.2 Модули группы watch

Модули данной группы разделены на несколько типов и позволяют:

- отслеживать появление записей в системных журналах (тип *log*);
- планировать выполнение определенных действий по расписанию (тип *schedule*);
- отслеживать загрузку системы (тип *startup*);
- отслеживать размер свободного пространства файловой системы (тип *space*);
- отслеживать изменения состоянииий сетевых интерфейсов (тип *iface*);
- отслеживать результ выполнения события (тип *react*).

Для создания и настройки модуля watch необходимо выполнить слудующую команду:

```
Router(config—service—watcher)# watch <тип watch> <имя watch>
```

Удаление модуля watch watch происходит командой:

```
Router(config—service—watcher)# no watch <тип watch> <имя watch>
```

Общие параметры для watch:

- **activate** - включить данный watch;
- **link-react <тип react> <имя react>** - связать данный watch и react.

Примечание. Некоторые команды для настройки модулей данной группы могут принимать в качестве аргумента **POSIX-совместимые** регулярные выражения. Обратите внимание, что экранирование служебных символов в POSIX происходит с помощью обратной косой черты (****), которая в DionisNX также требует экранирования. Таким образом строка "**ip access-list**", которая в POSIX-совместимом формате должна выглядеть как "**ip access\-list**", в DionisNX будет выглядеть следующим образом:

```
"ip access\\—list"
```

Символ '-' здесь экранируется, так как в POSIX-совместимых регулярных выражениях он имеет специальный смысл.

49.2.1 watch log

Данный watch осуществляет слежение за содержимым системных журналов. Содержимое журналов, подпадающее под заданные параметры, будет отправлено на указанные react.

Параметры:

- **log** - выбор системного журнала;
- **search** - параметры поиска;
- **limit** - ограничение на количество передаваемых строк.

Параметру **log** в качестве значения можно указать системный журнал:

- как файл из **log:/**;
- по имени (**messages**, **dish**, **daemon** и т.п.).

Необязательному параметру **limit** в качестве значения можно указать максимальное число строк, которые могут быть переданы на указанные react.

Параметр **search** (может быть задан несколько раз) принимает в качестве значения строку или **POSIX-совместимое** регулярное выражение для поиска по журналу.

Кроме того можно указать максимально допустимое количество строк которое может соответствовать регулярному выражению (по умолчанию поиск происходит по каждой строке отдельно). Таким образом могут быть учтены последние строки журнала, которые ранее не соответствовали регулярному выражению.

49.2.2 watch schedule

Данный watch запускает указанные react по расписанию.

Параметры:

- **at** - настройки расписания (может быть задан несколько раз).

Параметру **at** задаются поля в следующем порядке:

- минуты (0 - 59);
- часы (0 - 23);
- дата (1 - 31);
- месяц (1 - 12);
- день недели (1-7).

Каждое поле может содержать символ * означающий любое значение, а также диапазоны значений (через тире) и список значений, диапазонов (через запятую).

Примеры:

Параметр	Описание
at 5 11,0-5 * * *	- ежедневно в 11 часов 5 минут и каждый час с 0 до 5 в 5 минут
at 0 12 1 * *	- ежемесячно 1 числа в 12 часов 0 минут
at 35,45,55 17 * * 5	- еженедельно в пятницу в 17 часов 35, 45, 55 минут
at 30 20 10 1,4,7,10 *	- ежеквартально (10 числа 1, 4, 7, 10 месяца) в 20 часов 30 минут

49.2.3 watch startup

Данный watch запускает указанные react при старте системы.

49.2.4 watch space

Данный watch осуществляет слежение за свободным местом на системном диске. Строки с описанием состояния системного диска, подпадающее под заданные параметры, будут отправлены на указанные react.

Параметры:

- **free space** - проверка свободного места;
- **free ratio** - проверка свободного места в % от общего объема;
- **used space** - проверка занятого места;

- **used ratio** - проверка занятого места в % от общего объема;
- **period** - периодичность проверки.

Параметры **free space**, **used space** принимают в качестве значения:

- ключевое слово **above** (выше) или **below** (ниже);
- значение объема в KB/MB/GB/TB/PB.

Параметры **free ratio**, **used ratio** принимают в качестве значения:

- ключевое слово **above** (выше) или **below** (ниже);
- значение в % от общего объема.

Параметру **period** в качестве значения можно указать **5min**, **10min**, **30min**, **hourly**, **daily** или **weekly**.

49.2.5 watch iface

Данный watch осуществляет слежение за состоянием интерфейсов системы. Появление отслеживаемого события будет отправлено на указанные react.

Параметры:

- **iface** - выбор интерфейса;
- **event** - отслеживаемое состояние.

Параметр **event** (может быть задан несколько раз) принимает в качестве значения одно из следующих состояний:

- CREATE - появление интерфейса;
- DELETE - удаление интерфейса;
- UP - включение интерфейса;
- DOWN - выключение интерфейса;
- NO-CARRIER - отсутствие несущей на интерфейсе;
- LOWER_UP - появление несущей на интерфейсе.

49.2.6 watch react

Данный watch осуществляет слежение за результатом выполнения react модулей, выполняющих команды на локальном узле (react exec). Появление отслеживаемого результата будет отправлено на указанные react.

Параметры:

- **on-react** - выбор react модуля, результат выполнения которого будет отслеживаться;
- **search** - параметры поиска.

Параметр **search** (может быть задан несколько раз) принимает в качестве значения строку или **POSIX-совместимое** регулярное выражение для поиска по выводу результата выполнения react модуля. Так как в react exec может быть задан скрипт или несколько команд, то успешным выполнением react модуля считается успешное выполнение всех заданных команд. По умолчанию происходит поиск только для успешного выполнения react модуля. Если необходимо произвести поиск по выводу неуспешного результата выполнения, то после строки поиска необходимо задать параметр **error**.

Для примера предположим, что уже настроен react exec backup, который выполняет резервоное копирование файлов. Тогда следующая настройка позволит отследить ошибку выполнения резервного копирования, которая содержит слово “Error”.

```
adm@DionisNX(config)# service watcher
adm@DionisNX(config—watch)# watch react errbackup
adm@DionisNX(config—watch—react—errbackup)# on—react exec backup
adm@DionisNX(config—watch—react—errbackup)# search Error error
```

49.3 Модули группы react

Модули данной группы разделены на несколько типов и позволяют следующее:

- запись в журнал службы watcher (тип *log*);
- отправка e-mail сообщения (тип *mail*);
- выполнение команд на локальном узле (тип *exec*);
- выполнение команд на удаленном узле по ssh (тип *ssh*);
- отправка trap и inform сообщений на удаленный узлел (тип *trap*).

Для создания и настройки модуля react необходимо выполнить слудующую команду:

```
Router(config—service—watcher)# react <тип react> <имя react>
```

Удаление модуля react происходит командой:

```
Router(config—service—watcher)# no react <тип react> <имя react>
```

Общие параметры для react:

- **activate** - включить данный react.

49.3.1 react log

Данный react записывает полученные от watch строки в системный журнал службы watcher.

Параметры:

- **priority** - уровень сообщения (приоритет syslog).

Значение priority	Действие
alert	Послать сообщение в журнал watcher с уровнем alert
crit	Послать сообщение в журнал watcher с уровнем crit
emerg	Послать сообщение в журнал watcher с уровнем emerg
err	Послать сообщение в журнал watcher с уровнем err
info	Послать сообщение в журнал watcher с уровнем info
notice	Послать сообщение в журнал watcher с уровнем notice
warning	Послать сообщение в журнал watcher с уровнем warning

Обратите внимание, что значение **alert** удобно использовать для звукового и визуального оповещения о поступивших событиях.

49.3.2 react mail

Данный react отправляет e-mail сообщение с полученными от watch строками.

Примечание: для отправки сообщения по электронной почте необходимо предварительно настроить и активировать службу пересылки почтовых сообщений - mailer.

Параметры:

- **account** - e-mail аккаунт, сконфигурированный для службы mailer;
- **subject** - тема сообщения;
- **send-to** - e-mail адрес получателя сообщения (может быть задан несколько раз).

49.3.3 react exec

Данный react запускает на локальном узле набор команд в соответствии с параметрами. Полученные от watch строки игнорируются.

Параметры:

- **user** - локальный пользователь;
- **run** - запустить команду или скрипт.

Параметр **user** в качестве значения принимает имя локального пользователя, в контексте которого будут выполняться команды и подключаться указанный в параметре **run script** файл.

Параметр **run** (может быть задан несколько раз) в качестве значения принимает:

- ключевое слово **command** и команду (текстовая строка), которая будет выполнена на локальном узле;
- ключевое слово **script** и файл (из **file:/** или **share:/**), содержащий набор команд (строк), которые будут выполнены на локальном узле.

49.3.4 react ssh

Данный react запускает по ssh на удаленном узле набор команд в соответствии с параметрами. Полученные от watch строки игнорируются.

Для отправки команд на удаленный узел необходимо предварительно настроить ssh-соединение без использования паролей.

Параметры:

- **connect** - параметры соединения;
- **user local** - локальный пользователь;
- **user remote** - удаленный пользователь;
- **run** - запустить команду или скрипт.

Параметр **connect** в качестве значений принимает адрес и порт (по умолчанию 22) ssh сервера удаленного узла.

Параметр **user local** в качестве значения принимает имя локального пользователя, в контексте которого будут использоваться ssh ключи и указанный в параметре **run script** файл.

Параметр **user remote** в качестве значения принимает имя пользователя удаленного узла.

Параметр **run** (может быть задан несколько раз) в качестве значения принимает:

- ключевое слово **command** и команду (текстовая строка), которая будет выполнена на удаленном узле;
- ключевое слово **script** и файл (из **file:/** или **share:/**), содержащий набор команд (строк), которые будут выполнены на удаленном узле.

49.3.5 react trap

Данный react производит отправку заданных trap/inform сообщений на удаленные узлы в соответствии с настройками службы SNMP.

Для отправки trap/inform сообщений необходимо предварительно настроить правила нотификаций и активировать службу SNMP.

Параметры:

- **oid** - OID события;
- **info** - OID, описывающий дополнительную информацию;

Параметр **oid** является обязательным и задает OID который будет отправлятся в trap/inform сообщении.

Параметр **info** (может быть задан несколько раз) в качестве значения принимает:

- OID описывающий уточняющий параметр сисиемы;
- значение уточняющего параметра (Если значение не задано, оно будет подставлено автоматически).

Например:

```
adm@DionisNX(config)# service watcher
adm@DionisNX(config—service—watcher)#react trap testTrap
adm@DionisNX(config—react—trap—testTrap)# oid IF—MIB::linkDown
adm@DionisNX(config—react—trap—testTrap)# info IF—MIB::ifName ethernet50
adm@DionisNX(config—react—trap—testTrap)# info IF—MIB::ifIndex.106
adm@DionisNX(config—react—trap—testTrap)# info SNMPv2—MIB::sysName NX_WATCHER
```

50. Служба CONNSTAT

Данный сервис предназначен для сбора статистики трафика по ip-адресам, а также журналирования соединений.

Чтобы активировать сервис, необходимо выполнить команды в режиме конфигурации:

```
(config)# service connstat
(config—service—connstat)# enable
```

При этом сервис не будет производить сбор статистики и журналирование, поскольку не указаны сети в формате ip4/mask (ip6/mask), которые необходимо отслеживать.

Например, чтобы добавить для сбора статистики все адреса ip4, необходимо переконфигурировать сервис, выполнив команду:

```
(config—service—connstat)# trace 0.0.0.0/0
```

и перезапустить сервис:

```
(config—service—connstat)# disable
(config—service—connstat)# enable
```

Аналогично, для адресов ip6 и настройки журналирования тех же диапазонов адресов, необходимо выполнить:

```
(config—service—connstat)# trace6 ::/0
(config—service—connstat)# log 0.0.0.0/0
(config—service—connstat)# log6 ::/0
```

и перезапустить сервис.

Для вывода статистики используются команды в режиме enable:

```
# show ip connstat 192.168.0.1
```

При этом будет выведена статистика для адреса 192.168.0.1 в виде:

```
@#|            < rx               |              tx >
16| 8 Mb               <<<<<<<<<<< | >                  259 Kb
17| 7 Mb                 <<<<<<<<< | >                190 Kb
18| 30 Kb                      < | >                 21 Kb
19| 10 Mb           <<<<<<<<<<<<<< | >                 301 Kb
20| 5 Mb                  <<<<<<< | >                201 Kb
21| 5 Mb                  <<<<<<< | >                115 Kb
22| 2 Mb                     <<< | >                 57 Kb
23| 0 b                        |                    0 b
24| 13 Mb        <<<<<<<<<<<<<<<<<< | >                  460 Kb
25| 5 Mb                  <<<<<<< | >                120 Kb
26| 5 Mb                  <<<<<<< | >                146 Kb
27| 5 Mb                  <<<<<<< | >                187 Kb
28| 6 Mb                 <<<<<<<< | >                176 Kb
29| 7 Mb               <<<<<<<<<< | >                  223 Kb
```

30| 1011 Kb < | > 28 Kb
31| 9 Mb <<<<<<<<<<<< | > 352 Kb
32| 6 Mb <<<<<<<< | > 622 Kb
33| 2 Mb <<< | > 84 Kb
34| 7 Mb <<<<<<<<<< | > 204 Kb
35| 4 Mb <<<<<< | > 245 Kb
36| 5 Mb <<<<<<< | > 229 Kb
37| 4 Mb <<<<< | > 160 Kb
38| 13 Mb <<<<<<<<<<<<<<<<< | > 840 Kb
39| 14 Mb <<<<<<<<<<<<<<<<<< | > 684 Kb
40| 12 Mb <<<<<<<<<<<<<<< | >>> 2 Mb
41| 4 Mb <<<<<< | > 717 Kb
42| 7 Mb <<<<<<<<< | > 399 Kb
43| 4 Mb <<<<<< | > 286 Kb
44| 4 Mb <<<<< | > 153 Kb
45| 1005 Kb < | > 31 Kb
46| 0 b | 0 b
47| 603 Kb < | > 120 Kb
48| 17 Kb < | > 18 Kb
49| 3 Mb <<<<< | > 553 Kb
50| 448 Kb < | > 219 Kb
51| 8 Mb <<<<<<<<<< | > 248 Kb
52| 7 Mb <<<<<<<<<< | > 95 Kb
53| 11 Mb <<<<<<<<<<<<<< | > 752 Kb
54| 1 Mb < | > 127 Kb
55| 11 Mb <<<<<<<<<<<<<< | > 190 Kb
56| 5 Mb <<<<<<< | > 126 Kb
57| 6 Mb <<<<<<<<< | > 134 Kb
58| 9 Mb <<<<<<<<<<< | > 199 Kb
59| 3 Mb <<<< | > 80 Kb
00| 7 Mb <<<<<<<<< | > 135 Kb
01| 4 Mb <<<<<< | > 74 Kb
02| 4 Mb <<<<<< | > 75 Kb
03| 2 Mb <<<< | > 44 Kb
04| 0 b | 0 b
05| 18 Mb <<<<<<<<<<<<<<<<<<<<<<<< | > 404 Kb
06| 4 Mb <<<<<< | > 86 Kb
07| 5 Mb <<<<<<< | > 105 Kb
08| 9 Kb < | > 6 Kb
09| 12 Mb <<<<<<<<<<<<<<< | > 246 Kb
10| 8 Mb <<<<<<<<<< | > 159 Kb
11| 6 Mb <<<<<<<<< | > 132 Kb
12| 0 b | 0 b
13| 13 Mb <<<<<<<<<<<<<<<< | > 185 Kb
14| 3 Mb <<<< | > 25 Kb
15| 9 Mb <<<<<<<<<<< | > 98 Kb

В таблице представлена статистика по адресу 192.168.0.1 за последние 60 минут. В первом столбце - минуты, во втором объем входящего трафика и относительное псевдографическое представление в виде символов “<”, в третьем - аналогично второму, только для исходящего трафика и используются символы “>”.

Кроме минут, возможен вывод статистики по часам:

```
# show ip connstat 192.168.0.1 hours
```

А так же по дням, неделям и месяцам: последним параметром вместо hours должен быть либо days, либо weeks, либо months соответственно.

При необходимости можно записать статистику в файл:

```
# show ip connstat 192.168.0.1 weeks file file:/stat
```

Вывод будет продублирован в файл.

Есть возможность оценить скорость посекундно (не строго) в непрерывном режиме:

```
# show ip connstat 192.168.0.1 rate
```

В результате формируется вывод:

```
12:30:15  rx: 4 Mb (514 pkts)        tx: 102 Kb (1625 pkts)
12:30:16  rx: 0 b                    tx: 0 b
12:30:17  rx: 0 b                    tx: 0 b
12:30:18  rx: 0 b                    tx: 0 b
12:30:19  rx: 0 b                    tx: 0 b
12:30:20  rx: 1 Mb (152 pkts)        tx: 32 Kb (557 pkts)
12:30:21  rx: 2 Kb (16 pkts)         tx: 1 Kb (17 pkts)
12:30:23  rx: 940 b (7 pkts)         tx: 579 b (5 pkts)
12:30:24  rx: 1 Mb (578 pkts)        tx: 81 Kb (712 pkts)
12:30:26  rx: 4 Kb (33 pkts)         tx: 5 Kb (20 pkts)
12:30:27  rx: 0 b                    tx: 0 b
12:30:29  rx: 4 Kb (48 pkts)         tx: 2 Kb (16 pkts)
12:30:30  rx: 1 Kb (21 pkts)         tx: 4 Kb (12 pkts)
12:30:31  rx: 1 Kb (10 pkts)         tx: 1 Kb (10 pkts)
12:30:32  rx: 2 Kb (25 pkts)         tx: 28 Kb (28 pkts)
12:30:34  rx: 402 b (5 pkts)         tx: 941 b (3 pkts)
```

Для вывода журнала соединений необходимо выполнить команду вида:

```
# do show ip connlog src 192.168.0.1/24
```

В результате будет получен вывод следующего вида:

```
2020—02—10 12:38:53: ipv4 tcp 192.168.33.84:34374 —> 185.162.92.7:80
2020—02—10 12:38:53: ipv4 udp 192.168.33.84:4719 —> 192.168.33.254:53
2020—02—10 12:38:53: ipv4 udp 192.168.33.84:4719 —> 192.168.0.5:53
2020—02—10 12:38:53: ipv4 udp 192.168.33.84:4719 —> 8.8.8.8:53, nat
2020—02—10 12:38:53: ipv4 udp 192.168.33.84:49487 —> 192.168.33.254:53
2020—02—10 12:38:54: ipv4 udp 192.168.33.84:123 —> 94.247.111.10:123, nat
2020—02—10 12:38:55: ipv4 icmp 192.168.33.38:0 —> 192.168.40.225:0
2020—02—10 12:38:57: ipv4 tcp 192.168.33.15:57448 —> 192.168.33.254:22
```

Запрос журнала соединений может быть уточнён дополнительными параметрами:

src [port]	ip/порт источника соединения
dst [port]	ip/порт места назначения соединения
since	выводить соединения, произошедшие с указанного момента времени
until	выводить соединения, произошедшие до указанного момента времени
count	количество записей в выводе
proto	IP протокол
nat|!nat	Соединение NAT, либо не NAT
file	Продублировать вывод в указанный файл

51. L2TP-туннели

51.1 Введение

Dionis-NX имеет поддержку протокола L2TPv3. L2TP (англ. Layer 2 Tunnelling Protocol) — сетевой протокол туннелирования канального уровня, сочетающий в себе протокол L2F (Layer 2 Forwarding), разработанный компанией Cisco, и протокол PPTP корпорации Microsoft.

Протокол L2TP позволяет передавать пакеты PPP через TCP/IP-сеть посредством инкапсуляции PPP в L2TP в UDP.

Основные сущности протокола:

- LAC (концентратор доступа L2TP) - настраивается командой lac в интерфейсе l2tp;
- LNS (сетевой сервер L2TP) - настраивается командой lns в службе l2tp
- Виртуальный IP-адрес - это IP-адрес конца туннеля, назначаемый при его создании.

Допустим, удаленная система хочет соединиться с удаленной LAN по L2TP туннелю. Существует две схемы взаимодействия удаленной системы, LAC и LNS:

1. Удаленная система инициирует PPP-соединение с LAC через коммутируемую телефонную сеть PSTN. LAC затем прокладывает туннель для PPP-соединения через Интернет или другие сети к LNS, и таким образом осуществляется доступ к удаленной LAN.
2. Удаленная система может сама являться LAC-клиентом и участвовать в туннелировании до исходной LAN без использования отдельного LAC, если ЭВМ, содержащая программу LAC-клиента, уже имеет соединение с Интернет. Создается “виртуальное” PPP-соединение и LAC-клиент формирует туннель до LNS.

В Dionis-NX используется 2-я схема: lac-клиент в виде interface l2tp и lns в виде службы l2tp.

Сообщения протокола:

- управляющие: создание, поддержка и удаление туннеля; состоят из множества AVP(пара «атрибут-значение»)-заголовков; в Dionis-NX обрабатываются в режиме пользователя;
- информационные сообщения: передача данных по туннелю; в Dionis-NX обрабатываются в режиме пользователя или в режиме ядра (по умолчанию), зависит от настройки.

При настроенном IPSec в Dionis-NX возможна защищенная передача пакетов L2TP через туннели IPSec.

Настройка L2TP в системе Dionis-NX сводится к настройке серверной части (служба l2tp) и клиентской части (интерфейс l2tp или LAC).

Для клиентских и серверных интерфейсов l2tp возможно применение правил NAT, ACL и прочих настроек интерфейсов.

51.2 Настройка службы l2tp

Данная служба представляет собой пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов L2TP.

Формат реально создаваемого динамического интерфейса следующий: l2tp@<N>

Рассмотрим данный формат:

- l2tp@ - это тип интерфейса, знак @ говорит о том, что это серверный интерфейс.
- N - это номер серверного интерфейса

Для настройки службы l2tp выполните:

```
(config)# service l2tp
```

В службе далее необходимо создать 1 или больше lns, которая описывает серверный пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов (LAC):

```
(config—service—l2tp)# lns srv
```

Клиент lac попадет в ту lns, в которой разрешенный интервал IP-адресов клиентов будет содержать IP-адрес клиента. Настройка интервала разрешенных IP-адресов LAC будет рассмотрена далее разделе “Настройка LNS”.

В службе может быть несколько LNS, каждая со своими L2TP- и PPP-параметрами.

51.2.1 Особенности работы при IPv6-адресации.

Для клиента и сервера l2tp существует два типа IP-адресов, условно названных control и tunnel-адресами:

- control-адреса - обеспечивают первоначальную связь клиента и сервера посредством управляющего канала.
- tunnel-адреса - виртуальные адреса выделяемые сервером для клиента и принимаемые клиентом.

Каждая из команд клиента и сервера, которая задает IP адреса,принадлежит к одному из двух указанных типов.

Для правильной настройки клиента (интерфейс l2tp) и сервера (служба l2tp) необходимо, чтобы команды одного типа (control или tunnel) клиента и сервера, задавали IP-адреса одинаковой версии протокола IP (4 или 6).

К командам типа control относятся:

- служба l2tp: ipv6 control, listen, permit/deny lac-range;

- интерфейс l2tp: listen, srv;

К командам типа tunnel относятся:

- служба l2tp: ipv6 tunnel, localip, localip6, permit/deny ip-range;
- интерфейс l2tp: ipv6.

51.2.2 Глобальные настройки службы l2tp.

51.2.2.1 debug

Эта команда включает вывод в лог службы более подробную информации о ее функционировании по части L2TP протокола.

51.2.2.2 ipv6 <control | tunnel >

Эта команда устанавливает режим IPv6 адресации для контрольного канала (control) и/или для туннеля (tunnel).

Если установлен режим ipv6 control, то установление соединения с клиентом будет идти по IPv6 протоколу.

Если установлен режим ipv6 tunnel, то для выделения адресов туннелю будут использоваться IPv6 интервалы адресов (см. permit/deny ip-range).

Более подробно про настройку для IPv6 см. “Особенности работы при IPv6-адресации.”.

По умолчанию: используется IPv4 адресация.

51.2.2.3 listen <IP>

Эта команда задает IPv4- или IPv6-адрес для принятия L2TP-запросов.

По умолчанию: 0.0.0.0

51.2.2.4 port <PORT>

Эта команда задает порт для принятия и отправки L2TP-запросов.

По умолчанию: 1701

51.2.2.5 userspace

Эта команда определяет, что обработка информационных сообщений L2TP будет производится в режиме пользователя.

По умолчанию: в режиме ядра.

51.2.2.6 lns <NAME>

Эта команда создает LNS с именем NAME.

51.2.3 Общие настройки для LNS и LAC (interface l2tp).

51.2.3.1 challenge-auth

Эта команда включает использование Challenge AVP протокола L2TP для взаимной аутентификации концов туннеля посредством общего секрета (см. далее «L2TP аутентификация»).

По умолчанию: выключено.

51.2.3.2 hidden-bit

Эта команда включает скрытие поля данных в AVP, содержащих важную информацию управляющих сообщений, такую как пароль пользователя или его ID.

По умолчанию: выключено.

51.2.3.3 length-bit

Эта команда включает использование поля длины сообщения в заголовке пакета L2TP.

По умолчанию: выключено.

51.2.3.4 txspeed <VAL>

Эта команда устанавливает скорость отправления данных через туннель в значение VAL бит/сек.

По умолчанию: 10Мбит/сек.

51.2.3.5 rxspeed <VAL>

Эта команда устанавливает скорость приема данных через туннель в значение VAL бит/сек.

По умолчанию: 10Мбит/сек.

51.2.3.6 rws <VAL>

Эта команда устанавливает максимальное число входящих неподтвержденных пакетов контрольного канала.

По умолчанию: 4.

51.2.4 Настройка LNS

Далее перечислены команды,специфичные для LNS.

51.2.4.1 localip <IP>

Команда устанавливает виртуальный IP-адрес концу туннеля на стороне LNS.

51.2.4.2 localip6 <IP>

Команда устанавливает виртуальный IPv6-адрес концу туннеля на стороне LNS.

51.2.4.3 permit lac-range <IP_START> [IP_END]

Эта команда задает IP-адрес или интервал IP-адресов, от которых данному LNS разрешено принимать запросы на создание туннеля. Под проверкой попадания IP-адреса в интервал IP-адресов понимается численное сравнение IP-адреса с начальным и конечным адресами IP_START и IP_END соответственно. Таким образом, для указания максимально возможного интервала (на примере IPv4-адресов) можно указать

```
permit lac—range 0.0.0.0 255.255.255.255
```

Это будет означать, что будут приниматься запросы от на создание туннеля от любых IP-адресов.

Кроме того, есди не указано ни одного интервала и нет ни одной LNS с интервалами lac-range, то попытка соединения разрешена всем LAC.

Можно задать интервал IPv6 адресов. Адреса IPv6 будут всегда раздаваться из подсети link-local (FE80::/10). Причем для диапазона адресов выделены 2 младших 16-битных слова. Формат задания IPv6 адреса для данной команды: ::AAAA:BBBB.

51.2.4.4 deny lac-range <IP_START> [IP_END]

Эта команда задает IP-адрес или интервал IP-адресов, от которых данному LNS не разрешено принимать запросы на создание туннеля.

Можно задать интервал IPv6 адресов. Подробности см. в описании команды permit lac-range.

51.2.4.5 permit ip-range <IP_START> [IP_END]

Эта команда задает интервал виртуальных IP-адресов, которые разрешено назначать удаленному концу туннеля (L2TP клиенту).

Можно задать интервал IPv6 адресов. Подробности см. в описании команды permit lac-range.

51.2.4.6 deny ip-range <IP_START> [IP_END]

Эта команда задает интервал виртуальных IP-адресов, которые не разрешено назначать удаленному концу туннеля (L2TP клиенту).

Можно задать интервал IPv6 адресов. Подробности см. в описании команды permit lac-range.

51.2.4.7 template <*|USER> {#l2tp-templ}

Команда задает привязку шаблона интерфейса с номером NUM к имени аутнтифицированного пользователя, для которого создается туннель, либо к любому пользователю, если имя пользователя задано символом "*", в том числе к анонимному.

Команда позволяет применить различные настройки к вновь создаваемому интерфейсу, например привязать NAT или ACL, отключить multicast и другие настройки, доступные по команде ip.

Для работы данной команды необходимо создать шаблон интерфейса:

51.2.4.8 (config)# interface template-l2tp 0

В данном шаблоне можно описать различные настройки, которые будут применены к реально создаваемому серверному интерфейсу типа l2tp@.

51.2.5 Настройка PPP

Настройки опций протокола PPP задаются для interface l2tp и LNS службы l2tp,а также для других протоколов и служб динамических туннелй, например, PPTP (см. «PPTP-туннели»).

Все команды для задания PPP-опций, кроме команд для задания PPP-аутентификации, начинаются со слова ppp.

Некоторые ppp-команды возможно задать только для служб протоколов, а другие имеют смысл только для клиентских интерфейсов.

51.2.5.1 ppp debug

Эта команда включает вывод в лог службы более подробной информации о ее функционировании по части PPP протокола.

51.2.5.2 ppp compression bsd [LEV]

Эта команда включает степень сжатия данных по BSD-compresion алгоритму.

Если значение LEV установлено в 0, то сжатие не будет применяться.

По умолчанию: 0.

51.2.5.3 ppp compression deflate [LEV]

Эта команда включает степень сжатия данных по Deflate-алгоритму.

Если значение LEV установлено в 0, то сжатие не будет применяться.

По умолчанию: 0.

51.2.5.4 ppp defaultroute

Команда предписывает добавлять в систему маршрут по умолчанию на IP-адрес удаленного конца клиентского туннеля.

Если маршрут по умолчанию уже есть в системе (например, получен от DHCP-сервера), то данная команда не будет заменять его.

По умолчанию: отключено

51.2.5.5 ppp idle <N>

Эта команда задает максимальный период (в сек.) бездействия туннеля. По истечении данного периода времени туннель будет закрыт. Команда ppp redial-interval не относится к случаю закрытия по бездействию.

Доступно только для интерфейсов l2tps, pptps, pppoes.

По умолчанию: не проверять период бездействия туннеля.

51.2.5.6 ppp redial-interval <N>

Эта команда задает время ожидания после прерывания связи (отключения туннеля) перед повторной попыткой соединения с сервером. Только для клиентских интерфейсов, за исключением l2tp интерфейса, для него существует аналог этой команды - redial-interval.

По умолчанию: 30

51.2.5.7 ppp maxconnect <N>

Эта команда задает максимальный период (в сек.) активности туннеля (передачи по нему пакетов данных). По истечении данного периода времени туннель будет закрыт.

Доступно только для интерфейсов l2tps, pptps, pppoes.

По умолчанию: не проверять период работы туннеля.

51.2.5.8 ppp lcp-echo-failure <N>

Эта команда задает максимальное число LCP Echo-запросов без ответа. При превышении этого предела туннель будет закрыт.

По умолчанию: 5.

51.2.5.9 ppp lcp-echo-interval <N>

Эта команда задает интервал времени в секундах между LCP Echo-запросами.

По умолчанию: 3.

51.2.5.10 ppp mppe [stateless]

Эта команда включает использование MPPE-протокола - протокол шифрования данных, используемый поверх PPP.

Можно дополнительно включить режим stateless - без состояния.

Для работы команды ppp mppe stateless необходимо задать данную опцию как на серверном, так и на клиентском интерфейсе.

Для работы команды ppp mppe достаточно задать данную опцию только на серверном интерфейсе.

Данная опция отключает использование любых протоколов компрессии,заданных командой ppp compression.

По умолчанию: отключено.

51.2.5.11 ppp mru <VAL>

Эта команда устанавливает размер MRU (Maximum receive unit) в байтах.

По умолчанию: 1500.

51.2.5.12 ppp mtu <VAL>

Эта команда устанавливает размер MTU (Maximum transmit unit) в байтах.

По умолчанию: 1500.

51.2.5.13 ppp getdns

Эта команда предписывает запрашивать до двух адресов DNS-серверов у провайдера (серверного интерфейса).

Эту команду возможно задать только при остутвсии в конфигурации системы таких команд, как: ip resolver nameserver и ip address dhcp.

51.2.5.14 ppp getroutes

Эта команда предписывает запрашивать маршруты у провайдера (серверного интерфейса).

Передача маршрутов клиенту (до 3-х) возможна только с Radius-сервера.

ВНИМАНИЕ! Для работы данной команды оба конца туннеля должны быть под управлением системы Dionis-NX.

51.2.5.15 ppp proxyarp

Эта команда предписывает добавлять запись об IP- и MAC-адресах удаленного узла в ARP-таблицу. При этом будет полезным, если VPN-сеть разделяет адресное пространство с реальной сетью LAN.

По умолчанию: не добавлять записи.

51.2.5.16 ppp pap require

Эта команда устанавливает обязательную аутентификацию удаленного узла по PAP-протоколу.

При использовании данной команды необходимо задать секреты PAP командой pap.

По умолчанию: не требуется аутентификация узла по PAP-протоколу.

51.2.5.17 ppp chap require

Эта команда устанавливает обязательную аутентификацию удаленного узла по CHAP-протоколу.

При использовании данной команды необходимо задать секреты CHAP командой chap.

По умолчанию: не требуется аутентификация узла по CHAP-протоколу.

51.2.5.18 ppp ms-chap-v2 require

Эта команда устанавливает обязательную аутентификацию удаленного узла по MS-CHAPv2-протоколу.

при использовании данной команды необходимо задать секреты MS-CHAPv2 командой chap.

По умолчанию: не требуется аутентификация узла по MS-CHAPv2-протоколу.

51.2.5.19 ppp pap refuse

Эта команда предписывает отказать удаленному узлу в аутентификации себя по PAP-протоколу.

По умолчанию: не отказывать удаленному узлу в аутентификации себя по PAP-протоколу.

51.2.5.20 ppp chap refuse

Эта команда предписывает отказать удаленному узлу в аутентификации себя по CHAP-протоколу.

По умолчанию: не отказывать удаленному узлу в аутентификации себя по CHAP-протоколу.

51.2.5.21 ppp ms-chap-v2 refuse

Эта команда предписывает отказать удаленному узлу в аутентификации себя по MS-CHAPv2-протоколу.

По умолчанию: не отказывать удаленному узлу в аутентификации себя по MS-CHAPv2-протоколу.

51.2.5.22 ppp multilink

Эта команда включает на интерфейсе или в службе режим Multilink.

Этот режми позволяет объеденить клиентские интерфейсы, которые образуют туннели к одному и тому же удаленному узлу, в один бандл-интерфейс (далее, бандл), т.е. "пучок интерфейсов".

Режим Multilink добавляет после заголовка PPP новый промежуточный заголовок MP (Multilink-заголовок), после которого уже следует инкапсулируемый пакет. В случае успешного согласования параметров и образования бандла, данный интерфейс (клиентский, либо серверный) становится частью бандла.

Бандл управляется масетром бандла, который создается по принципу "кто первый": первый созданный туннель в режиме **ppp multilink** становится мастером бандла (основным). Остальные туннели в режиме мультилинк будут к нему присоединяться, если тоже ведут к одному и тому же удаленному узлу.

Если туннель мастера бандла останавливается по внешней причине разрыва связи (например, получения пакета LCP Terminate-Request), а не командой disable, выполненной администратором системы и существуют еще другие туннели в этом бандле, то матсер бандла будет продолжать работу в системе, периодически пытаясь восстановить соединение, для того, чтобы бандл не разрушился.

Остальные туннели, входящие в бандл, могут быть разоврваны или остановлены по любым причинам - бандл не должен разрушиться и связь не должна прекратиться в этом случае.

Для работы режима Multilink необходимо:

- задать данную команду в локальном клиентском интерфейсе (например, в **interface pptp 0**)
- задать данную команду в удаленой службе PPP интерфейсов (например, в **service pptp**)
- при наличии аутентификации, имена **ppp localname** клиентских интерфейсов, составляющих бандл, должны быть одинаковыми.

51.2.5.23 ppp mrru <VAL>

Эта команда устанавливает размер MRRU (Maximum Reconstructed receive unit) в байтах.

Это MRU, т.е. максимальный размер пакета, который может быть принят на бандл-интерфейсе для режима **ppp multilink**.

По умолчанию: 1500.

51.2.6 L2TP аутентификация

Для настройки взаимной аутентификации LNS и LAC необходимо задать в их настройках опцию challenge-auth (см. «Общие настройки для LNS и LAC» выше в данной главе), которая предписывает узлу добавлять в управляющее сообщение L2TP заголовок Challange AVP.

Далее в глобальных настройках интерфейса l2tp и в службе l2tp необходимо задать общий секрет следующей командой:

51.2.6.1 auth <LOCAL|*> <REMOTE|*> <SECRET>

Эта команда задает общий секрет SECRET для локального узла с именем LOCAL и удаленного узла с именем REMOTE.

Знак «*» - означает любой узел.

Если LAC и LNS создаются в системе Dionis-NX, то:

- для службы l2tp: LOCAL - это имя LNS; REMOTE - это имя LAC
- для интерфейса l2tp: LOCAL - это имя LAC; REMOTE - это имя LNS

Если LAC и/или LNS создаются не в системе Dionis-NX, то передаваемые имена узлов LOCAL и REMOTE могут быть, например, доменными именами соответствующих узлов.

51.2.7 Локальная PPP-аутентификация

Аутентификация уровня PPP имеет своими целями:

- задать пароли доступа клиентов к серверу L2TP, т.е. доступ LAC к LNS;
- для клиентов, прошедших аутентификацию, возможно задать IP-адрес, который будет присвоен данному клиенту при создании туннеля с ним.

Для задания паролей для алгоритмов аутентификации CHAP/MS-CHAP-V2 следует использовать команду chap, а для задания паролей для PAP следует использовать команду pap:

51.2.7.1 <chap|pap> <HOST_PASSIVE|*> <HOST_ACTIVE|*> <SECRET> [IP]

Параметры и поля:

- HOST_PASSIVE: имя узла, который должен быть аутентифицирован;
- HOST_ACTIVE: имя узла, на котором должен быть аутентифицирован HOST_PASSIVE;
- SECRET: пароль узла HOST_PASSIVE на узле HOST_ACTIVE;
- IP: адрес или сеть, который может быть назначен после успешной аутентификации; особенности:
  - имеет приоритет над адресами из permit ip-range интервала;
  - при задании активен, только если HOST_PASSIVE не равен «*».

Также необходимо сообщить службе и клиентскому интерфейсу имена узлов. Особенности задания HOST_PASSIVE и HOST_ACTIVE:

- для интерфейсов l2tp и l2pt@: имена узлов соответсвуют именам LAC (задано командой lac) и LNS (задано командой lns) соответственно;
- для всех других интерфейсов: имена узлов задаются командой ppp localname.

Знак «*» - означает любой узел.

51.2.8 Удаленная PPP-аутентификация на Radius-сервере

Данную настройку необходимо производить только в службе динамического интерфейса (например, service pptp). На клиентском интерфейсе необходимо только задать логин и пароль командами pap или chap.

Логин и пароль от клиента, как и в случае локальной PPP-аутентификации, передается по протоколу PAP, CHAP или MSCHAPv2, однако дальнейшая аутентификация и выделение IP-адреса и, возможно, других настроек осуществляется Radius-сервером.

Какой именно протокол передачи пароля использовать необходимо сообщить в настройке службе командой ppp chap/pap/mschap-v2 require.

Задать Radius-сервер аутентификации:

51.2.8.1 radius authserver <IP>[:PORT]

Задать адрес Radius-сервер аккаутинга:

51.2.8.2 radius acctserver <IP>[:PORT]

Обычно authserver и acctserver - один и тот же сервер.

Задать секрет (SECRET) для доступа к указанному Radius-серверу (IP):

51.2.8.3 radius secret <IP> <SECRET>

Можно задать таймаут соединения с Radius-сервером:

51.2.8.4 radius timeout <VAL>

Для использования Radius-аутентификации необходимо указать это командой ppp active-auth:

51.2.8.5 ppp active-auth <radius | local >

По умолчанию используется active-auth local.

51.2.9 Radius-опции для туннеля

L2TP/PPTP/PPPOE туннели могут получать от Radius-сервера и применять к себе опередленные настройки, часть из которых являются специфичными для туннелей в системе Dionis-NX.

Спеицфичные для Dionis-NX настройки необходимо задать на Radius-сервере в отдельном словаре, например dictionary.factor. В этом словаре необходимо также задать значение для VENDOR Factor-TS. Рассмтрим описание специфичных настройек словаря на Radius-сервере:

```
VENDOR          Factor—TS                   1156
BEGIN—VENDOR    Factor—TS
ATTRIBUTE       Nx—PPP—Client—Route1        40 string
ATTRIBUTE       Nx—PPP—Client—Route2        41 string
ATTRIBUTE       Nx—PPP—Client—Route3        42 string
ATTRIBUTE       Nx—PPP—Upstream—Speed—Limit   230 integer
ATTRIBUTE       Nx—PPP—Downstream—Speed—Limit 231 integer
END—VENDOR      Factor—TS
```

Если такой словарь уже существует, т.к. были ранее настроены другие подсистемы Dionisx-NX, в описании которых было рекомендовано создать данный словарь на Radius-сервере, то необходимо просто добавить строки с описанием аттрибутов в уже существующий словарь для VENDOR Factor-TS. Строки с описанием аттрибутов начинаются со слова ATTRIBUTE и задают настройки, которые Radius-сервер может предлагать клиентам, например в нашем случае туннельным интерфейсам, после аутентификации.

Кроме этого существуют и стандартные, не специфичные, настройки, такие как Framed-IP-Address, которыая задает IP-адрес для пользователя, а также Framed-Route, которая задает маршрут для серверного конца туннеля.

Рассмтрим далее какие именно настройки применяются на серверном интерфейсе, а какие на клиентском.

В указанных ниже таблицах приняты следующие обозначения в начале строки описания Radius-атрибута:

- слово "**проверка**" - означает, что данный атрибут лишь анализируется на корректность, но не производит настройку туннеля; правильное значение настройки указано в скобках;
- слово "**настройка**" - означает, что данный атрибут осуществляет настройку туннеля; в скобках могут быть указаны возможные правильные варианты значений данного атрибута или же команду Dionis-NX, которая также осуществляет данную настройку;
- словосочетание "**неиспользуемая настройка**" - означает, что данный атрибут принимается, но сооветствующие ему настройки в Dionis-NX не производятся;
- слово **VENDOR** - указывает значение для VENDOR, которому принадлежит указанный атрибут.

51.2.9.1 Настройки Radius-сервера для серверного интерфейса

На серверном конце туннеля, в случае задания данных настроек на Radius-сервере и в случае успешно прошедешей аутентификации, будут быть применены/проанализированны следующие настройки:

- Service-Type - проверка тип сервиса ("Framed-User")
- Framed-Protocol - проверка используемого туннельный протокол ("PPP")

- Session-Timeout - настройка максимального времени соединения (см. ‘ppp maxconnect’)
- Idle-Timeout - настройка максимального времени простоя соединения (см. ‘ppp idle’)
- Framed-Route - настройка маршрутов серверного конца туннеля
- Acct-Interim-Interval - настройка периодичности посылки отчетов об использовании тунеля на Radius Accounting-сервер
- Class - настройка имя класса и длину имени класса, которое посылается в отчетах на Radius Accounting-сервер
- NAS-IP-Address - настройка IP-адреса серверного конца туннеля
- Nx-PPP-Upstream-Speed-Limit - (VENDOR Factor-TS) настройка лимита скорости передачи данных
- Nx-PPP-Downstream-Speed-Limit - (VENDOR Factor-TS) настройка лимита скорости передачи данных
- MS-CHAP2-Success - (VENDOR Microsoft) внутренняя настройка сообщения от Radius-сервера при успешной аутентификации (проверка на длину сообщения = 43 и значение сообщения начиная со 2го октета должно начинаться на “S=”)
- MS-CHAP-MPPE-Keys - (VENDOR Microsoft) внутренняя настройка (расшифрование) ключа от Radius-сервера, который используется в дальнейшем в MPPE-шифровании;
- MS-MPPE-Send-Key - (VENDOR Microsoft) внутренняя настройка ключей MPPE
- MS-MPPE-Recv-Key - (VENDOR Microsoft) внутренняя настройка ключей MPPE
- MS-MPPE-Encryption-Policy - (VENDOR Microsoft) внутренняя настройка политики MPPE-шифрования (Encryption-Allowed или Encryption-Required)
- MS-MPPE-Encryption-Types - (VENDOR Microsoft) внутренняя настройка разрешенных длин ключей MPPE-шифрования (RC4-128bit-Allowed/RC4-40bit-Allowed); необходимы также настройки MS-MPPE-Encryption-Policy, а также хотя бы одна из настроек MS-MPPE-Send-Key / MS-MPPE-Recv-Key / MS-CHAP-MPPE-Keys
- MS-Primary-DNS-Server - (VENDOR Microsoft) неиспользуемая настройка первичного DNS сервера
- MS-Secondary-DNS-Server - (VENDOR Microsoft) неиспользуемая настройка вторичного DNS сервера
- MS-Primary-NBNS-Server - (VENDOR Microsoft) неиспользуемая настройка вторичного WINS сервера
- MS-Secondary-NBNS-Server - (VENDOR Microsoft) неиспользуемая настройка вторичного WINS сервера

51.2.9.2 Настройки Radius-сервера для клиентского интерфейса

На клиентском конце туннеля, в случае задания данных настроек на Radius-сервере и в случае успешно прошедешей аутентификации, будут быть применены/проанализированны следующие настройки:

- Framed-IP-Address - настройка адреса клиентского конца туннеля
- Framed-IP-Netmask - настройка маски адреса клиентского конца туннеля
- Nx-PPP-Client-Route1 - (VENDOR Factor-TS) настройка 1-го маршрута клиентского конца туннеля
- Nx-PPP-Client-Route2 - (VENDOR Factor-TS) настройка 2-го маршрута клиентского конца туннеля
- Nx-PPP-Client-Route3 - (VENDOR Factor-TS) настройка 3-го маршрута клиентского конца туннеля

51.2.9.3 Настройка пользователя на Radius-сервере

Рассмотрим пример настройки пользователя на Radius-сервер.

```
cli  Cleartext—Password := ”123”
  Service—Type = Framed—User,
    Framed—IP—Netmask = 255.255.255.0,
    Framed—IP—Address = 10.0.0.100,
    Framed—Routing = Broadcast—Listen,
    Framed—Route += ”10.0.5.0/24 10.0.0.100”,
    Framed—Route += ”10.0.4.0/24 10.0.0.100”,
    Nx—PPP—Client—Route1 = ”5.6.7.0/24 10.0.0.1”,
    Nx—PPP—Client—Route2 = ”6.6.7.0/24 10.0.0.1”,
    Nx—PPP—Client—Route3 = ”7.6.7.0/24 10.0.0.1”,
    Framed—Protocol = PPP
```

Видим, что задан пользователь cli с паролем 123 и настройками: IP-адрес и маска IP-адреса, 2 маршрута серверного конца туннеля, 3 маршрута клиентского конца туннеля.

51.3 Настройка клиентского интерфейса l2tp.

Для настройки LAC необходимо войти в режим настройки l2tp-интерфейса:

```
(config)# interface l2tp 0
```

Данный интерфейс представляет собой клиентский динамический интерфейс, появляющийся в системе при создании туннеля с LNS.

51.3.1 Команды настройки интерфейса.

После входа в режим настройки l2tp-интерфейса строка приглашения будет иметь следующий вид:

```
(config—if—l2tp0)#
```

Команды настройки интерфейса l2tp аналогичны п.«Глобальные настройки службы l2tp» кроме команды lns, listen и п.“Общие настройки для LNS и LAC.”, с учетом того, что вместо службы l2tp или LNS в описании команд следует понимать интерфейс l2tp или LAC соответственно.

Также есть специфичные для данного интерфейса команды:

51.3.1.1 bind <IP>

Эта команда задает IPv4- или IPv6-адрес от которого осуществлять соединение с сервером L2TP, и на котором принимать сообщения от L2TP сервера.

По умолчанию: 0.0.0.0

51.3.1.2 lac <NAME>

Эта команда задает имя LAC, используемое для подключение к LNS.

51.3.1.3 srv <IP[:PORT]>

Эта команда указывает IPv4- или IPv6-адрес и,возможно, порт LNS, с которому нужно создать туннель.

51.3.1.4 ipv6

Эта команда указывает, что необходимо получить IPv6-адреса туннеля, вместо IPv4-адресов.

51.3.1.5 redial-interval <TO>

Эта команда задает интервал времени в секундах между попытками повторного соединения.

По умолчанию: 30 сек.

51.3.2 Настройка PPP

См. п. 51.2.5

51.3.3 L2TP аутентификация

См. п. 51.2.6

51.3.4 PPP аутентификация

См. п. 51.2.7

51.4 Информация о работе

51.4.1 Просмотр журналов

Для просмотра журналов службы l2tp следует ввести команду:

51.4.1.1 show service l2tp log

Для просмотра журналов интерфейса l2tp следует ввести команду:

51.4.1.2 show interface log l2tp

IFNUM - номер интерфейса.

51.4.2 Информация о туннелях службы l2tp.

Для просмотра информации о подключенных пользователях к службе l2tp можно по команде

51.4.2.1 show service l2tp users

Число анонимных пользователей показано в строке “ANONYMOUS:”.

51.4.2.2 service l2tp kill <USER>

Разорвать связь с пользователем USER.

51.5 Пример настройки

51.5.1 L2TP-туннель с локальной аутентификацией и привязкой NAT-группы.

Рассмотрим пример настройки серверного и клиентского L2TP-интерфейсов.

Имеется два изделия Dionis-NX:

- одно из них - L2TP-сервер сети (LNS) с адресом 10.0.0.1/24 из сети 10.0.0.0/24; также имеются nat-list n0 и n1 с некоторыми настройками.
- другое - L2TP-концентратор доступа (LAC) L2TP с адресом 10.0.0.2/24 из сети 10.0.0.0/24.

Цель: создать L2TP VPN 192.168.1.0/24 между сервером и клиентом в виде L2TP-туннеля.

Настройка службы L2TP:

```
(config)# service l2tp
(config—service—l2tp)# lns nx1
(config—service—l2tp—nx1)# permit ip—range 192.168.1.2 192.168.1.100
(config—service—l2tp—nx1)# localip 192.168.1.1
(config—service—l2tp—nx1)# permit lac—range 10.0.0.2 10.0.0.100
(config—service—l2tp—nx1)# ppp chap requrie
(config—service—l2tp—nx1)# chap nx2 nx1 123
(config—service—l2tp—nx1)# template nx2 0
(config—service—l2tp—nx1)# template * 1
(config—service—l2tp—nx1)# enable
(config)# interface template—l2tp 0
(config—if—template—l2tp0)# ip nat—group n0
(config)# interface template—l2tp 1
(config—if—template—l2tp0)# ip nat—group n1
```

Реальные l2tp@* интерфейсы будут создаваться по мере успешного подключения L2TP-клиентов. Для клиента с именем nx2 будет создан интерфейс с привязкой NAT-списка n0. Для остальных клиентов будет привязан NAT-список n1.

Клиентский интерфейс L2TP:

```
(config)# interface l2tp 0
(config—if—l2tp0)# lac nx2
(config—if—l2tp0)# srv 10.0.0.1
(config—if—l2tp0)# chap nx2 nx1 123
(config—if—l2tp0)# enable
```

После успешного подключения реальный l2tp0 интерфейс будет создан на nx2. Кроме этого на nx1 появится интерфейс l2tp@0 (в том случае,если это 1й l2tp@ интерфейс, иначе вместо “0” может быть другой номер интерфейса).

Примечание. Если необходимо настроить туннель, в котором клиент является машиной под управлением ОС Windows, может возникнуть проблема с тем, что по умолчанию некоторые версии ОС Windows (например Windows 7), заворачивают L2TP-трафик поверх IPSEC. В связи с этим возможны 2 варианта решения данной проблемы:

- настроить IPSEC на Dionis-NX и IPSEC на Windows, используя, например, preshared-ключ;
- отключить использование IPSEC в ОС Windows: создать ключ реестра типа dword-32bit с именем ProhibitIpSec и значением 1

51.5.2 L2TP-туннель с удаленной Radius-аутентификацией

Как пример, рассмотрим PAP-аутентификацию LAC(10.0.0.2) сервером LNS(10.0.0.1) на Radius-сервере (192.168.0.1).

Настройка LAC (10.0.0.2):

```
(config)# interface l2tp 0
(config—if—l2tp0)# lac nx2
(config—if—l2tp0)# srv 10.0.0.1
(config—if—l2tp0)# chap lac1 lns1 123
(config—if—l2tp0)# enable
```

При соединении с LNS, LAC передает ему PAP логин и пароль.

Настройка LNS (10.0.0.1):

```
(config)# service l2tp
(config—service—l2tp—nx1)# lns nx1
(config—service—l2tp—nx1)# permit ip—range 192.168.1.2 192.168.1.100
(config—service—l2tp—nx1)# permit lac—range 10.0.0.2 10.0.0.2
(config—service—l2tp—nx1)# localip 192.168.1.1
(config—service—l2tp—nx1)# ppp chap requrie
(config—service—l2tp—nx1)# pap lac1 lns1 123
(config—service—l2tp—nx1)# radius authserver 192.168.0.1
(config—service—l2tp—nx1)# radius acctserver 192.168.0.1
(config—service—l2tp—nx1)# radius secret 192.168.0.1 pas
(config—service—l2tp—nx1)# enable
```

Рассмотрим,что происходит при формировании туннеля:

- При соединении с LNS, LAC передает ему CHAP логин и пароль.
- LNS осуществляет соединение с Radius-сервером, аутентифицируя себя на нем по секрету “pas”, заданному командой radius secret 192.168.0.1 pas
- после успешной аутентификации LNS по секрету “pas”, LNS передает CHAP логин и пароль на Radius-сервер для аутентификации клиента
- после успешной аутентификации LAC по секрету PAP-логину и паролю, Radius-сервер передает для LNS выделенные для туннеля параметры,такие как удаленный IP-адрес туннеля и другие параметры.

- в свою очередь LNS передает LAC окончательные параметры туннеля
- на LAC и LNS успешно создан L2TP-туннель.

52. PPTP-туннели

52.1 Введение

Dionis-NX поддерживает протокол PPTP. PPTP - это туннельный протокол типа точка-точка, позволяющий компьютеру устанавливать защищённое соединение с сервером за счёт создания специального туннеля в стандартной, незащищённой сети. PPTP инкапсулирует кадры PPP в IP-пакеты для передачи по IP-сети.

Настройка PPTP в системе Dionis-NX сводится к настройке серверной части (служба pptp) и клиентской части (интерфейс pptp).

Для клиентских и серверных интерфейсов pptp возможно применение правил NAT, ACL и прочих настроек интерфейсов.

52.2 Настройка службы pptp

Данная служба представляет собой пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов PPTP.

Формат реально создаваемого динамического интерфейса следующий: pptp@<N>

Рассмотрим данный формат:

- pptp@ - это тип интерфейса, знак @ говорит о том, что это серверный интерфейс.
- N - это номер серверного интерфейса

Для настройки службы pptp выполните:

```
(config)# service pptp
```

52.2.0.1 log

Включить ведение журналов службы pptp.

По умолчанию: отключены

52.2.0.2 listen <IP>

Эта команда задает сокет IP:1723 для принятия PPTP запросов.

По умолчанию: 0.0.0.0:1723.

52.2.0.3 localip <IP_START> <IP_END_OCTET>

Эта команда задает один или несколько IP-адресов, которые будут использоваться в качестве локальных (серверных) адресов туннеля. Причем IP_END_OCTET задает последний октет (D-октет для A.B.C.D адреса) конца интервала IP-адресов.

52.2.0.4 remoteip <IP_START> <IP_END_OCTET>

Эта команда задает один или несколько IP-адресов, которые будут использоваться в качестве удаленных (клиентских) адресов туннеля. Причем IP_END_OCTET задает последний октет (D-октет для A.B.C.D адреса) конца интервала IP-адресов.

52.2.0.5 template <*|USER>

См. п. **??**

52.2.1 Настройка PPP

См. п. 51.2.5

52.2.2 Локальная PPP-аутентификация

См. п. 51.2.7

52.2.3 Удаленная PPP-аутентификация на Radius-сервере

См. п. 51.2.8

52.2.4 Radius-опции для туннеля

См. п. 51.2.9

52.3 Настройка интерфейса pptp

Для настройки клиентского интерфейса необходимо войти в режим настройки pptp-интерфейса при помощи команды:

```
(config)# interface pptp 0
```

Данный интерфейс представляет собой динамический интерфейс, создаваемый в системе после успешного соединения с сервером PPTP.

52.3.1 Настройка интерфейса

52.3.1.1 srv <DOMAIN | IP>

Эта команда задает имя или IP-адрес PPTP-сервера, с которым нужно создать туннель.

52.3.2 Настройка PPP

См. п. 51.2.5

52.3.3 PPP аутентификация

См. п. 51.2.7

52.4 Пример настройки

Рассмотрим пример настройки службы pptp и интерфейса pptp для соединения со службой и создания туннеля pptp.

Имеются два изделия Dionis-NX:

- одно из них - сервер pptp, слушающий на адресе 192.168.32.1
- другое - клиент pptp
- Radius-сервер аутентификации с адресом 192.168.32.201

Цель: создать pptp VPN 192.168.1.0/24 между сервером и клиентом в виде pptp-туннеля.

Настройка службы pptp:

```
(config)# service—pptp
(config—service—pptp)# listen 192.168.32.1
(config—service—pptp)# localip 192.168.1.1 1
(config—service—pptp)# remoteip 192.168.1.2 100
(config—service—pptp)# ppp mschap—v2 require
(config—service—pptp)# ppp localname srv
```

```
(config—service—pptp)# radius acctserver 192.168.32.201
(config—service—pptp)# radius authserver 192.168.32.201
(config—service—pptp)# radius secret     192.168.32.201 123
(config—service—pptp)# ppp active—auth radius
(config—service—pptp)# enable
```

Настройка клиентского интерфейса pptp:

```
(config)# interface pptp 0
(config—if—pptp0)# srv 10.0.0.2
(config—if—pptp0)# chap cli srv 123
(config—if—pptp0)# ppp localname cli
(config—if—pptp0)# enable
```

Настройка шаблона интерфейса template-pptp командой template аналогична примеру для службы l2tp (см. п. 51.5.1).

53. PPPOE-туннели

53.1 Введение

Dionis-NX поддерживает протокол PPPOE. PPPOE (англ. Point-to-point protocol over Ethernet) — сетевой протокол канального уровня передачи кадров PPP через Ethernet. Предоставляет такие дополнительные возможности,как аутентификация, сжатие данных, шифрование.

Настройка PPPOE в системе Dionis-NX сводится к настройке серверной части (служба pppoe) и клиентской части (интерфейс pppoe).

Для клиентских и серверных интерфейсов pptp возможно применение правил NAT, ACL и прочих настроек интерфейсов.

53.2 Настройка службы pppoe

Данная служба представляет собой пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов PPPOE.

Формат реально создаваемого динамического интерфейса следующий: pppoe@

Рассмотрим данный формат:

- pppoe@ - это тип интерфейса, знак @ говорит о том, что это серверный интерфейс.
- N - это номер серверного интерфейса

Для настройки службы pppoe выполните:

```
(config)# service pppoe
```

53.2.0.1 ac <NAME>

Команда задает имя концентратора доступа. Клиенты могут указать в своих PPPOE-запросах, к какому именно концентратору доступа они хотят подсоединиться.

53.2.0.2 srv <NAME>

Команда задает имя одного или нескольких сервисов, предоставляемых данным концентратором доступа. Клиенты могут указать в своих PPPOE-запросах, какие именно сервисы они хотят использовать на данном концентраторе доступа.

53.2.0.3 iface <IFACE>

Эта команда задает интерфейс, на котором принимать запросы PPPOE-клиентов.

53.2.0.4 localip <IP_START>

Эта команда задает начальный IPv4- или IPv6-адрес серверного конца туннелей.

53.2.0.5 remoteip <IP_START>

Эта команда задает начальный IPv4- или IPv6-адрес клиентского конца туннелей.

53.2.0.6 frames <DISCOVERY-TYPE> <SESSION_TYPE>

Этак команда переопределяет стандартные значения типов PPPOE-фрэймов. Можно задать новое значение типа для Discovery PPPOE фрэйма (вместо стандартного 0x8863) и новое значение типа для Session PPPOE фрэйма (вместо стандартного 0x8864).

53.2.0.7 max-client-sessions <N>

Установить максимальное число сессий для одного клиента (определяемого по MAC-адресу).

По умолчанию: максимальное число сессий для всех клиентов - 65534.

53.2.0.8 radnom-sid

Назначать ID сессии клиентам из разрешенного интервала ID сессий не последовательно,а случайным образом.

53.2.1 Настройка PPP

53.2.1.1 ppp ipv6

Включить режим ipv6.

См. п. 51.2.5

53.2.2 Локальная PPP-аутентификация

См. п. 51.2.7

53.2.3 Удаленная PPP-аутентификация на Radius-сервере

См. п. 51.2.8

53.2.4 Radius-опции для туннеля

См. п. 51.2.9

53.3 Настройка интерфейса pppoe

Для настройки клиентского интерфейса необходимо войти в режим настройки pppoe-интерфейса при помощи команды:

```
(config)# interface pppoe 0
```

53.3.1 Настройка интерфейса

53.3.1.1 log

Эта команда включает ведение журналов.

По умолчанию: отключены.

53.3.1.2 ac <NAME>

Команда задает имя концентратора доступа, к которому подключаться.

53.3.1.3 srv <NAME>

Подключаться только к тем концентраторам доступа,заданным командой ac, которые предоставляют сервис с указанным именем NAME.

53.3.1.4 mac <MAC>

Подключаться только к тем концентраторам доступа, которые имеют указанный MAC-адрес.

53.3.2 Настройка PPP

53.3.2.1 ppp ipv6

Включить режим ipv6.

См. п. 51.2.5

53.3.3 PPP аутентификация

См. п. 51.2.7

53.4 Пример настройки

Рассмотрим пример настройки службы pppoe и интерфейса pppoe для соединения со службой и создания туннеля PPPOE.

Имеются два изделия Dionis-NX:

- одно из них - сервер pppoe слушающий на интерфейсе ethernet0
- другое - клиент pppoe, взаимодействующий с сервером через интерфейс ethernet0
- Radius-сервер аутентификации с адресом 192.168.32.201

Цель: создать pppoe VPN 192.168.1.0/24 между сервером и клиентом в виде pppoe-туннеля.

Настройка службы pppoe:

```
(config)# service—pppoe
(config—service—pppoe)# ac ac1
(config—service—pppoe)# srv inet
(config—service—pppoe)# iface ethernet 0
(config—service—pppoe)# localip 192.168.1.1
(config—service—pppoe)# remoteip 192.168.1.2
(config—service—pppoe)# ppp mschap—v2 require
(config—service—pppoe)# ppp localname srv
(config—service—pppoe)# radius acctserver 192.168.32.201
(config—service—pppoe)# radius authserver 192.168.32.201
(config—service—pppoe)# radius secret     192.168.32.201 123
(config—service—pppoe)# ppp active—auth radius
(config—service—pppoe)# enable
```

Настройка клиентского интерфейса pppoe:

```
(config)# interface pppoe 0
(config—if—pppoe0)# ac ac1
(config—if—pppoe0)# iface ethernet 0
(config—if—pppoe0)# srv inet
(config—if—pppoe0)# chap cli srv 123
(config—if—pppoe0)# ppp localname cli
(config—if—pppoe0)# enable
```

Настройка шаблона интерфейса template-pppoe командой template аналогична примеру для службы l2tp (см. п. 51.5.1).

54. Механизмы качества обслуживания (QoS)

Механизмы QoS Dionis-NX позволяют классифицировать проходящий через маршрутизатор трафик и применять к различным классам разную политику обслуживания. Обработке подвергается исходящий трафик интерфейса, если к нему применена политика обслуживания, которая позволяет задать:

- гарантированную полосу пропускания;
- максимальную полосу пропускания;
- приоритет;

54.1 Классификация

Для классификации трафика используются списки отображения классов (ip class-map). Каждый список представляет собой правила классификации с набором критериев отбора. При выполнении всех правил списка принимается решение о принадлежности трафика к описываемому классу.

Для создания списка отображения класса, используется команда: ip class-map <имя класса> в режиме configure.

Например:

```
DionisNX(config)# ip class—map web
DionisNX(config—cmap—web)# match tcp dport 80
DionisNX(config—cmap—web)# match tcp sport 80
```

После выполнения этих команд создается класс web, к которому будет отнесен tcp-трафик с 80 и на 80 TCP-порт.

Для работы с элементами данного списка применяется тот же подход, что и при работе со списками контроля доступа. Команда no <номер правила>|all - удаляет соответствующие элементы. Правила отбора, начинающиеся с числового префикса, добавляют правило в заданную позицию. Удаление списка осуществляется командой: no ip class-map <имя>.

Для классифицирования трафика используются два правила: match и exclude, после которых следуют критерии отбора (подмножество критериев списков контроля доступа). Список просматривается сверху вниз, при этом последовательно анализируются критерии отбора каждого правила. При выполнении критериев match трафик начинает относиться к заданному классу. При выполнении критерия exclude снимается принадлежность трафика к заданному классу. В обоих случаях, анализ дальнейших правил продолжается. В качестве критериев отбора могут быть применены значения TOS и DSCP.

Для просмотра информации о списках отображения классов используется команда: show ip class-map [имя]. Команда определена для enable-режима. Если не указано имя списка, будет показана информация о всех списках.

Например (из режима configure):

```
DionisNX(config)# do show ip class—map web
```

Несмотря на возможность создания неограниченного количества классов, количество *активных* (т.е. используемых в загруженных политиках или других подсистемах) ограничено и не может превышать 64. При превышении этого лимита на консоль и в лог выводится предупреждение, и команда, требующая активации дополнительных классов может не сработать.

54.2 Политика обслуживания policy-map

Политика обслуживания описывает то, каким образом обслуживаются различные классы трафика. Для создания/редактирования политики обслуживания необходимо выполнить команду: ip policy-map <имя политики> из режима configure.

Например:

```
DionisNX(config)# ip policy—map outworld
DionisNX(config—pmap—outworld)#
```

При этом, произойдет вход в режим редактирования политики. Каждая политика состоит из списка правил. Правило определяет качество обслуживания конкретного класса и задается в форме: class <имя класса> rate <гарантированная скорость> [другие необязательные параметры]. Для пакета, попадающего в несколько классов политики, выполнится первое правило, которому он соответствует. Для пакетов, не попадающих ни в один из классов политики, необходимо создать правило при помощи команды default rate <гарантированная скорость> [другие необязательные параметры].

В случае наличия класса A команда class A […] изменит существующее правило вместо добавления новго. Для удаления правила используется команда no class <имя класса>

Для просмотра редактируемого списка удобно воспользоваться командой: do show.

Пропускная способность описывается в правилах в виде числа с необязательным постфиксом.

Постфикс	Смысл
нет	бит в секунду
kbps	килобайт в секунду
mbps	мегабайт в секунду
kbit	килобит в секунду
mbit	мегабит в секунду
bps	байт в секунду

Основные параметры правил политики:

Название параметра	Смысл
rate <пропускная способность>	Гарантированная пропускная способность
ceil <пропускная способность>	Пиковая пропускная способность
priority <число>	Приоритет обработки (чем меньше значение – тем выше приоритет)

Если в правиле не указан параметр ceil, то пропускная способность заданного класса не будет

превышать rate, то есть отсутствие параметра ceil равносильно наличию параметра ceil, равному rate.

Для корректной работы параметра ceil на интерфейсе должна быть задана пропускная способность (команда bandwidth).

Для удаления списка политики необходимо использовать команду: no policy-map <имя> в режиме configure.

В качестве примера приведем реализацию простой приоритезации на основе tos-значений:

```
ip class—map prt0
 match tos 0/0xe0
!
ip class—map prt1
 match tos 0x20/0xe0
!
ip class—map prt2
 match tos 0x40/0xe0
!
ip class—map prt3
 match tos 0x60/0xe0
!
ip class—map prt4
 match tos 0x80/0xe0
!
ip class—map prt5
 match tos 0xa0/0xe0
!
ip class—map prt6
 match tos 0xc0/0xe0
!
ip class—map prt7
 match tos 0xe0/0xe0
!
ip policy—map prio
 class prt0 rate 1kbit ceil 10000mbit priority 7
 class prt1 rate 1kbit ceil 10000mbit priority 6
 class prt2 rate 1kbit ceil 10000mbit priority 5
 class prt3 rate 1kbit ceil 10000mbit priority 4
 class prt4 rate 1kbit ceil 10000mbit priority 3
 class prt5 rate 1kbit ceil 10000mbit priority 2
 class prt6 rate 1kbit ceil 10000mbit priority 1
 class prt7 rate 1kbit ceil 10000mbit priority 0
```

Для просмотра списка/списков политик обслуживания следует использовать команду show ip policy-map [имя], доступную из enable-режима. Например:

```
DionisNX(config)# do show ip policy—map prio
```

Если имя не задано, будут показаны все политики policy-map.

Для просмотра загруженной в ядро действующей конфигурации используется команда show ip policy [интерфейс].

Также на классы policy-group можно применить другие политики, например, class cls1 rate 1mbit red-map a, или class cls1 rate 1mbit policy-map pol1, что позволяет создавать иерархичность политик. Подробнее про поведение политик обслуживания в этом случае будет рассказано далее.

54.3 Пропускная способность интерфейса

Пропускная способность интерфейса задаётся командой bandwidth:

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# bandwidth rate 10mbit burst 150000 latency 5msec
```

Параметры команды bandwidth:

Параметр	Значение
rate	Пропускная способность
burst	Количество байт, которое может быть отправлено на неперегруженном интерфейсе до начала урезания полосы
limit	Размер очереди пакетов в байтах
latency	Размер очереди пакетов, выраженный в интервале, за который очередь опустошится при скорости rate (в этом случае limit = rate * latency)

Параметры limit и latency не могут быть заданы одновременно

54.4 Политика обслуживания prio-map

Политика обслуживания prio-map позволяет задать приоритет пакетов без явного выделения полосы пропускания для каждой из очередей. Так, если заданы три очереди, то сначала будут отправляться пакеты из первой очереди, потом если первая очередь пуста, то из второй, и только если и первая, и вторая очередь пусты, то трафик будет отправляться из третьей очереди. В политике обслуживания prio-map пакет попадает в очередь исходя из class-map’ов, в которые он входит. На очередь prio-map также можно применить другие политики обслуживания. Кроме того, можно ограничить скорость очереди при помощи параметра rate. Параметры burst, limit и peakrate соответствуют таким же параметрам в команде bandwidth.

Пример настройки:

```
ip class—map ssh
 match tcp sport 22
 match tcp dport 22
!
```

```
ip class—map http
 match tcp sport 80
 match tcp dport 80
 match tcp sport 443
 match tcp dport 443
!
ip class—map non—http
 exclude tcp sport 80
 exclude tcp dport 80
 exclude tcp sport 443
 exclude tcp dport 443
!
ip fq—codel—map fqcm
!
ip prio—map prio
 1 band class ssh rate 1mbit burst 100000 limit 1000000
 2 band class non—http fq—codel—map fqcm
 3 band class http
!
interface ethernet 0
 ip prio—group prio
!
```

В данном случае наивысший приоритет у класса ssh, а наименьший — у класса http. Прочий трафик попадает в очередь non-http, а из неё - в политику fqcm(fq-codel-map).

54.5 Политика обслуживания red-map

Политика обслуживания red-map является политикой борьбы с перегрузками, которая позволяет достичь более равномерного ограничения трафика за счёт использования вероятностного метода отбрасывания пакетов при перегрузке.

Для создания red-map используется команда ip red-map. Параметры:

Параметр	Значение
limit	Ограничение на размер очереди
avpkt	Средний размер пакета (в большинстве случаев хорошим значением является 1000)
bandwidth	Пропускная способность RED (используется для расчёта вероятности отбрасывания, RED не ограничивает пропускную способность сам по себе)
ecn	Использовать Explicit Congestion Notification (Явное уведомление о перегрузке) вместо отбрасывания пакетов при перегрузке
min	Средний размер очереди, при котором пакеты начинают отбрасываться/помечаться

Параметр	Значение
max	Средний размер очереди, при котором вероятность того, что пакеты отбросятся/пометятся, достигает максимума (0.02)
burst	Параметр, определяющий то, насколько быстро средний размер очереди следует за реальным
adaptive	Продвинутый режим, при котором максимальная вероятность отбрасывания меняется в процессе работы политики таким образом, чтобы обеспечить средний размер очереди (min + max) / 2

Параметры min, max и burst не рекомендуется менять, по умолчанию используется достаточно хорошее значение:

Параметр	Значение по умолчанию
max	limit / 4
min	max / 3
burst	(min + min + max) / (3 * avpkt)

Пример:

```
ip red—map red limit 500000 avpkt 1000 bandwidth 10mbit ecn
```

54.6 Политика обслуживания codel-map

Политика обслуживания codel-map является политикой борьбы с перегрузками, в которой вместо ограничения размера очереди ограничивается максимальное время нахождения пакета в очереди.

Параметры:

Параметр	Значение
target	Ограничение на время нахождения пакета в очереди
ecn	Использовать Explicit Congestion Notification (Явное уведомление о перегрузке) вместо отбрасывания пакетов при перегрузке

54.7 Политика обслуживания fq-codel-map

Политика, которая отличается от codel-map тем, что пакеты разделяются на некоторое количество (определяется параметром flow, по умолчанию 1024) виртуальных очередей на основе соединения, которому они принадлежат. В первую очередь отбрасываются (или помечаются при использовании ecn)

пакеты, которые принадлежат наиболее загруженному соединению. Позволяет обеспечить одновременно быстрое прохождение пакетов и отсутствие потерь на низкозагруженных соединениях без сложной настройки.

Параметры:

Параметр	Значение
target	Ограничение на время нахождения пакета в очереди
flows	Количество виртуальных очередей
ecn	Использовать Explicit Congestion Notification вместо отбрасывания пакетов при перегрузке

54.8 Политика обслуживания sfq-map

Политика sfq-map также содержит некоторое количество виртуальных очередей, и отбрасывает в первую очередь пакеты из более нагруженных соединений. В отличие от fq-codel-map, sfq-map отбрасывает пакеты при заполнении очереди, а не при превышении времени нахождения пакета в очереди. Также возможно использовать sfq-map в комбинации с RED. Параметр limit определяет максимальное количество пакетов в виртуальной очереди, параметр perturb — время, через которое меняется хэш-функция, относящая пакет к какой-либо очереди.

Для определения очереди используется хэш от IP-адреса источника, IP-адреса назначения и портов TCP/UDP.

Параметры:

Параметр	Значение
limit	Ограничение на размер виртуальной очереди (в пакетах)
perturb	Время, через которое меняется хэш-функция, относящая пакет к какой-либо очереди. Нужно для того, чтобы избежать ситуации, когда какое-либо малонагруженное соединение всегда попадает в одну очередь с высоконагруженной.
red	Использовать в качестве вложенной политики борьбы с перегрузками RED
avpkt, redflowlimit, min, max, burst	Параметры RED (redflowlimit аналогичен limit в RED). Подробнее см. политику red-map

54.9 Политика обслуживания gred-map

Политика обслуживания gred-map похожа на политику red-map, но позволяет задать несколько red-очередей с разным приоритетом. Пакет попадает в одну из 16 очередей на основе значения четырёх

битов поля DSCP. По умолчанию используются биты 2-6. При помощи параметров shift (по умолчанию равен 2) и mask можно настраивать то, какие именно биты используются (при этом биты 1-2 не могут использоваться). В случае, если необходимая очередь не задана, пакет попадает в очередь default.

Пример:

```
ip gred—map gred
 default 0
 queue 0 limit 20000 avpkt 1000 priority 0
 queue 1 limit 20000 avpkt 1000 priority 1
 queue 2 limit 20000 avpkt 1000 priority 2
 bandwidth 10000mbit
```

54.10 Ingress

Также, существует возможность ограничения входящего трафика. Для этого используется команда ingress: ingress rate <скорость>.

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ingress rate 10mbit
```

54.11 Привязка политики к интерфейсу

Политика обслуживания начинает действовать на исходящий трафик только после привязки политики к интерфейсу. Чтобы осуществить такую привязку, необходимо перейти в режим настройки интерфейса и выполнить команду: ip policy-group <имя политики>, например:

```
DionisNX(config)# interface ethernet 0
DionisNX(config—if—ethernet0)# ip policy—group prio
```

Для удаления связи с интерфейсом, необходимо выполнить команду: no ip policy-group <имя>, например:

```
DionisNX(config—if—ethernet0)# no ip policy—group prio
DionisNX(config—if—ethernet0)# do show
```

Аналогично привязываются другие политики, например, ip red-group red

Для просмотра информации о политиках на выбранном интерфейсе (или на всех интерфейсах) следует использовать команду: show ip policy [интерфейс] в enable-режиме, например:

```
DionisNX# show ip policy ethernet 0
```

54.12 Вложенные политики

Политики policy-map и prio-map позволяют применять к своим очередям другие политики. Это может быть полезно для того, чтобы задать более сложный механизм борьбы с перегрузками, чем отбрасывание при переполнении очереди.

Пример работы policy-map в комбинации с sfq-map и fq-codel-map:

```
ip sfq—map sfq
ip fq—codel—map nodelay target 2msec
ip policy—map prio
  class ssh rate 1mbit ceil 10000mbit priority 0 sfq—map sfq
  class voice rate 10mbit ceil 10000mbit priority 1 fq—codel—map nodelay
  class other rate 1kbit ceil 10000mbit priority 2
!
```

В данном примере для класса ssh будет использоваться политика борьбы с перегрузками sfq, которая будет отбрасывать пакеты из более нагруженных соединений, а для voice – fq-codel-map, что позволит обеспечить стабильную работу с небольшой задержкой для ненагруженных соединений.

Пример работы prio-map в комбинации с sfq-map и fq-codel-map:

```
ip sfq—map sfq
ip fq—codel—map nodelay target 2msec
ip prio—map prio
  band class ssh rate 1mbit sfq—map sfq
  band class voice rate 10mbit fq—codel—map nodelay
  band default
!
```

Пример аналогичен предыдущему за исключением того, что в политике prio-map классы не смогут делиться неиспользованной полосой, как в policy-map.

Пример иерархичной политики policy-map:

```
ip policy—map inner
  class ssh rate 1mbit ceil 10000mbit priority 0 sfq—map sfq
  class voice rate 10mbit ceil 10000mbit priority 1 fq—codel—map nodelay
  class other rate 1kbit ceil 10000mbit priority 2
!
ip policy—map prio
  class room1 rate 100mbit ceil 10000mbit policy—map inner
  class room2 rate 100mbit ceil 10000mbit policy—map inner
!
```

В данном примере полоса поровну разбита между двумя классами room1 и room2, и в каждом из этих классов настроена приоритизация трафика в соответствии с выше разобранным примером.

55. Расширенная статическая маршрутизация

В некоторых случаях функций статической маршрутизации может быть недостаточно. Например, когда необходимо маршрутизировать разные подсети через разные узлы. Для этого можно воспользоваться расширенным механизмом статической маршрутизации.

Внимание!!! Приоритет расширенных правил статической маршрутизации ниже, чем у правил маршрутизации, задаваемых командой ip route, поэтому, если используются расширенные правила статической маршрутизации, предварительно необходимо удалить правила ip route default. Ели этого не сделать, весь трафик будет маршрутизироваться правилами, задаваемыми ip route, так как он попадет под правило маршрута по умолчанию.

Задание маршрута IPv4 с использованием расширенных правил выполняется с помощью команды ip policy-route, например:

```
DionisNX(config)# ip policy—route src 192.168.32.0/24 gateway 192.168.33.254
```

Команда имеет следующий синтаксис:

```
[префикс] ip policy—route <правила отбора>
[class <ip class—list>] <gateway шлюз|interface сетевой интерфейс|blackhole>
```

Правила ip policy-route упорядочены, поиск маршрута осуществляется последовательно, правило за правилом. При этом, администратор может вставлять правила в произвольную позицию и удалять правила с заданной позицией, явно используя порядковый номер правила, например:

```
DionisNX(config)# ip policy—route src 192.168.32.0/24 gateway 192.168.33.254
DionisNX(config)# 1 ip policy—route icmp src 192.168.32.0/24 gateway 192.168.33.1
DionisNX(config)# do show ip policy—route config
!
1 ip policy—route icmp src 192.168.32.0/24 gateway 192.168.33.1
2 ip policy—route src 192.168.32.0/24 gateway 192.168.33.254
DionisNX(config)# no ip policy—routing 1
DionisNX(config)# do show ip policy—route config
!
1 ip policy—route src 192.168.32.0/24 gateway 192.168.33.254
```

В качестве правил отбора используются такие-же правила, как и в списках доступа (ip access-list). Кроме того, можно задать класс трафика (см. главу “Механизмы качества облуживания (QoS)”) с помощью параметра class.

Назначение трафика задается в виде IP-адреса, сетевого интерфейса или blackhole – для уничтожения трафика.

Для просмотра информации о правилах следует использовать команду show ip policy-route [config]. Для быстрого удаления всех правил следует использовать команду no ip policy-route all.

Как уже было сказано выше, при использовании ip policy-route совместно с правилами ip route, необходимо, чтобы маршрут по умолчанию также был задан с помощью ip policy-route. Для этого существует специальная команда: ip policy-route default

```
DionisNX(config)# no ip route default
DionisNX(config)# ip policy—route default gateway 192.168.33.1
```

Синтаксис команды:

```
[префикс] ip policy—route default [src <адрес/сеть>] [dst <адрес/сеть>] <gateway шлюз|interface
        сетевой интерфейс|blackhole> [priority <приоритет>]
```

Может быть задано несколько маршрутов "по умолчанию". Работа с маршрутами по умолчанию, в отличие от правил ip policy-route, ведется не по номерам, а по их описанию. Приоритет правил определяется параметром priority. Для просмотра правил "по умолчанию" используйте show ip policy-route default.

```
DionisNX(config)# ip policy—route default gateway 192.168.33.1 priority 1
DionisNX(config)# ip policy—route default gateway 192.168.22.1 priority 2
DionisNX(config)# no ip policy—route default gateway 192.168.22.1 priority 2
DionisNX(config)# do show ip policy—route default
```

Расширенные правила маршрутизации поддерживают механизм icmp-проб. При этом, соответствующий маршрут будет активным только при успешном прохождении icmp-пробы. Для задания icmp-пробы, необходимо дописать к правилу следующие аргументы:

keepalive <IP адрес> [via интерфейс] <время ожидания> [число попыток]

Например:

```
DionisNX(config)# ip policy—route gateway 192.168.33.1 keepalive 192.168.33.1 5
```

Правила ip policy-route и ip policy-route default выполняются после правил статической маршрутизации. Однако, существует возможность задать правила, которые действуют до всех остальных правил маршрутизации. Для создания такого правила, необходимо добавить параметр preroute в команде ip policy-route.

```
DionisNX(config)# ip policy—route preroute in ethernet 0 interface ethernet 1.1
```

Маршрутизация IPv6

Для задания маршрутов IPv6 (при условии, что этот протокол был ранее активирован) используйте команды с префиксом ip6. Ниже приводится таблица соответствий команд IPv4 и IPv6.

IPv6 команда	IPv4 команда
[no] ip6 policy-route	[no] ip policy-route
[no] ip6 policy-route default	[no] ip policy-route default
show ip6 policy-route [default]	show ip policy-route [default]

56. Динамическая маршрутизация

56.1 Списки

См. документацию Cisco и FRRouting.

56.2 RIP

56.2.1 Описание протокола RIP

RIP обеспечивает маршрутизацию внутри автономной системы (АС), RFC 2453 «RIP Version 2».

RIP использует UDP (порт 520) в качестве транспорта для анонсируемых маршрутов. Пакеты UDP инкапсулируются в мультикаст-датаграммы (IP-адрес: 224.0.0.9).

RIP-маршрутизатор отправляет и принимает мультикаст-датаграммы в широковещательных сетях. При широковещании, фактически, не происходит установления соседства, так как RIP-маршрутизаторы анонсируют маршруты не для конкретного соседа, а для всех.

Если сеть не поддерживает широковещание (например, NBMA), то возможно использование уникаст-датаграмм. В этом случае между RIP-маршрутизаторами необходимо установить соседские отношения.

RIP - дистанционно-векторный протокол. RIP-маршрутизаторы присваивают маршрутам метрики (дистанции), по сумме которых выбирается наилучший путь. Маршрут и метрика составляют вектор, который передается соседнему RIP-маршрутизатору в анонсе. Сосед увеличивает метрики полученного вектора на величину метрики маршрута к источнику анонса и добавляет к вектору свои маршруты с метрикой. Полученный вектор анонсируется другим соседям.

Фактически, метрика, это счетчик хопов, число маршрутизаторов на пути к цели. Напрямую подключенная сети имеет метрику 0, недостижимая сеть – метрику 16. Такой малый диапазон метрик делает RIP не пригодным для сетей большой вложенности.

56.2.2 RIP-маршрутизатор

56.2.2.1 Включение RIP-маршрутизатора (router rip)

Первое, что требуется для начала настройки RIP, это включить RIP-маршрутизатор. Сделать это можно командой:

```
(config)# router rip
```

56.2.2.2 Интерфейсы RIP-маршрутизатора (network)

Для работы RIP-маршрутизатора требуется указать, какие интерфейсы IP-маршрутизатора следует использовать. Это можно сделать либо, явно указав имя и номер интерфейса, либо, объявив сеть, настроенную на интерфейсе. Сделать так можно командой:

```
(config—rip)# network <ip/m>|<iface>
```

<ip/m> - IP-адрес и маска сети.

<iface> - имя и номер интерфейса.

Пример использования RIP-маршрутизатором интерфейса Ethernet 1 (IP-адрес 192.168.1.1).

```
!
interface ethernet 0
  ip address 192.168.0.1/24
  enable
!
interface ethernet 1
  ip address 192.168.1.1/24
  enable
!
interface ethernet 2
  ip address 192.168.2.1/24
  enable
!
```

```
(config—rip)# network 192.168.1.1/24
```

56.2.2.3 Объявление соседа (neighbor)

Для нешироковещательных сетей следует явно указать соседа. Следующая команда позволяет организовать с соседом соединение точка-точка:

```
(config—rip)# neighbor <ip>
```

<ip> - IP-адрес соседнего RIP-маршрутизатора.

56.2.2.4 Пассивный интерфейс (passive-interface)

Иногда требуется запретить рассылку анонсов с определенных интерфейсов. Сделать это можно, обозначив интерфейс пассивным при помощи команды:

```
(config—rip)# passive—interface default|<iface>
```

passive-interface default – делает все интерфейсы пассивными, в этом случае RIP-маршрутизатор только принимает анонсы.

На пассивном интерфейсе анонсы принимаются и сеть этого интерфейса анонсируется соседям. Также можно статически указать соседа командой neighbor и тогда для такого соседа анонсы будут и приниматься, и отправляться.

56.2.2.5 Версия (version)

По умолчанию RIP-маршрутизатор принимает анонсы 1-ой и 2-ой версии протокола, а отправляет только 2-ой версии. Если требуется использовать определенную версию протокола, сделать это можно командой:

```
(config—rip)# version 1|2
```

Восстановить значения по умолчанию можно командой:

```
(config—rip)# no version
```

Можно установить использование определенной версии только на выбранных интерфейсах про помощи команды:

```
(config—if—ethernet)# ip rip receive|send version 1|2
```

ip rip receive – версия для приема.

ip rip send - версия для передачи.

56.2.2.6 Избежание петель (split-horizon)

RIP избегает петель при помощи механизма «Split horizon». Суть работы split horizon в том, чтобы не отправлять анонсы о сетях в интерфейс, через который они были получены.

По умолчанию split horizon включен. Если требуется его выключить, сделать это можно командой на определенном интерфейсе:

```
(config—if—ethernet)# no ip rip split—horizon
```

Вернуть работу по умолчанию:

```
(config—if—ethernet)# ip rip split—horizon
```

Механизм split horizon можно также настроить таким образом, что RIP-маршрутизатор будет отправлять анонс о сети в тот же интерфейс, через который его получил, но при этом маршрут в эту сеть будет иметь метрику 16 «недостижим». Такое поведение можно настроить командой:

```
(config—if—ethernet)# ip rip split—horizon poisoned—reverse
```

56.2.2.7 Таймеры (timers)

RIP-маршрутизатор отправляет анонсы каждые 30 секунд, если анонс с определенным маршрутом не приходит в течение 180 секунд, маршрут помечается, как неиспользуемый, но пока остается в таблице маршрутизации, если еще через 120 секунд анонс не приходит, то маршрут удаляется.

Если требуется изменить интервалы отправки и ожидания, сделать это можно командой:

```
(config—rip)# timers basic <update> <timeout> <garb_collect>
```

<update> – интервал отправки анонсов, по умолчанию 30 секунд.

<timeout> – интервал, после которого маршрут помечается как неиспользуемый, по умолчанию 180 секунд.

<garb_collect> – интервал после которого неиспользованные маршруты удаляются, по умолчанию 120 секунд.

Вернуть значения по умолчанию можно командой:

```
(config—rip)# no timers basic
```

56.2.3 Метрики

56.2.3.1 Метрика по умолчанию (default-metric)

По умолчанию RIP-маршрутизатор присваивает перераспределенным маршрутам метрику 1. Изменить это значение можно командой:

```
(config—rip)# default—metric <n>
```

Это работает для всех маршрутов, кроме непосредственно подключенных (connected). Изменить метрику для них можно либо командой «redistribute connected metric», либо командой «offset-list».

56.2.3.2 Изменение метрики по списку (offset-list)

По умолчанию RIP-маршрутизатор увеличивает метрику маршрутов на 1. Существует механизм, позволяющий увеличивать метрику на определенное значения для маршрутов, выбранных по списку доступа. Сделать это можно при помощи команды:

```
(config—rip)# offset—list <racl_name> in|out <metric> [<iface>]
```

<racl_name> - имя списка доступа, по которому отбирать маршруты.

in|out – применять метрику к принимаемым либо отправляемым маршрутам.

<metric> - значение, на которое увеличивается метрика; по умолчанию 1.

<iface> - определенный интерфейс.

Пример увеличения метрики на 10 для входящих анонсов, содержащих маршруты в сеть 10.0.0.0/8

```
!
ip access—list myacl
  1 permit dst 10.0.0.0/8
!

(config—rip)# offset—list myacl in 10
```

56.2.3.3 Административная дистанция (distance)

AD используется для изменения приоритета путей, полученных от разных протоколов. Работает после выбора лучшего пути до помещения пути в таблицу маршрутизации. Чем меньше AD, тем приоритетнее путь. Значения AD: подключенный интерфейс 0, статический маршрут 1, EBGP 20, OSPF 110, RIP 120, IBGP 200.

Изменить административную дистанцию можно командой:

```
(config—rip)# distance <n> [<ip/m> [<racl_name>]]
```

<n> - новое значение дистанции, по умолчанию 120.

<ip/m> - префикс источника маршрута, дистанцию будет изменяться только для маршрутов, полученных от этих источников.

<racl_name> - список доступа с параметрами источников маршрута.

56.2.4 Аутентификация

Аутентификация возможна только для RIP версии 2. При использовании аутентификации следует принудительно установить версию протокола 2 при помощи команды «version» для обеспечения защиты таблицы маршрутизации. Если этого не сделать, то RIP-маршрутизатор по умолчанию будет принимать анонсы, как аутентифицированные (версии 2) так и не аутентифицированные (версии 1).

Настройка аутентификации может быть выполнена с использованием простого текстового пароля или с использованием хэшей MD5.

56.2.4.1 Простой текстовый пароль (authentication mode text)

Выбор типа аутентификации по паролю выполняется командой:

```
(config—if—ethernet)# ip rip authentication mode text
```

Текстовый пароль задается при помощи команды:

```
(config—if—ethernet)# ip rip authentication string <passw>
```

<passw> - пароль не более 16-ти смволов.

Пример настройки аутентификации с текстовым паролем:

```
!
interface ethernet 0
  ip rip authentication string secertpasswd
  ip rip authentication mode text
!
router rip
  version 2
!
```

56.2.4.2 Хэш MD5 (authentication mode md5)

Выбор типа аутентификации по хэшу выполняется командой:

```
(config—if—ethernet)# ip rip authentication mode md5
```

Для типа MD5 можно дополнительно настроить режим совместимости

```
(config—if—ethernet)# ip rip authentication mode md5 [auth—length old—ripd|rfc]
```

auth-length old-ripd - совместимость со старыми реализациями ripd.

auth-length rfc – совместимость с реализациями по RFC.

Ключи, для которых высчитывается хэш MD5, задаются командой:

```
(config—if—ethernet)# ip rip authentication key—chain <name>
```

<name> - имя связки ключей

Сами ключи создаются в режиме конфигурирования связки ключей, попасть в который можно при помощи команды:

```
(config)# router key chain <name>
```

<name> - имя связки ключей, которое будет использовано при аутентификации.

В этом режиме можно создать несколько ключей по команде:

```
(config—router—keychain—name)# key <n>
```

<n> - номер ключа.

Создав ключ, следует создать его содержимое командой:

```
(config—router—keychain—name—n)# key—string <str>
```

<str> - строка с ключом.

Дополнительно для каждого ключа можно задать свои сроки действия.

Сроки действия ключа на приём задаются при помощи команды:

```
(config—router—keychain—?—?)# accept—lifetime <HH>:<MM>:<SS> <month> <day> <year>
      infinite|(duration <secs>)|(<HH>:<MM>:<SS> <month> <day> <year>)
```

Срок действия ключа на отдачу задаются при помощи команды:

```
(config—router—keychain—?—?)\# send—lifetime <HH>:<MM>:<SS>
```

infinite|(duration)|(::)

Первый параметр <HH>:<MM>:<SS> <month> <day> <year> - соответственно час, минута, секунда, месяц, день и год начала действия срока. Конец срока можно задать в таком же виде. Либо в виде продолжительности в секундах «duration <secs>», либо бесконечным «infinite».

Пример настройки аутентификации в режиме MD5 с использованием цепочки ключей:

```
!
interface ethernet 0
  ip rip authentication key—chain mykey
  ip rip authentication mode md5
!
router rip
  version 2
!
key chain mykey
  key 1
  key—string secretpasswd
!
```

56.2.5 Анонсирование

56.2.5.1 Перераспределение (redistribute)

RIP-маршрутизатор распространяет маршруты, полученные по протоколу RIP. Помимо этих маршрутов, возможно анонсировать сети путем перераспределения маршрутов из таблицы IP-маршрутизатора в BGP-маршрутизатор. Сделать это можно командой:

```
(config—rip)# redistribute kernel|connected|static|ospf|bgp [metric <n>] [route—map <rmap_name>]
```

redistribute kernel – анонсирует маршруты, используемые ядром linux.

redistribute connected – анонсирует маршруты интерфейсов, подключенных к коммутатору.

redistribute static – анонсирует статические маршруты, т.е. прописанные вручную администратором.

redistribute ospf – анонсирует маршруты, полученные по OSPF.

redistribute bgp – анонсирует маршруты, полученные по bgp.

metric <n> - метрика, с которой будут анонсированы эти маршруты.

route-map <rmap> - анонсирует сеть с параметрами карты маршрута.

56.2.5.2 Маршрут по умолчанию (default-information originate)

Маршрут по умолчанию можно сообщить соседу и при этом его можно не создавать в таблице IP-маршрутизации. Сделать так можно командой:

```
(config—rip)# default—information originate
```

56.2.5.3 Статический маршрут (route)

RIP-маршрутизатор позволяет создавать в таблице RIP-маршрутизации статические маршруты, которые анонсируются, но не попадают в таблицу IP-маршрутизатора. Создать такой маршрут можно командой:

```
(config—rip)# route <ip/m>
```

56.2.6 Фильтрация анонсов

56.2.6.1 Списки доступа (distribute-list)

Фильтрация при помощи distribute-list использует списки доступа маршрутизатора и префиксные списки. Включить такую фильтрацию можно командой:

```
(config—rip)# distribute—list <racl_name>|(prefix <prlist_name>) in|out [\<iface\>]
```

<racl_name> - имя списка доступа маршрутизатора.

prefix <prlist_name> - имя префиксного списка.

distribute-list in – фильтрует входящие анонсы, distribute-list out – исходящие.

<iface> - название интерфейса к которому применяется список.

56.2.6.2 Карты маршрутов (route-map)

Фильтрацию при помощи карты маршрутов можно включить командой:

```
(config—rip)# route—map <rmap_name> in|out <iface>
```

<rmap_name> - имя карты маршрутов.

in|out - применяет карту к входящим либо исходящим путям.

<iface> - название интерфейса к которому применяется карта маршрутов.

56.3 RIPNG (RIP для IPv6 сети)

56.3.1 Описание протокола

Протокол обеспечивает поддержку поддержку IPv6 адресов.

Чтобы активировать службу и войти в режим настроек RIPNG, нужно ввести команду конфигурации:

```
(config)# router ripng
(config—ripng)#
```

Также существуют настройки RIPNG, относящиеся к сетевым интерфейсам. Для редактирования таких настроек необходимо войти в режим конфигурации конкретного интерфейса и ввести необходимые опции с префиксом «ip6 ripng». Например:

```
(config)# interface ethernet 0
(config—if—ethernet0)# ip6 ripng опция параметры ...
```

Процесс настройки сильно похож на RIP, только ip адрес меняется на адрес ipv6, и используется соответствующий формат адресов.

56.3.2 Доступные команды

Для более детального описания команд необходимо обратиться к разделу RIP данного руководства, а также к документации Cisco и FRRouting.

default-information originate	Сообщить соседу маршрут по умолчанию
default-metric <n>	Назначить метрику импортированным маршрутам из других протоколов маршрутизации
network <ip/m>|<iface>	Включение сети или интерфейса в работу RIPNG маршрутизатора
offset-list <racl_name> in|out <metric> [<iface>]	Настройка увеличения метрики
passive-interface	Запрет рассылки LSA с определённых интерфейсах
redistribute [metric <n>] [route-map <rmap_name>]	Импортировать маршруты из других протоколов маршрутизации
route <ip/m>	Создать статический маршрут для анонса
timers basic <update> <timeout> <garb_collect>	Настройка таймеров протокола

Конадны для диагностики маршрутизатора имеют вид:

```
# show ripng [params]
```

(Команды «show ripng …» и «show router ripng …» являются синонимами).

56.4 OSPF

В данном разделе приводится предельно краткая информация о настройке протокола динамической маршрутизации OSPF на Dionis-NX. Перед прочтением раздела администратору настоятельно рекомендуется изучить соответствующую подробную литературу о протоколе OSPF (в частности, документацию от компании Cisco).

56.4.1 Основные понятия

OSPF (Open Shortest Path First) - протокол динамической маршрутизации внутри автономной системы.

Автономная система - группа сетей и маршрутизаторов, управляющаяся одним администратором (или группой администраторов, способных договориться между собой).

Протокол OSPF основан на алгоритме Дейкстры - алгоритме нахождения кратчайшего пути. Маршрутизаторы обмениваются информацией о *состоянии каналов (link-state advertisements - LSA)*.

В OSPF вводится понятие *области (area)*. *Область* - это набор маршрутизаторов, имеющих одинаковый идентификатор области (число).

Все маршрутизаторы OSPF должны принадлежать хотя бы одной области. Автономная система должна состоять хотя бы из одной области - области 0 (ноль).

Метрика (metric) - численный показатель «стоимости» пересылки данных по каналу. Чем больше - тем хуже канал.

Стоимость маршрута (cost) - сумма метрик каналов, через которые проходит маршрут.

Административное расстояние - численный показатель, определяющий «достоверность» информации о маршруте. Чем меньше - тем достоверней. Выбирается наиболее «достоверный» маршрут. Административное расстояние имеет приоритет над стоимостью маршрута.

Идентификатор маршрутизатора (router ID - RID) - 32-битовое число, которое уникально идентифицирует маршрутизатор в пределах одной автономной системы.

Суммирование (обобщение) маршрутов - объединение адресов нескольких подсетей в один с целью уменьшения количества анонсируемых маршрутов. Например, внутриобластные сети 192.168.32.0/24 и 192.168.33.0/24 можно объединить в 192.168.32.0/23 для анонсирования в другие области.

Импорт (redistribution) маршрутов - анонсирование в среде OSPF маршрутов, полученных из других протоколов маршрутизации.

Виртуальный канал (virtual link) - механизм OSPF, позволяющий связать удалённую область с опорной через другую область. Виртуальный канал не может пролегать через тупиковые области.

Типы областей:

Область 0 (backbone area - опорная область) - область, с которой должны быть соединены все остальные области автономной системы - либо через общий маршрутизатор (ABR), либо через виртуальный канал.

Стандартная область - область, которая может граничить как с другими областями, так и с другими автономными системами.

Для уменьшения объёма таблиц маршрутизации рекомендуется использовать тупиковые области:

Стандартная тупиковая область (stub area) - область, граничащая только с другими областями (желательно с одной). Стандартная тупиковая область не может граничить с другой автономной системой. Если необходимо передать пакет в другую автономную систему - используется маршрут по умолчанию (лежащий через граничную область). Стандартная тупиковая область может принимать маршруты от других областей.

Полностью тупиковая область (totally stubby area) - область, граничащая только с одной областью. При необходимости передачи пакета в другую область или автономную систему используется маршрут по умолчанию. Маршрутизаторы полностью тупиковой области содержат только внутриобластные маршруты.

Не полностью тупиковая область (not-so-stubby area - NSSA) - стандартная тупиковая область с возможностью введения граничного маршрутизатора другой системы динамической маршрутизации (например, RIP). В данной области разрешаются анонсы LSA типа 7.

Полностью тупиковая область NSSA (NSSA no-summary) - аналогична полностью тупиковой области, но разрешены LSA типа 7.

Типы LSA, использующиеся в областях разных типов:

Тип LSA	Описание анонса	Стд. обл.	Стд. тупик.	Полн. тупик.	NSSA	NSSA no-summary
1	Внутриобластные маршруты через данный маршрутизатор	Да	Да	Да	Да	Да
2	Внутриобластные маршруты через DR	Да	Да	Да	Да	Да
3	Суммарные межобластные маршруты через ABR	Да	Да	Нет	Да	Нет
4	Суммарные маршруты через ASBR	Да	Да	Нет	Да	Нет
5	Маршруты через ASBR	Да	Нет	Нет	Нет	Нет
7	Маршруты NSSA через ABR	Нет	Нет	Нет	Да	Да

Типы маршрутизаторов:

Соседние маршрутизаторы - маршрутизаторы OSPF, находящиеся в одной сети.

Назначенный маршрутизатор (designated router - DR) - маршрутизатор, выбирающийся главным относительно остальных соседних маршрутизаторов. Все остальные соседние маршрутизаторы устанавливают с ним отношение *смежности (adjacency)*. Маршрутизатор DR принимает анонсы маршрутов от соседей и осуществляет рассылку другим соседям. DR вводится для уменьшения трафика лавинной рассылки анонсов OSPF.

Резервный назначенный маршрутизатор (backup designated router - BDR) - маршрутизатор, берущий на себя функции DR в случае отказа основного DR.

Межобластной граничный маршрутизатор (area border router - ABR) - маршрутизатор, соединяющий две (или более) областей OSPF одной автономной системы.

Граничный маршрутизатор автономной системы (autonomous system boundary router - ASBR) - маршрутизатор, граничащий с другой автономной системы.

Рекомендуемое количество устройств/маршрутизаторов/областей в автономной системе:

(На основе RFC 2329 и документации Cisco)

	Рекоменд.	Макс.
Количество соседних устройств на 1 маршрутизатор	50	100
Количество маршрутизаторов в области	< 150	350
Количество областей в автономной системе	< 25	60

Типы маршрутов:

Внутриобластные маршруты - маршруты к сетям, находящимся в пределах области. Стоимость маршрута = сумма метрик каналов.

Межобластные маршруты - маршруты к сетям, находящимся за пределами области. Стоимость маршрута = сумма метрик каналов.

Внешние маршруты E1 - маршруты к сетям, находящимся за пределами автономной системы OSPF. Стоимость маршрута = сумма метрик внутренних каналов + метрика внешнего маршрута.

Внешние маршруты E2 - маршруты к сетям, находящимся за пределами автономной системы OSPF. Стоимость маршрута = метрика внешнего маршрута. Маршрут по умолчанию в тупиковой области имеет класс E2.

56.4.2 Базовая настройка

Активация и настройка службы OSPF на узле Dionis-NX

Чтобы активировать службу и войти в режим настроек OSPF, нужно ввести команду конфигурации:

```
(config)# router ospf
(config—ospf)#
```

В режиме «config-ospf» вводятся общие настройки OSPF для данного узла Dionis-NX. Также существуют настройки OSPF, относящиеся к сетевым интерфейсам. Для редактирования таких настроек необходимо войти в режим конфигурации конкретного интерфейса и ввести необходимые опции с префиксом «ip ospf». Например:

```
(config)# interface ethernet 0
(config—if—ethernet0)# ip ospf опция параметры ...
...
```

Для отмены опций необходимо ввести соответствующую команду с префиксом «no».

Для останова службы OSPF и удаления всех настроек можно использовать команду:

```
(config)# no router ospf
```

Активация OSPF на интерфейсах, объявление сетей и областей

Чтобы узел начал выполнять функции маршрутизатора OSPF, необходимо объявить:

- Интерфейсы, участвующие в OSPF-маршрутизации;
- IP-адреса интерфейсов;
- Области OSPF, к которым подключены интерфейсы;
- Принадлежность интерфейсов к областям.

Эти функции выполняет команда «network» в режиме «config-ospf»:

```
(config—ospf)# network <iface_ip>/<mask> area <area_id>
```

Команда «network» осуществляет «привязку» сетевого(ых) интерфейса(ов) данного узла к области с номером <area_id>. Все интерфейсы, IP-адреса которых попадают в диапазон <iface_ip>/<mask>, «привязываются» к данной области. На данных интерфейсах начинается OSPF-маршрутизация, и их IP-сети анонсируются соседним маршрутизаторам.

Если не указывать специальных команд «area», то области, объявленные командой «network», считаются стандартными (не тупиковыми).

Следует помнить, что для корректной работы протокола OSPF в конкретной сети, необходима поддержка многоадресных (multicast) рассылок в данной сети, чтобы маршрутизатор мог обнаруживать соседние маршрутизаторы. Если данная сеть не поддерживает multicast, то необходимо явно указать соседние маршрутизаторы с помощью команды «neighbor» (см. ниже).

Для корректной работы OSPF необходимы согласованные настройки на соседних маршрутизаторах (совпадение идентификаторов и типов областей для соответствующих интерфейсов).

Также активировать OSPF на итерфейсе и объявить область можно с помощью команды «ip ospf area» в режиме настройки сетевого интерфейса:

```
(config—if—ethernet0)# ip ospf area <area_id> <iface_ip>
```

Примечание. Активировать OSPF на интерфейе возможно только одним способом: или с помощью команды «network» или с помощью команды «ip ospf area».

Пример минимальной настройки маршрутизатора OSPF

Допустим, необходимо настроить межобластной (ABR) маршрутизатор с 3-мя сетевыми интерфейсами:

- Интерфейс 0. Подключён к опорной области. Сеть 192.168.1.0/24;
- Интерфейс 1. Подключён к области 1. Сеть 192.168.32.0/24;
- Интерфейс 2. Подключён к области 1. Сеть 192.168.33.0/24.

Минимально необходимая настройка:

```
interface ethernet 0
 ip address 192.168.1.1/24
interface ethernet 1
 ip address 192.168.32.1/24
```

```
interface ethernet 2
 ip address 192.168.33.1/24
router ospf
 network 192.168.1.0/24 area 0
 network 192.168.32.0/23 area 0.0.0.1
```

Вторая команда «network» подключает к области 1 сразу 2 интерфейса - 1 и 2. Идентификатор области может задаваться как в виде числа, так и в четырехбайтном десятичном представлении A.B.C.D.

Явное указание соседей

Если локальная сеть не поддерживает multicast, то необходимо явно указать IP-адреса соседних маршрутизаторов с помощью команды «neighbor»:

```
(config—ospf)# neighbor <ip> [poll—interval <secs>] [priority <n>]
```

Необязательные параметры:

- poll-interval - определяет интервал, с которым будут посылаться hello-пакеты соседу, даже когда он признан «умершим». В соответствии с RFC1247 рекомендуется устанавливать это время гораздо большим, чем hello-интервал. (См. «Таймеры» ниже);
- priority - принудительная установка приоритета соседнего маршрутизатора. Приоритет влияет на выбор DR. (см. «Приоритет маршрутизатора» ниже). По умолчанию - 0 (сосед не участвует в выборах на DR).

Явное указание типа сети

В OSPF различаются следующие типы локальных сетей:

- broadcast - соединение «все-со-всеми» с возможностью multicast рассылок. Выбираются DR и BDR;
- non-broadcast - соединение «все-со-всеми» без возможности multicast рассылок (Non-Broadcast Multi-Access - NBMA). DR и BDR выбираются только на основе приоритетов;
- point-to-multipoint - соединение «один-с-остальными». Multicast не возможен. DR и BDR не выбираются;
- point-to-point - соединение «точка-точка». DR и BDR не выбираются.

В зависимости от типа физического интерфейса OSPF задаёт для него соответствующий тип сети по умолчанию. Например, для интерфейсов Ethernet устанавливается тип broadcast. Если существует необходимость изменить тип сети по умолчанию (например, если Ethernet-интерфейс не поддерживает multicast), то это можно сделать в режиме конфигурации интерфейса командой:

```
(config—if—ethernet0)# ip ospf network <тип>
```

56.4.3 ID и приоритет маршрутизатора

В автономной системе OSPF каждый маршрутизатор должен иметь свой уникальный 32-битный номер. (В частности ID маршрутизатора используется при создании виртуальных каналов). По умолчанию

маршрутизатору присваивается ID, численно равный наибольшему IP-адресу сетевых интерфейсов. Если необходимо вручную установить ID, то это можно сделать с помощью команды режима конфигурации OSPF:

```
(config—ospf)# router—id <A.B.C.D>
```

Для каждого интерфейса маршрутизатора OSPF определено понятие приоритета. Приоритет маршрутизатора - это численное значение от 0 до 255. Приоритет играет роль при выборе маршрутизаторов на роль DR и BDR (в рамках одной локальной сети). Чем выше приоритет, тем выше вероятность назначения данного маршрутизатора в качестве DR/BDR. По умолчанию, каждый интерфейс маршрутизатора имеет приоритет 1. Если требуется исключить маршрутизатор из выборов DR/BDR, то необходимо назначить ему приоритет 0.

Приоритет назначается с помощью команды конфигурации интерфейса:

```
(config—if—ethernet0)# ip ospf priority <n>
```

56.4.4 Настройка тупиковых областей

По умолчанию, объявленная область считается стандартной.

Чтобы объявить область, как стандартную тупиковую, нужно ввести опцию:

```
(config—ospf)# area <area_id> stub
```

Чтобы объявить область, как полностью тупиковую, нужно ввести опцию:

```
(config—ospf)# area <area_id> stub no—summary
```

Чтобы объявить область, как стандартную NSSA, нужно ввести опцию:

```
(config—ospf)# area <area_id> nssa [<translate_mode>]
```

Чтобы объявить область, как полностью тупиковую NSSA, нужно ввести опцию:

```
(config—ospf)# area <area_id> nssa [<translate_mode>] no—summary
```

Для NSSA-областей существует необязательная настройка «translate», которая играет роль только для межобластных (ABR) маршрутизаторов тупиковых областей, и только тогда, когда их несколько. Опция влияет на то, какой именно ABR будет транслировать LSA типа 7 в LSA типа 5 при выходе из NSSA-области. Опция может принимать значения:

- translate-candidate - значение по умолчанию. Транслирующий ABR выбирается автоматически;
- translate-always - Данный ABR всегда будет являться транслятором;
- translate-never - Данный ABR никогда не будет являться транслятором.

56.4.5 Маршрут по умолчанию

В тупиковых (stub) и полностью тупиковых (stub no-summary) областях граничным маршрутизатором ABR автоматически распространяется маршрут по умолчанию (0.0.0.0) внутрь тупиковой области, указывающий на ABR. В областях NSSA и стандартных областях иногда требуется принудительно распространить маршрут по умолчанию (например, ведущий в другую автономную систему). Это делается на маршрутизаторе ASBR командой режима конфигурации OSPF:

```
(config—ospf)# default—information originate [always] [metric <n>] [metric—type 1|2] [route—map
        <rmap_name>]
```

Необязательные параметры:

- always - всегда распространять маршрут 0.0.0.0, даже если он не определён на самом маршрутизаторе ASBR;
- metric <n> - установить значение метрики для маршрута по умолчанию;
- metric-type 1|2 - тип маршрута - E1 или E2. (По умолчанию - E2);
- route-map <name> - распространять маршрут 0.0.0.0 только в том случае, если он удовлетворяет указанной схеме (см. «Схемы маршрутов»).

56.4.6 Фильтрация маршрутов

В OSPF существует возможность фильтровать межобластные маршруты (LSA типа 3), если они по каким-то причинам не требуются в данной зоне. Фильтрация осуществляется на межобластных граничных маршрутизаторах (ABR) с помощью следующих команд (в режиме конфигурации OSPF).

Фильтрация межобластных маршрутов, анонсируемых **в** данную область, с помощью списка router ACL:

```
(config—ospf)# area <id_области> import—list <имя_или_номер_router_ACL>
```

Фильтрация межобластных маршрутов, анонсируемых **в** данную область, с помощью префиксного списка:

```
(config—ospf)# area <id_области> filter—list prefix <имя_префиксного_списка> in
```

Фильтрация межобластных маршрутов, анонсируемых **из** данной области, с помощью списка router ACL:

```
(config—ospf)# area <id_области> export—list <имя_или_номер_router_ACL>
```

Фильтрация межобластных маршрутов, анонсируемых **из** данной области, с помощью префиксного списка:

```
(config—ospf)# area <id_области> filter—list prefix <имя_префиксного_списка> out
```

О префиксных списках и списках router ACL см. раздел «Списки».

Примеры:

В следующем примере из области 10 в опорную область будут анонсированы маршруты, попадающие в диапазон от 10.10.0.0 до 10.10.255.255. Другие маршруты (например, 10.11.0.0) анонсированы не будут.

```
router access—list foo permit 10.10.0.0/16
router access—list foo deny any
router ospf
 network 192.168.1.0/24 area 0.0.0.0
 network 10.0.0.0/8 area 0.0.0.10
 area 0.0.0.10 export—list foo
```

Следующий пример аналогичен предыдущему, но реализован с помощью префиксного списка.

```
router prefix—list foo2 permit 10.10.0.0/16
router prefix—list foo2 deny any
router ospf
 network 192.168.1.0/24 area 0.0.0.0
 network 10.0.0.0/8 area 0.0.0.10
 area 0.0.0.10 filter—list prefix foo2 out
```

56.4.7 Обобщение маршрутов

Для уменьшения таблиц маршрутизации необходимо по возможности «обобщать» маршруты, анонсируемые в другие области. Например, если область содержит подсети 10.1.0.0/24, 10.1.1.0/24, 10.1.2.0/24, 10.1.3.0/24, то на межобластном маршрутизаторе ABR можно обобщить маршруты к данным сетям в один маршрут 10.1.0.0/22.

Обобщение выполняется на ABR, и применяется к LSA типа 1 и 2 (транслируются в LSA типа 3). Обобщение для типов LSA 5 и 7 не поддерживается.

Для обобщения маршрутов необходимо задать явную команду (в режиме конфигурации OSPF):

```
(config—ospf)# area <id_области> range <ip/m> [<параметры>]
```

где <ip/m> - обобщённая подсеть (из примера выше - 10.1.0.0/22).

Необязательные параметры:

- not-advertise - вместо анонсирования обобщённого маршрута в LSA типа 3 (поведение по умолчанию), маршруты, попадающие в указанную подсеть, анонсироваться во внешнюю область не будут;
- cost <n> - назначить стоимость обобщённого маршрута;
- substitute <ip2/m> - анонсировать префикс <ip2/m> вместо <ip/m>.

56.4.8 Импорт маршрутов

Чтобы импортировать маршруты из других протоколов маршрутизации в OSPF, необходимо указать опцию(и) «redistribute» (в режиме конфигурации OSPF). Формат опции:

```
(config—ospf)# redistribute <тип_маршрута> [metric <n>] [metric—type 1|2] [route—map
        <rmap_name>]
```

Для каждого типа маршрута можно указать свою опцию «redistribute».

Типы маршрутов:

- connected - маршруты, появляющиеся автоматически при назначении IP-адресов сетевым интерфейсам;
- static - принудительно назначенные статические маршруты;
- kernel - маршруты, загруженные в ядро Linux, минуя систему конфигурации Dionis-NX (на данный момент таких нет);
- bgp, rip - маршруты, создаваемые соответствующими службами динамической маршрутизации.

Параметры:

- metric <n> - назначить данным импортируемым маршрутам метрику;
- metric-type 1|2 - тип импортируемых маршрутов (E1 или E2);
- route-map <name> - применить схему маршрута к импортируемым маршрутам (для фильтрации и установки параметров). См. «Схемы маршрутов».

Также импортируемые маршруты можно отфильтровать на основе списка router ACL (см. «Списки») с помощью команды:

```
(config—ospf)# distribute—list <имя_или_номер_списка_router_acl> out <тип_маршрута>
```

56.4.9 Пассивный интерфейс

Иногда возникает необходимость запретить рассылку LSA с определённых интерфейсов. Для этого надо объявить сетевой интерфейс пассивным с помощью команды режима конфигурации OSPF:

```
(config—ospf)# passive—interface default|<интерфейс>
```

Хотя пассивный интерфейс не рассылает анонсы LSA, он всё равно может принимать анонсы от других маршрутизаторов.

Если указать параметр «default», то все интерфейсы становятся пассивными. Также можно выборочно «активизировать» несколько интерфейсов, оставив остальные пассивными. Например, допустим есть интерфейсы ethernet 0, 1, 2, 3.

```
router ospf
 passive—interface default
 no passive—interface ethernet 2
```

В данной конфигурации интерфейс 2 будет активным, а 0, 1, 3 - пассивными.

56.4.10 Метрики, стоимость, административное расстояние

Метрики интерфейсов

По умолчанию, всем интерфейсам присваивается метрика, соответствующая пропускной способности интерфейса. Чем выше пропускная способность, тем меньше метрика. Метрика 1 присваивается всем интерфейсам, чья пропускная способность >= 100 Мбит/с. Если в системе имеются более быстродействующие интерфейсы, то можно изменить формулу вычисления метрик от пропускной способности с помощью команды «auto-cost reference-bandwidth». Например:

```
(config—ospf)# auto—cost reference—bandwidth 1000
```

Данная опция указывает, что метрика 1 будет присваиваться интерфейсам с пропускной способностью >= 1000 Мбит/с. Соответственно интерфейсам с пропускной способностью 100 Мбит/с будет присвоена метрика 10.

Также можно указать явную метрику для интерфейса (в режиме конфигурации интерфейса):

```
(config—if—ethernet0)# ip ospf cost <метрика> [<ip>]
```

Параметр <ip> имеет значение, если интерфейсу назначено несколько IP-адресов.

Метрики импортированных маршрутов

Чтобы назначить метрику импортированным (redistributed) маршрутам из других протоколов маршрутизации, нужно указать опцию (в режиме конфигурации OSPF):

```
(config—ospf)# default—metric <n>
```

Стоимость маршрутов в тупиковой области

Чтобы задать стоимость суммарных маршрутов, импортируемых в тупиковую или NSSA-область, нужно указать опцию (в режиме конфигурации OSPF):

```
(config—ospf)# area <id_области> default—cost <n>
```

Административное расстояние

По умолчанию, административное расстояние для всех маршрутов OSPF равно 110. Если необходимо изменить это значение, то это можно сделать с помощью команды режима конфигурации OSPF:

```
(config—ospf)# distance <n>
```

Если требуются разные значения административного расстояния для маршрутов разных типов, то следует использовать команду:

```
(config—ospf)# distance ospf <[external <n>] [inter—area <n>] [intra—area <n>]>
```

- external <n> - AP для маршрутов за пределы автономной системы OSPF;
- inter-area <n> - AP для межобластных маршрутов;
- intra-area <n> - AP для внутреобластных маршрутов.

56.4.11 Виртуальные каналы

В автономной системе OSPF требуется, чтобы каждая область была подключена к опорной области 0. Если нет возможности подключить область к области 0 непосредственно через ABR, но область (1) граничит с другой областью (2), подключённой к опорной (0), то можно создать между областью (0) и областью (1) *виртуальный канал* через область (2).

Чтобы настроить виртуальный канал между двумя ABR, надо на обоих маршрутизаторах прописать опцию (в режиме конфигурации OSPF):

```
(config—ospf)# area <id_области> virtual—link <id_маршрутизатора> [<таймеры_ospf_для_канала>]
```

- id_области - номер области, через которую будет пролегать виртуальный канал;
- id_маршрутизатора - идентификатор противоположного маршрутизатора;
- таймеры_ospf - интервалы hello, dead, transmit, retransmit (см. «Таймеры»).

Также поддерживается режим shortcut для ABR-маршрутизаторов. См. draft-ietf-shortcut-abr-02. Следующая команда управляет режимом shortcut.

```
(config—ospf)# area <id_области> shortcut default|disable|enable
```

Для режима shortcut также необходимо указать опцию типа маршрутизатора:

```
(config—ospf)# ospf abr—type shortcut
```

56.4.12 Защита

В OSPF реализована возможность аутентификации маршрутизаторов между собой с целью исключения возможности подмены маршрутизаторов и навязывания ложных маршрутов.

Чтобы задать режим аутентификации для интерфейсов, подключённых к области, нужно указать опцию (в режиме конфигурации OSPF):

```
(config—ospf)# area <id_области> authentication [message—digest]
```

Опция «message-digest» предписывает использование алгоритма MD5. Если не указать «message-digest», то пароли будут передаваться в открытом виде.

Чтобы задать режим аутентификации для конкретного интерфейса, нужно указать опцию (в режиме конфигурации интерфейса):

```
(config—if—ethernet0)# ip ospf authentication [message—digest|null] [<ip>]
```

Опция <ip> имеет смысл, если интерфейсу назначено несколько IP-адресов.

Чтобы задать пароль для аутентификации (если не используется алгоритм MD5), следует указать опцию для интерфейса:

```
(config—if—ethernet0)# ip ospf authentication—key <пароль> [<ip>]
```

Пароли должны совпадать на всех соседних маршрутизаторах.

Чтобы задать пароль при использовании алгоритма MD5, нужно указать опцию интерфейса:

```
(config—if—ethernet0)# ip ospf message—digest—key <номер_пароля> md5 <пароль> [<ip>]
```

На всех соседних маршрутизаторах пары (номер, пароль) должны совпадать. Можно ввести несколько паролей. Это обычно делается при плановой замене ключей, чтобы каналы OSPF не прерывались.

Пример смены ключей. Допустим на узлах установлены пароли с номером 1:

```
Host1 (config)# interface ethernet 0
Host1 (config—if—ethernet0)# ip ospf message—digest—key 2 md5 NOVYPAROL

Host2 (config)# interface ethernet 0
Host2 (config—if—ethernet0)# ip ospf message—digest—key 2 md5 NOVYPAROL
Host2 (config—if—ethernet0)# no ip ospf message—digest—key 1

Host1 (config—if—ethernet0)# no ip ospf message—digest—key 1
```

Защита виртуального канала

На концах виртуального канала также можно установить взаимную аутентификацию с помощью опций (в режиме конфигурации OSPF):

```
area <id_области> virtual—link <id_маршрутизатора> authentication message—digest|null
area <id_области> virtual—link <id_маршрутизатора> authentication—key <пароль>
area <id_области> virtual—link <id_маршрутизатора> message—digest—key <номер_пароля> md5
        <пароль>
```

56.4.13 Таймеры

Для каждого сетевого интерфейса можно настроить следующие временные параметры протокола OSPF:

- hello-interval - интервал посылки hello-пакета соседним маршрутизатором. (По умолчанию - 10 с);
- retransmit-interval - время, по истечению которого маршрутизатор повторно отправит запрос соседу, если он не получил подтверждение. (По умолчанию - 5 с);
- dead-interval - время, по истечению которого сосед считается «умершим», если от него не было hello-пакетов в течение этого времени. (По умолчанию - 40 с);
- transmit-delay - время, добавляемое на пересылку анонса соседу (для медленных сетей). (По умолчанию - 1 с).

Если требуется изменить значения по умолчанию для таймеров, то это можно сделать следующей командой в режиме конфигурации соответствующего интерфейса:

```
(config—if—ethernet0)# ip ospf <тип_таймера> <секунды> [<ip>]
```

Опция <ip> имеет значение, когда интерфейсу назначено несколько IP-адресов.

Чтобы вернуть значение по умолчанию, нужно выполнить команду:

```
(config—if—ethernet0)# no ip ospf <тип_таймера> [<ip>]
```

Интервал рассылки LSA

```
(config—ospf)# refresh timer <секунды>
```

Значение по умолчанию - 10 с.

SPF throttling

В системах с часто меняющейся топологией иногда необходимо регламентировать частоту вычислений маршрутов по алгоритму SPF. Это можно настроить с помощью следующей команды режима конфигурации OSPF:

```
(config—ospf)# timers throttle spf <init_delay> <init_hold> <max_hold>
```

- init_delay - начальное время задержки вычисления SPF (в мс) после получения LSA;
- init_hold - последующая задержка вычисления SPF после получения LSA (в мс). Удваивается каждый раз (до max_hold);
- max_hold - максимальное значение последующей задержки (в мс).

Если обновления топологии приходят часто, то вычисление SPF будет регламентировано следующим образом:

init_delay, init_hold, 2*init_hold, 4*init_hold, …, max_hold, max_hold, …

56.4.14 Тонкие настройки OSPF

Включение/выключение поддержки Opaque LSA (RFC 2370)

```
(config—ospf)# [no] capability opaque
```

Синоним:

```
(config—ospf)# [no] ospf opaque—lsa
```

Включение/выключение совместимости с RFC 1583

```
(config—ospf)# [no] compatible rfc1583
```

Синоним:

```
(config—ospf)# [no] ospf rfc1583compatibility
```

Тупиковый маршрутизатор (RFC 3137)

Если объявить маршрутизатор, как тупиковый, то он будет анонсировать маршруты к себе с бесконечной метрикой, и другие маршрутизаторы будут стараться не прокладывать маршруты через него.

Объявление тупикового маршрутизатора:

```
(config—ospf)# max—metric router—lsa <режим>
```

Возможные режимы:

- administrative - объявить маршрутизатор тупиковым немедленно и на неопределённое время (до отмены);
- on-startup <секунды> - объявлять тупиковым после старта системы на заданное время;
- on-shutdown <секунды> - после выполнения команды «no router ospf» не сразу останавливать службу OSPF, но объявлять маршрутизатор тупиковым на заданное время, а потом останавливать.

Команда «no» отменяет режим тупикового маршрутизатора:

```
(config—ospf)# no max—metric router—lsa <режим>
```

Типы маршрутизаторов ABR (RFC 3509)

Опция «ospf abr-type» настраивает поведение ABR маршрутизатора согласно RFC 3509 и draft-ietf-ospf-shortcut-abr-02.

```
(config—ospf)# ospf abr—type cisco|ibm|shortcut|standard
```

Игнорирование несовпадения MTU

По умолчанию в OSPF включена проверка совпадения MTU между соседними маршрутизаторами. В случае несовпадения не будет установлено отношение смежности. Если требуется отключить данную проверку, то это можно сделать опцией в режиме конфигурации интерфейса:

```
(config—if—ethernet0)# ip ospf mtu—ignore [<ip>]
```

Параметр <ip> имеет смысл, если интерфейсу назначено несколько IP-адресов.

Включение поддержки протокола BFD

```
(config—if—ethernet0)# ip ospf bfd
```

56.4.15 Диагностика

Следующие команды привилегированного режима выводят различную диагностическую информацию о OSPF.

(Команды «show ospf …» и «show router ospf …» являются синонимами).

show log router [all|follow|number <n>| search <REGEX>]	Вывод журнала служб динамической маршрутизации
show ip route	Вывод всей таблицы маршрутизации узла (всех протоколов)
show ospf	Вывод краткой информации о состоянии службы OSPF
show ospf route	Вывод таблицы маршрутизации для OSPF

show ospf border-routers	Вывод только внешних и межобластных маршрутов
show ospf interface <iface>	Информация об интерфейсе (с точки зрения OSPF)
show ospf neighbor [all|<ip>|(<iface> [detail])|detail]	Информация о соседнем(их) маршрутизаторе(ах)

Команды вывода базы данных OSPF:

show ospf database	Краткая информация о базе данных OSPF
show ospf database router	Информация о LSA type 1
show ospf database network	Информация о LSA type 2
show ospf database summary	Информация о LSA type 3
show ospf database asbr-summary	Информация о LSA type 4
show ospf database external	Информация о LSA type 5
show ospf database nssa-external	Информация о LSA type 7
show ospf database opaque-link	Информация о LSA type 9
show ospf database opaque-area	Информация о LSA type 10
show ospf database opaque-as	Информация о LSA type 11
show ospf database max-age	Информация о LSA, находящихся в списке MaxAge

Почти ко всем командам «show ospf database» применимы дополнительные параметры:

- self-originate - показать только те данные, источником которых является данный маршрутизатор;
- adv-router <ip> - показать только те данные, источником которых является указанный маршрутизатор.

Если необходима более подробная информация об изменении отношений смежности с соседними маршрутизаторами, то нужно указать опцию в режиме конфигурации OSPF:

```
(config—ospf)# log—adjacency—changes [detail]
```

Информация об изменениях отношений смежности будет протоколироваться в журнале служб динамической маршрутизации.

56.5 OSPF6 (OSPFv3 для IPv6 сети)

56.5.1 Описание протокола

Протокол обеспечивает поддержку IPv6 адресов.

Чтобы активировать службу и войти в режим настроек OSPFv3, нужно ввести команду конфигурации:

```
(config)# router ospf6
(config—ospf6)#
```

В режиме «config-ospf6» вводятся общие настройки OSPFv3 для данного узла Dionis-NX. Также существуют настройки OSPFv3, относящиеся к сетевым интерфейсам. Для редактирования таких настроек необходимо войти в режим конфигурации конкретного интерфейса и ввести необходимые опции с префиксом «ip6 ospf6». Например:

```
(config)# interface ethernet 0
(config—if—ethernet0)# ip6 ospf6 опция параметры ...
```

Процесс настройки сильно похож на OSPFv2, только ip адрес меняется на адрес ipv6, и используется соответствующий формат адресов. Кроме того необходимо включить OSPFv3 на интерфейсе с помощью команды **iface**:

```
(config—ospf6)# iface <interface—type> <interface—num> area <id_области>
```

Все IPv6-адреса, назначенные интерфейсу, участвуют в OSPFv3. Поскольку в hello-пакетах в качестве отправителя указывается локальный адрес интерфейса (link-local address), соседская связь будет сформирована, даже если сосед сконфигурирован с другим префиксом.

56.5.2 Доступные команды

Для более детального описания команд необходимо обратиться к разделу OSPF данного руководства, а также к документации Cisco и FRRouting.

area <id_области> [params]	Объявление зоны
auto-cost reference-bandwidth	Настройка вычисления метрик в зависимости от пропускной способности канала
log-adjacency-changes [detail]	Более подробное протоколирование информации об изменениях отношений смежности в журнале служб динамической маршрутизации.
router-id <A.B.C.D>	ID маршрутизатора
redistribute <тип_маршрута> [route-map]	Импорт маршруов из других протоколов маршрутизации в OSPFv3
timers throttle spf <init_delay> <init_hold> <max_hold>	Частота вычислений маршрутов по алгоритму SPF
stub-router administrarive	Отключение режима транзитного маршрутизатора

Конадны для диагностики маршрутизатора имеют вид:

```
# show ospf6 [params]
```

(Команды «show ospf6 …» и «show router ospf6 …» являются синонимами).

56.6 BGP

56.6.1 Описание протокола BGP

BGP обеспечивает маршрутизацию без петель между автономными системами (AS), RFC 4271 «A Border Gateway Protocol 4 (BGP-4)». Маршрутизаторы используют протоколы внутренней маршрутизации (IGP) внутри AS, а вне AS - протокол BGP.

Когда BGP работает между маршрутизаторами в одной AS, это называется внутренний BGP (IBGP). Когда BGP работает между маршрутизаторами, которые принадлежат к разным AS, это называется внешний BGP (EBGP). BGP использует TCP в качестве транспорта (порт 179). Два BGP-маршрутизатора устанавливают TCP-соединения между собой. Такие маршрутизаторы называются соседними маршрутизаторам (соседями).

Соседи обмениваются информацией о путях (BGP-анонсами). BGP-путь, это набор номеров AS, которые следует пройти к сети назначения.

BGP-пути хранятся в трех BGP-таблицах: Adj-RIB-In (Adjacent Routing Information Base, Incoming), Loc-RIB (Local Routing Information Base) и Adj-RIB-Out (Adjacent Routing Information Base, Outgoing). Получив анонс от соседа, BGP помещает пути из него в таблицу Adj-RIB-In, затем обрабатывает их в соответствии с политиками и перемещает в таблицу Loc-RIB. Также в Loc-RIB хранятся локальные пути, которые настроены администратором, и пути, которые перераспределены из маршрутов таблицы IP-маршрутизации. Пути, предназначенные для анонса соседям, BGP помещает из таблицы Loc-RIB в таблицу Adj-RIB-Out.

BGP выбирает лучший путь из Loc-RIB, сравнивает административную дистанцию (AD), между остальными путями в ту же сеть, но полученными из других протоколов, например OSPF, RIP. Путь с наименьшей AD помещаются в таблицу IP-маршрутизатора.

56.6.2 BGP-маршрутизатор

56.6.2.1 Включение BGP-маршрутизатора (router bgp)

Первое, что требуется для начала настройки BGP, это включить BGP-маршрутизатор. Сделать это можно командой:

```
(config)# router bgp <AS>
```

<AS> - номер AS, в которую входит настраиваемый BGP-маршрутизатор.

После ввода команды система переходит в режим конфигурирования BGP, в котором вводятся остальные команды настройки BGP-маршрутизатора.

Пример включения BGP-маршрутизатора, который входит в AS с номером 65001.

```
(config)# router bgp 65001
(config—bgp—65001)#
```

Выключить BGP-маршрутизатор можно командой:

```
(config)# no router bgp <AS>
```

Конфигурация BGP-маршрутизатора при этом удаляется.

Для настройки маршрутизатора доступны несколько семейств адресов. Перейти к конкретному семейству адресов можно командой:

```
(config—bgp—65001)# address—family ipv4|ipv6 unicast|multicast
```

56.6.2.2 Идентификатор BGP-маршрутизатора (router-id)

При включении BGP-маршрутизатора создается идентификатор. Это IP-адрес, который совпадает с максимальным IP-адресом интерфейсов маршрутизатора. Если в маршрутизаторе не существует ни одного интерфейса, то идентификатору BGP-маршрутизатора присваивается значение 0.0.0.0 и его необходимо изменить вручную. Изменить идентификатор можно командой:

```
(config—bgp—65001)# bgp router—id <RID>
```

<RID> - IP-адрес, который будет идентифицировать BGP-маршрутизатор. Пример использования IP-адреса 192.168.1.1 в качестве идентификатора.

```
(config—bgp—65001)# bgp router—id 192.168.1.1
```

Удалить заданный идентификатор и вернуться к выбору по умолчанию можно командой:

```
(config—bgp—65001)# no bgp router—id
```

56.6.3 BGP-соединение

56.6.3.1 Создание BGP-соединения (neighbor)

BGP-соединение устанавливается между двумя соседями. Для установления BGP-соединения у обоих соседей в конфигурации должна быть команда:

```
(config—bgp—65001)# neighbor <ip>|<group> remote—as <AS>
```

<ip> - IP-адрес соседнего BGP-маршрутизатора. Для EBGP-соединения, этот адрес должен быть в непосредственно подключенной сети (можно отключить такую проверку командой disable-connected-check). Для IBGP-соединения, к этому адресу должен существовать маршрут.

<group> - имя группы BGP-маршрутизаторов.

<AS> - номер AS, в которую входит соседний BGP-маршрутизатор или группа.

Пример настройки EBGP-соединения между соседом 1 из AS 65001 с IP-адресом 192.168.1.1 и соседом 2 из AS 65002 с IP-адресом 192.168.1.2:

```
Сосед—1(config—bgp—65001)# neighbor 192.168.1.2 remote—as 65002

Сосед—2(config—bgp—65002)# neighbor 192.168.1.1 remote—as 65001
```

Проверить возможность установления соединения можно пингом:

```
Сосед—1# ping 192.168.1.2 source 192.168.1.1
```

56.6.3.2 Удаление и административное выключение BGP-соединения (shutdown)

Удалить настройку BGP-соединения можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> remote—as
```

При этом удаляются все остальные настройки, связанные с этим соседом. Для временной блокировки лучше использовать команду административного выключения:

```
(config—bgp—65001)# neighbor <ip>|<group> shutdown
```

Включить соседа обратно можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> shutdown
```

56.6.3.3 Группы BGP-маршрутизаторов (peer-group)

Несколько BGP-маршрутизаторов можно объединить в единую группу, если они находятся в одной AS. В этом случае они будут использовать общие настройки группы. Создать группу можно командой:

```
(config—bgp—65001)# neighbor <group> peer—group
```

<group> - имя группы. Добавить BGP-маршрутизатор в группу можно командой:

```
(config—bgp—65001)# neighbor <ip> peer—group <group>
```

<ip> - IP-адрес соседа.

Пример создания группы mybgpgroup из трех BGP-маршрутизаторов, находящихся в AS 65003. Перед добавлением BGP-маршрутизаторов в группу следует сначала определить для группы номер удаленной AS, а затем добавить узлы.

```
(config—bgp—65001)# neighbor mybgpgroup peer—group
(config—bgp—65001)# neighbor mybgpgroup remote—as 65003
(config—bgp—65001)# neighbor 192.168.3.1 peer—group mybgpgroup
(config—bgp—65001)# neighbor 192.168.3.2 peer—group mybgpgroup
(config—bgp—65001)# neighbor 192.168.3.3 peer—group mybgpgroup
```

56.6.3.4 Использование dummy-интерфейса (update-source)

При использовании dummy-интерфейса, BGP-маршрутизатор сообщает соседу IP-адрес, который не принадлежит ни одному физическому интерфейсу и, следовательно, не зависит от его состояния (dummy-интерфейс всегда активен). Для использования dummy-интерфейса нужно выполнить команду:

```
(config—bgp—65001)# neighbor <ip>|<group> update—source <ip—iface>|<iface>
```

<ip-iface> - IP-адрес dummy-интерфейса.

<iface> - имя dummy-интерфейса.

Пример использования dummy-интерфейса с именем dummy0 и IP-адресом 192.168.1.1 в качестве источника BGP-обновлений.

```
(config)# interface dummy 0
(config—if—dummy0)# ip address 192.168.1.1/24
(config—if—dummy0)# enable
(config—bgp—65001)# neighbor 195.220.1.2 update—source dummy 0
```

Отменить использование dummy-интерфейса можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> update—source
```

56.6.3.5 Включение обязательных политик фильтрации для EBGP сессии

Данная команда требует наличия входящих и исходящих правил фильтрации для EBGP сессии:

```
(config—bgp—65001)# bgp ebgp—requires—policy
```

При отсутствии фильтров маршруты не будут приниматься и/или анонсироваться. Отключить обязательное наличие правил фильтрации можно следующей командой:

```
(config—bgp—65001)#no bgp ebgp—requires—policy
```

56.6.3.6 Отмена проверки подключенной сети для EBGP (disable-connected-check, enforce-multihop)

При EBGP-соединении BGP-маршрутизатор ищет соседа в одной из подключенных сетей. Например, при использовании dummy-интерфейса, который никуда не подключен, следует отключать такую проверку, либо использовать команду ebgp-multihop. Отключить проверку можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> disable—connected—check
```

У этой команды существует синоним, команда enforce-multihop, при выполнении которой в конфигурацию все равно запишется disable-connected-check.

56.6.3.7 Настройка TTL для EBGP (ebgp-multihop, ttl-security hops)

Допустим EBGP-соседи не находятся в одной сети. Тогда для установления BGP-соединения следует указать, что для достижения соседа требуется проходить несколько маршрутизаторов. Сделать это можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> ebgp—multihop [<hop_count>]
```

<hop_count> - не обязательный параметр устанавливает TTL в IP-датаграмме.

Пример увеличения TTL для соседа с IP-адресом 195.220.1.2.

```
(config—bgp—65001)# neighbor 195.220.1.2 ebgp—multihop 128
```

Отменить изменение TTL можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> ebgp—multihop
```

Иногда, в целях безопасности, требуется явное указание расстояния до EBGP-соседа. Сделать это можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> ttl—security hops <n>
```

<n> - чисто маршрутизаторов на пути к соседу.

Внимание! Команда применима только для EBGP и при использовании заменяет команду neighbor ebgp-multihop. Запрещено одновременное использование с neighbor ebgp-multihop.

Пример контроля TTL для соседа с IP-адресом 195.220.1.1. IP-датаграммы, у которых TTL меньше 253, будут сброшены.

```
(config—bgp—65001)# neighbor 192.168.1.1 ttl—security hops 2
```

Отменить изменение TTL можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> ttl—security hops
```

56.6.3.8 Контроль линка для EBGP (fast-external-failover)

В BGP-маршрутизаторе существует механизм, который немедленно обрывает BGP-соединение, если на интерфейсе пропало соединение. Этот механизм применяется только для EBGP-соединения и он включен по умолчанию. Если требуется отменить такое поведение, то сделать это можно командой:

```
(config—bgp—65001)# no bgp fast—external—failover
```

После отмены вместо механизма обнаружения падения соединения будут использоваться таймеры hold и keepalive. Таким образом, если пропадет линк, BGP-соединение еще некоторое время будет установлено. Включить механизм обратно можно командой:

```
(config—bgp—65001)# bgp fast—external—failover
```

56.6.3.9 Аутентификация (password)

Включить аутентификацию по паролю (MD5) можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> password <string>
```

<string> - пароль для аутентификации соседа Аутентификация настраивается на обоих соседях, устанавливающих BGP-соединение.

Пример настройки аутентификации по паролю 1ssap!

```
(config—bgp—65001)# neighbor 192.168.1.2 password 1ssap!
```

Отменить аутентификацию можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> password
```

56.6.3.10 Пассивный режим BGP-соединения (passive)

Пассивный режим позволяет получать анонсы соседа только по его инициативе. Включить пассивный режим можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> passive
```

Отключить пассивный режим можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> passive
```

56.6.3.11 IPv6 (activate)

Для разрешения обмена с адресами IPv6 перейти в подсекцию семейства адресов и выполнить команду **activate** :

```
(config—bgp—65001)# address—family ipv4 unicast
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> activate
```

Эту команду не следует использовать для IPv4.

56.6.3.12 Журнал (log-neighbor-changes)

Включить протокол изменений состояния соседей можно командой:

```
(config—bgp—65001)# bgp log—neighbor—changes
```

Изменения будут записываться в журнал маршрутизатора. Посмотреть журнал можно командой:

```
# show log router
```

Отменить ведение протокола можно командой:

```
(config—bgp—65001)# no bgp log—neighbor—changes
```

56.6.3.13 Изменение TCP-порта службы BGP (port)

BGP-соединение использует подключение к 179-му TCP-порту. Если требуется использовать подключение к другому порту, сделать это можно командой:

```
(config—bgp—65001)# neighbor <ip> port <n>
```

<n> - BGP-порт соседа.

Удалить настройку и вернуть значение по умолчанию можно командой:

```
(config—bgp—65001)# no neighbor <ip> port
```

56.6.3.14 Описание к настройкам BGP-соседа (description)

Для удобства администрирования можно задать описание для соседа или группы, при помощи команды:

```
(config—bgp—65001)# neighbor <ip>|<group> description <string>
```

<string> - Строка с описанием.

Удалить описание можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> description
```

56.6.4 Таймеры

56.6.4.1 Hold и keepalive

Для поддержания BGP-соединения используется механизм отправки сигналов жизни keepalive. При установлении соединения BGP-маршрутизаторы обмениваются параметром hold, который определяет время ожидания прихода keepalive. Изменить значения по умолчанию можно командой:

```
(config—bgp—65001)# timers bgp <keepalive> <hold>
```

<keepalive> – время отправки пакетов жизни, по умолчанию 60 секунд.

<hold> – время удержания BGP-соединения открытым до получения keepalive, по умолчанию 180 секунд.

Пример изменения таймеров keepalive на 30 секунд, hold на 90 секунд.

```
(config—bgp—65001)# timers bgp 30 90
```

Удалить введенные параметры и вернуться к значениям по умолчанию можно командой:

```
(config—bgp—65001)#no timers bgp
```

Эти же параметры можно изменить для определенного соседа командой:

```
(config—bgp—65001)# neighbor <ip>|<group> timers <keepalive> <hold>
```

Пример изменения таймеров keepalive на 30 секунд, hold на 150 секунд для соседа с IP-адресом 192.168.1.2.

```
(config—bgp—65001)# neighbor 192.168.1.2 timers 30 150
```

Удалить установленные таймеры и вернуться к значениям по умолчанию для конкретного соседа можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> timers
```

Посмотреть текущие таймеры можно командой:

```
# show bgp neighbors [<ip>|<group>]
```

56.6.4.2 Интервал анонсирования (advertisement-interval)

BGP-маршрутизатор анонсирует пути с определенным интервалом (по умолчанию 30 секунд). Изменить этот интервал можно командой:

```
(config—bgp—65001)# neighbor <ip> advertisement—interval <t>
```

<t> - время в секундах (по умолчанию 30 секунд).

Пример изменения интервала анонсирования на 10 секунд для соседа с IP-адресом 192.168.1.2:

```
(config—bgp—65001)# neighbor 192.168.1.2 advertisement—interval 10
```

Посмотреть текущий интервал анонсирования можно командой:

```
# show bgp neighbors [<ip>|<group>]
```

Удалить настроенные интервалы и вернуться к значению по умолчанию можно командой:

```
(config—bgp—65001)# no neighbor <ip> advertisement—interval
```

56.6.4.3 Интервал попыток подключения (timers connect)

BGP-маршрутизатор делает попытки установить BGP-соединение через определенный интервал (по умолчанию 120 секунд). Изменить этот интервал можно командой:

```
(config—bgp—65001)# neighbor <ip> timers connect <t>
```

<t> - время в секундах (по умолчанию 120 секунд).

Пример настройки интервала 600 секунд для соседа с IP-адресом 192.168.1.2:

```
(config—bgp—65001)# neighbor 192.168.1.2 timers connect 600
```

Таймер постоянно уменьшается от установленного значения до нуля, достигнув нуля, вновь увеличивается до максимума и начинает уменьшаться. Посмотреть интервал можно (пока не установлено BGP-соединение) командой:

```
# show bgp neighbors [<ip>]
```

Удалить настройки и вернуться к значению по умолчанию можно командой:

```
(config—bgp—65001)# no neighbor <ip> timers connect
```

56.6.5 Анонсирование

Команды анонсирования выполняются из подсекции семейства адресов Для перехода в подсекцию необходимо выполнить команду:

```
 (config—bgp—65001)# address—family ipv4|ipv6 unicast|multicast
```

56.6.5.1 Анонсирование сети (network)

При первоначальной настройке BGP-маршрутизатор не анонсирует для соседей ничего, кроме своего идентификатора. Анонсировать определенную сеть можно командой:

```
(config—bgp—65001—af4—unicast)# network <ip/m> [(route—map <rmap_name>)]
```

<ip/m> - IP-адрес и маска анонсируемой сети.

route-map <rmap_name> - анонсирует сеть с параметрами карты маршрута.

Пример анонсирования сети 10.0.0.0/8.

```
(config—bgp—65001—af4—unicast)# network 10.0.0.0/8
```

Удалить анонсированную ранее сеть можно командой:

```
(config—bgp—65001—af4—unicast)# no network <ip/m>
```

56.6.5.2 Проверка наличия маршрута в анонсируемую сеть (import-check)

По умолчанию BGP-маршрутизатор проверяет существование в таблице IP-маршрутизации маршрута к сети, которую он анонсирует соседям командой network. Проверку существования маршрута можно отключить командой:

```
(config—bgp—65001)# no bgp network import—check
```

Включить эту проверку можно командой:

```
(config—bgp—65001)# bgp network import—check
```

56.6.5.3 Перераспределение (redistribute)

Помимо явного анонсирования сетей при помощи команды network, возможно анонсировать сети путем перераспределения маршрутов из таблицы IP-маршрутизатора в BGP-маршрутизатор. Команда доступна из подсекции семейства адресов и имеет следующий вид:

```
(config—bgp—65001—af4—unicast)# redistribute kernel|connected|static|rip|ospf [metric <n>]
        [route—map <rmap>]
```

redistribute kernel – анонсирует маршруты, используемые ядром linux.

redistribute connected – анонсирует маршруты интерфейсов, подключенных к коммутатору.

redistribute static – анонсирует статические маршруты, т.е. прописанные вручную администратором.

redistribute rip – анонсирует маршруты, полученные по RIP.

redistribute ospf – анонсирует маршруты, полученные по OSPF.

metric <n> - метрика, с которой будут анонсированы эти маршруты.

route-map <rmap> - анонсирует сеть с параметрами карты маршрута.

Посмотреть какие именно маршруты находятся в таблице IP-маршрутизатора можно командой:

```
# show ip route
```

Пример перераспределения подключенных маршрутов:

```
(config—bgp—65001—af4—unicast)# redistribute connected
```

Удалить перераспределение можно командой:

```
(config—bgp—65001—af4—unicast)# no redistribute kernel|connected|static|rip|ospf
```

56.6.5.4 Суммарный путь (aggregate-address)

Суммарный путь для нескольких подсетей анонсируется соседям с целью уменьшения объемов передаваемой информации. Команда доступна из подсекции семейства адресов и имеет следующий вид:

```
(config—bgp—65001—af4—unicast)# aggregate—address <ip/m> [as—set] [summary—only]
```

<ip/m> - IP-адрес и маска сети, в которую входят подсети данного маршрутизатора.

as-set - создает новое поле AS_SET для суммарного пути, в которое записывает все AS, через которые проходят компоненты пути. Этот параметр требуется, если у путей в подсети различается поле AS_SEQ, которое при суммировании может приобрести нулевое значение, что может приводить к образованию петель.

summary-only - анонсирует только суммарный путь, пути в подсети не анонсируются.

Пример суммарного пути для сетей 192.168.1.0/25 и 192.168.1.128/25:

```
(config—bgp—65001—af4—unicast)# aggregate—address 192.168.1.0/24
```

В этом случае соседу анонсируется три пути.

```
B 192.168.1.0/24 via 195.220.1.1
B 192.168.1.0/25 via 195.220.1.1
B 192.168.1.128/25 via 195.220.1.1
```

Если требуется анонсировать только суммарный путь, то сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# aggregate—address 192.168.1.0/24 summary—only
```

В этом случае соседу анонсируется только один суммарный путь.

```
B 192.168.1.0/24 via 195.220.1.1
```

Удалить суммарный путь можно командой:

```
(config—bgp—65001—af4—unicast)# no aggregate—address <ip/m>
```

56.6.5.5 Выборочный анонс подавленных путей (unsuppress-map)

Иногда следует разрешить некоторые пути из суммарного пути для определенного соседа. Команда доступна из подсекции семейства адресов и имеет следующий вид:

```
(config—bgp—65001—af4—unicast)#  neighbor <ip>|<group> unsuppress—map <rmap_name>
```

<rmap_name> - имя карты маршрутов, содержащей параметры разрешенных путей.

Пример разрешения подсети 192.168.1.128/25 для соседа с IP-адресом 172.16.0.2.

```
(config—bgp—65001—af4—unicast)# aggregate—address 192.168.1.0/24 summary—only
(config—bgp—65001—af4—unicast)# neighbor 172.16.0.2 unsuppress—map myunmap
(config)# router router—map myunmap permit 1
(config—route—map—myunmap)# match ip address myunmapacl
(config)# router accsss—list myunmapacl permit 192.168.1.128/25
```

В этом случае соседу анонсируется только два пути.

```
B 192.168.1.0 via 195.220.1.1
B 192.168.1.128/25 via 195.220.1.1
```

Удалить настройки можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> unsuppress—map <rmap_name>
```

56.6.5.6 Маршрут по умолчанию (default-originate)

Маршрут по умолчанию можно сообщить соседу и при этом его не нужно создавать в таблице IP-маршрутизации. Команда доступна из подсекции семейства адресов и имеет следующий вид:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> default—originate [route—map
        <rmap_name>]
```

Пример анонсирования соседом 1 с IP-адреса 192.168.1.1 маршрута по умолчанию для соседа 2 с IP-адресом 192.168.1.2:

```
Сосед—1(config—bgp—65001)# neighbor 192.168.1.2 remote—as 65002
Сосед—1(config—bgp—65001)# address—family ipv4 unicast
Сосед—1(config—bgp—65001—af4—unicast)# neighbor 192.168.1.2 default—originate

Сосед—2(config—bgp—65002)# neighbor 192.168.1.1 remote—as 65001
```

У соседа 2 появится маршрут по умолчанию на шлюз с IP-адресом соседа, от которого пришел анонс.

```
B 0.0.0.0 via 192.168.1.1
```

56.6.6 Фильтрация анонсов

Команды фильтрации анонсов доступны из подсекции семейства адресов.

56.6.6.1 Distribute-list

Фильтрация при помощи distribute-list использует списки доступа маршрутизатора. Включить такую фильтрацию можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> distribute—list <racl_name>|<racl_num>
        in|out
```

<racl_name>|<racl_num> - имя либо номер списка доступа маршрутизатора.

distribute-list in – фильтруются входящие анонсы, distribute-list out – фильтруются исходящие анонсы.

Пример использования списка доступа myacl, разрешающего только пути в сеть 10.0.0.0/8, для фильтрации анонсов, получаемых от соседа с IP-адресом 192.168.1.2:

```
(config)# router access—list myacl permit 10.0.0.0/8
(config—bgp—65001—af4—unicast)# neighbor 192.168.1.2 distribute—list myacl in
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001—af4—unicast)#no neighbor <ip>|<group> distribute—list
        <racl_name>|<racl_num> in|out
```

56.6.6.2 Filter-list

Фильтрация при помощи filter-list использует списки доступа по AS-путям. Этот список применяет строку с регулярным выражением для поиска в атрибуте AS_PATH. Включить такую фильтрацию можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> filter—list <apath_name> in|out
```

<apath_name> - имя списка доступа по AS-путям.

Пример использования списка доступа myasacl, разрешающего только пути, проходящие через AS 65001, для фильтрации исходящих анонсов для соседа с IP-адресом 192.168.1.2.

```
(config)# router as—path access—list myasacl permit _65001_
(config—bgp—65001—af4—unicast)# neighbor 192.168.1.2 filter—list myasacl out
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001—af4—unicast)#no neighbor <ip>|<group> filter—list <apath_name> in|out
```

56.6.6.3 Prefix-list

Фильтрация при помощи prefix-list использует префиксные списки. Эти списки похожи на списки доступа маршрутизатора. В отличие от списков доступа маршрутизатора, префиксные списки можно применять удаленно (outbond route filtering). Включить такую фильтрацию можно следующей командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> prefix—list <prlist_name> in|out
```

<prlist_name> - имя префиксного списка.

Пример использования списка доступа с myplist, разрешающего только пути в сеть 10.0.0.0/8, для фильтрации анонсов, получаемых от соседа с IP-адресом 192.168.1.2.

```
(config)# router access—list myplist permit 10.0.0.0/8
(config—bgp—65001—af4—unicast)# neighbor 192.168.1.2 distribute—list myplist in
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> prefix—list <prlist_name> in|out
```

56.6.6.4 Maximum-prefix

Установить ограничение на количество префиксов (сетей), получаемых от соседа при переполнении памяти, можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> maximum—prefix <limit> <thresh>
        [(restart <mins>)|warning—only]
```

<limit> - число получаемых префиксов.

<thresh> - процент от ограничения, при превышении которого выдавать предупреждение.

restart <mins> - рестарт BGP-соединения немедленно либо через несколько минут, если превышен лимит.

warning-only – не делать рестарт, а только выдавать предупреждение.

Отменить такое ограничение можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> maximum—prefix
```

56.6.6.5 Route-map

Фильтрацию при помощи карты маршрутов можно включить командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> route—map <rmap_name> in|out
```

in|out – применяет карту к входящим либо исходящим путям.

Отменить использование карты маршрутов можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> route—map <rmap_name> in|out
```

56.6.7 Замена полносвязного графа BGP

56.6.7.1 IBGP-конфедерация (confederation)

Конфедерации описаны в RFC 5065 «Autonomous System Confederations for BGP». Конфедерация под своим номером объединяет несколько AS. Для BGP-маршрутизаторов вне конфедерации, эти AS представляются, как единая AS. AS внутри конфедерации общаются между собой по EBGP.

Конфедерация позволяет упростить задачу настройки BGP-маршрутизаторов внутри одной организации. Без использования конфедерации требуется что бы организация, имеющая один публичный номер AS, обеспечила внутри своей AS полносвязное IBGP-соединение.

Для объединения в конфедерацию следует указать номер конфедерации, сделать это можно командой:

```
(config—bgp—65001)# bgp confederation identifier <AS>
```

<AS> - номер AS, которым будут представляться члены конфедерации.

Пример включения BGP-маршрутизатора из AS 65001 в конфедерацию с номером 1000.

```
(config—bgp—65001)# bgp confederation identifier 1000
```

Исключить узел из конфедерации можно командой:

```
(config—bgp—65001)# no bgp confederation identifier
```

Для обеспечения связей внутри конфедерации следует указать остальные AS, которые входят в конфедерацию. Сделать это можно командой:

```
(config—bgp—65001)# bgp confederation peers <AS>
```

Пример добавления в конфедерацию BGP-маршрутизаторов из AS 65002

```
(config—bgp—65001)# bgp confederation peers 65002
```

Если AS больше не член конфедерации, то удалить ее можно командой:

```
(config—bgp—65001)# no bgp confederation peers <AS>
```

56.6.7.2 IBGP отражатель маршрутов (cluster, route-reflector-client)

Отражатель маршрутов описан в RFC 4456 «BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)». Отражатель маршрутов, это роль BGP-маршрутизатора в BGP-кластере. Внутри одной AS BGP-маршрутизаторы можно разделить на кластеры (группы), в каждом кластере выбрать BGP-маршрутизатор на роль отражателя маршрутов. В этом случае полносвязное BGP-соединение требуется обеспечить только между отражателями. Остальные BGP-маршрутизаторы, члены кластера, будут получать маршруты через своего отражателя, т.е. являться его клиентами.

Чтобы настроить BGP-маршрутизатор, как отражатель маршрутов, следует присвоить ему идентификатор кластера. Сделать это можно командой:

```
(config—bgp—65001)# bgp cluster—id <id>
```

<id> - номер кластера.

Удалить номер кластера можно командой:

```
(config—bgp—65001)# no bgp cluster—id
```

Затем следует обозначить клиентов кластера. Сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> route—reflector—client
```

При этом не требуется, чтобы клиенты в кластере имели полносвязную топологию, так как отражатель передает все маршруты между всеми клиентами. Если же клиенты кластера имеют полносвязную топологию, то передачу маршрутов между клиентами одного кластера следует отключить на отражателе. Сделать это можно командой:

```
(config—bgp—65001)# no bgp client—to—client reflection
```

56.6.7.3 EBGP сервер маршрутов (route-server-client)

Сервер маршрутов является центром в топологии звезда, при которой EBGP-соседи используют его, как транзитный узел, для обмена между собой. Для включения централизованного обмена на сервере следует объявить соседей, как Route Server Client. Сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> route—server—client
```

56.6.8 Перемещение BGP-путей в IP-маршрутизатор

56.6.8.1 Атрибуты пути

Пути имеют атрибуты, которые разделены на 4 категории:

1. Хорошо известные, обязательные (Well-known mandatory) —должны распознаваться и присутствовать во всех анонсах:
    - AS_PATH (Autonomous system path) - определяют автономные системы, через которые доступна сеть назначения;
    - NEXT_HOP - определяет IP-адрес пограничного маршрутизатора, который должен рассматриваться, как шлюз к сети назначения в таблице маршрутизации;
    - ORIGIN - определяет происхождение пути.
2. Хорошо известные, не обязательные (Well-known discretionary) —должны распознаваться, но наличие в анонсах не обязательно:
    - LOCAL_PREF (Local preference) - используется чтобы сообщить соседям внутри своей автономной системы степень предпочтения пути;
    - ATOMIC_AGGREGATE - используется для информирования соседей о суммарном пути.
3. Дополнительные пересылаемые (Optional transitive) — могут не распознаваться, но должны передаваться соседям:
    - AGGREGATOR - содержит номер последней AS, и IP-адрес BGP-маршрутизатора, который сформировал суммарный путь.
4. Дополнительные не пересылаемые (Optional non-transitive) — могут не распознаваться и отбрасываться:
    - MULTI_EXIT_DISC (Multi-exit discriminator, MED) - может использоваться при выборе одного из нескольких путей к соседней автономной системе.

56.6.8.2 Алгоритм выбора лучшего пути

1. Weight. Лучшим считается путь с наибольшим значением веса.
2. LOCAL_PREF. Если вес путей одинаков, то выбирается путь с наибольшим значением атрибута LOCAL_PREF.

3. Локальные пути. Если атрибуты LOCAL_PREF одинаковы, пути объявленные командами network, redistribute и aggregate-address предпочитаются над путями, полученными от соседей.
4. AS_PATH. Если нет локальных путей, то выбирается путь с самым коротким атрибутом AS_PATH.
5. ORIGIN. Если все пути имеют одинаковую длину атрибута AS_PATH, то выбирается путь с меньшим атрибутом ORIGIN (IGP < EGP < Incomplete).
6. MED. Если атрибут ORIGIN одинаков, то выбирается путь с наименьшим атрибутом MED.
7. EBGP или IBGP. Если атрибут MED одинаков, то предпочтение отдается маршрутам, полученным по EBGP, над маршрутами, полученными по IBGP.
8. Метрика маршрута до next hop. Чем меньше, тем предпочтительнее.
9. Маршрут первый появившийся в таблице (самый старый).

56.6.8.3 Вес пути (Weight)

Weight, это локальный атрибут, который не передается соседям. По умолчанию, значение веса 32768 для локальных путей, которые созданы на данном роутере, и 0 - для всех остальных путей. Если необходимо устанавливать определенный вес для пути, то сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> weight <n>
```

<n> - число определяющее вес пути, по умолчанию 0.

Пример увеличения веса путей, полученных от соседа с IP-адресом 192.168.1.2.

```
(config—bgp—65001—af4—unicast)# neighbor 192.168.1.2 weight 40000
```

Посмотреть вес можно командой:

```
# show bgp
```

Вернуть значения по умолчанию можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> weight
```

56.6.8.4 Атрибут LOCAL_PREF (local-preference)

Атрибут присваивается маршрутам, полученным через EBGP, и локальным по команде network, атрибут предается только по IBPG. Установить определенное значение можно командой:

```
(config—bgp—65001)# bgp default local—preference <n>
```

<n> новое значение атрибута, по умолчанию 100.

Вернуть значение по умолчанию можно командой:

```
(config—bgp—65001)# no bgp default local—preference
```

56.6.8.5 Атрибут AS_PATH

56.6.8.5.1 Разрешить принимать пути с номером собственной AS (allowas-in) По умолчанию для исключения петель маршрутизатор отбрасывает путь, полученный по EBGP, если видит в AS_PATH номер своей AS. Можно отменить это правило командой:

```
(config—bgp—65001—af4—unicast)#  neighbor <ip>|<group> allowas—in [<n>]
```

Вернуть поведение по умолчанию можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> allowas—in
```

56.6.8.5.2 Изменить номер собственной AS для отдельных соседей (local-as) Можно указать другую локальную AS для отдельного EBGP-соседа, отличную от той, в которую входит данный маршрутизатор. Сосед при этом должен указать другую AS в команде network remote-as. Сделать это можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> local—as <AS> [no—prepend]
```

<AS> - номер AS отличный от собственной.

no-prepend – пути, полученные через EBGP, не предваряются номером local-as при распространении по IBGP.

Отменить замену номера можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> local—as
```

56.6.8.5.3 Проверить номер первой AS (enforce-first-as) Принимать пути только от EBGP-соседей, чей номер AS стоит первым в AS_PATH. Включить такую проверку можно командой:

```
(config—bgp—65001)# bgp enforce—first—as
```

Отменить проверку можно командой:

```
(config—bgp—65001)# no bgp enforce—first—as
```

56.6.8.5.4 Удалить частные AS (remove-private-as) Номера частных автономных систем с 64512 по 65535 используются в частных сетях и не используются интернет-провайдерами. Можно удалять их из получаемых анонсов от EBGP-соседей. Включить удаление можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> remove—private—AS
```

Отменить удаление можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> remove—private—AS
```

56.6.8.5.5 Длина атрибута AS_PATH (confed, ignore) По умолчанию AS конфедераций не влияют на выбор «лучшего» маршрута. Если требуется учитывать также AS конфедераций, сделать это можно командой:

```
(config—bgp—65001)# bgp bestpath as—path confed
```

Если требуется отменить сравнение путей по длине атрибута AS_PATH, сделать это можно командой:

```
(config—bgp—65001)# bgp bestpath as—path ignore
```

56.6.8.6 Атрибут HEXT_HOP (next-hop-self)

По умолчанию, когда путь, полученный по EBGP, анонсируется IBGP-соседу, атрибут NEXT_HOP (шлюз) не изменяется. В случае если этот шлюз недоступен, его нужно подменить на доступный IP-адрес соседа. Сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> next—hop—self
```

По умолчанию, когда путь анонсируется EBGP-соседу, атрибут NEXT_HOP всегда меняется на IP-адрес соседа, который анонсирует маршрут. Отменить подмену можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> next—hop—self
```

56.6.8.7 Атрибут MED (always-compare-med, confed, missing-as-worst, deterministic-med)

По умолчанию MED сравнивается только для путей из одной и той же AS, т.е. для путей с одинаковой первой AS в атрибуте AS_PATH. Если требуется сравнивать MED для любых путей, сделать это можно командой:

```
(config—bgp—65001)# bgp always—compare—med
```

По умолчанию, MED не сравнивается для путей из конфедераций. Если требуется учитывать также конфедерации, сделать это можно командой:

```
(config—bgp—65001)# bgp bestpath med confed
```

По умолчанию, если атрибут MED отсутствует в полученном пути, то ему присваивается значение 0 (самый высокий приоритет). Если требуется присвоить отсутствующему атрибуту самый низкий приоритет (4294967295), то сделать это можно командой:

```
(config—bgp—65001)# bgp bestpath med missing—as—worst
```

Можно изменить алгоритм выбора лучшего пути, таким образом, что все пути в одну сеть будут сравниваются друг с другом, несмотря на разные AS этих путей и порядок получения. Сделать это можно командой:

```
(config—bgp—65001)# bgp deterministic—med
```

56.6.8.8 Сравнить по идентификаторам (compare-routerid)

Полностью одинаковые пути EBGP не сравниваются, а лучшим выбирается первый полученный (самый старый). Можно заменить такой выбор на выбор пути с наименьшем идентификатором. Сделать это можно командой:

```
(config—bgp—65001)# bgp bestpath compare—routerid
```

56.6.8.9 Контроль атрибутов через карту маршрутов

56.6.8.9.1 Вес пути (set weight) Установить вес можно следующей командой:

```
(config—route—map—myrmap)# set weight <n>
```

<n> - вес пути

56.6.8.9.2 Атрибут LOCAL_PREF (set local-preference) Установить атрибут можно следующей командой:

```
(config—route—map—myrmap)# set local—preference <n>
```

<n> - новое значение атрибута

56.6.8.9.3 Атрибут AS_PATH (match/set as-path) Критерий отбора по спискам доступа по AS-путям можно установить следующей командой:

```
(config—route—map—myrmap)#match as—path <as_path_acl>
```

Изменить атрибут AS_PATH, добавив либо удалив номера AS можно следующей командой:

```
set as—path prepend|exclude <as_num1> [<as_num2>] [<as_num3>] ...
```

prepend – добавить номера AS.

exclude – удалить номера AS.

56.6.8.9.4 Атрибут ORIGIN (match/set origin) Критерий происхождения маршрута:

```
(config—route—map—myrmap)# match origin egp|igp|incomplete
```

Установить происхождение маршрута:

```
(config—route—map—myrmap)# set origin egp|igp|incomplete
```

56.6.8.9.5 Атрибуты AGGREGATOR и ATOMIC_AGGREGATE (set aggregator, atomic-aggregate)

При формировании суммарного маршрута BGP-маршрутизатор добавляет к пути атрибуты AGGREGATOR и ATOMIC_AGGREGATE. Установить атрибут aggregator можно следующей командой:

```
(config—route—map—myrmap)# set aggregator as <as_num> <ip>
```

<as_num> - номер AS, обычно последняя AS.

<ip> - IP-адрес, обычно адрес BGP-маршрутизатора сформировавшего маршрут.

Установить атрибут atomic-aggregate можно следующей командой:

```
(config—route—map—myrmap)# set atomic—aggregate
```

56.6.8.9.6 Атрибут Ordinator-ID (set originator-id)

При работе BGP-маршрутизаторов в кластере к путям, которые проходят через отражатель маршрутов, добавляется атрибут Ordinator-ID. Данный атрибут - это IP-адрес - идентификатор маршрутизатора источника маршрута. Атрибут нужен для исключения петель. BGP-маршрутизатор отбрасывает анонсы со своим Ordinator-ID. Установить атрибут originator ID можно следующей командой:

```
(config—route—map—myrmap)# set originator—id <ip>
```

56.6.8.10 Сохранение атрибутов (attribute-unchanged)

При анонсе путей, которые были получены по BGP, BGP-маршрутизатору можно запретить менять исходные атрибуты пути. Сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> attribute—unchanged [as—path]
        [next—hop] [med]
```

Параметры as-path, next-hop и med – определяют, какие именно атрибуты оставить в неизменном виде.

Отменить сохранение атрибутов можно командой:

```
(config—bgp—65001—af4—unicast)# no neighbor <ip>|<group> attribute—unchanged
```

56.6.8.11 Административная дистанция (AD)

AD используется для изменения приоритета путей, полученных от разных протоколов. Работает после выбора лучшего пути до помещения пути в таблицу маршрутизации. Чем меньше AD, тем приоритетнее путь. Значения AD: подключенный интерфейс 0, статический маршрут 1, EBGP 20, OSPF 110, RIP 120, IBGP 200.

Если требуется изменить значение AD, которые будут использоваться для всех соседей, то сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# distance bgp <ext> <int> <local>
```

<ext> - AD для путей, полученных через EBGP, (по умолчанию 20).

<int> - AD для путей, полученных через IBGP, (по умолчанию 200).

<local> - AD для путей, настроенных вручную, через команду network, (по умолчанию 200).

Пример задания AD 200 для любых путей из BGP:

```
(config—bgp—65001—af4—unicast)# distance bgp 200 200 200
```

Возврат значений по умолчанию выполняется командой:

```
(config—bgp—65001—af4—unicast)# no distance bgp
```

Если требуется настроить значения AD для путей от конкретного соседа, это выполняется командой:

```
(config—bgp—65001—af4—unicast)# distance (<n> <ip/m> [<racl>])
```

<n> - значение AD.

<ip/m> - IP-адрес соседа.

<racl> - список доступа, который определяет, каким конкретно путям присваивать AD.

Пример указания AD 200 для путей, полученных от соседа с IP-адресом 195.220.1.2/32. Предварительно создан список доступа myadracl, разрешающий сеть 10.1.2.0/24. AD изменяется для путей в эту сеть.

```
(config)# router access—list myadracl permit 10.1.2.0/24
(config—bgp—65001—af4—unicast)# distance 200 195.220.1.2/32 myadracl
```

Для удаления изменений AD и возврата значений по умолчанию используется команда:

```
(config—bgp—65001—af4—unicast)# no distance <n> <ip/m>
```

56.6.8.12 Backdoor

Иногда требуется уменьшить приоритет пути, полученного по EBGP (AD 20), если есть путь в ту же сеть, но полученный, например по OSPF (AD 110). Сделать это можно командой, которая устанавливает административную дистанцию 200 вместо 20 для сети, полученной по EBGP:

```
(config—bgp—65001—af4—unicast)# network <ip/m> backdoor
```

<ip/m> - IP-адрес и маска сети, получаемой по EBGP, для которой требуется уменьшить приоритет. Данная сеть не анонсируется по BGP другим соседям.

56.6.9 Сообщества

56.6.9.1 Атрибуты сообществ (send-community)

Сообщества описаны в RFC 1997 «BGP Communities Attribute» и RFC 4360 «BGP Extended Communities Attribute». Предназначены для облегчения управления анонсами на основе политик. Сообщества добавляют атрибут COMMUNITY к пути. Вид атрибута AS:Number, где AS – номер AS, Number -

номер политики. По номеру политики можно устанавливать различные атрибуты, например LOCAL_PREF. Некоторые сообщества имеют зарезервированные имена:

internet - анонсировать этот путь в интернет. Любой путь принадлежит этому сообществу.

local-AS - это сообщество запрещает передачу путей за пределы собственной AS.

no-advertise - запрещает анонсы любому соседу.

no-export - запрещает анонсы EBGP-соседям.

Для того чтобы использовать атрибуты требуется разрешить пересылку и прием их для соседей. Сделать это можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> send—community standard|extended|both
```

standard, extended и both – определяет, какие списки сообществ отправлять и принимать. Списки сообществ бывают со стандартными (standard) и расширенными (extended) атрибутами; в каждом из списков можно применять регулярные выражения.

56.6.9.2 Стандартный список (community-list)

Создать стандартное сообщество можно командой:

```
(config)# router community—list <1..99>|(standard <name>) permit|deny <com_name>
```

<1..99> - задает номер списка сообществ.

standard <name> задет имя списка сообществ.

<com_name> - имя сообществ, может иметь вид <NN>:<NN> либо одно из стандартных названий internet, local-AS, no-advertise или no-export

Можно создать список сообществ с регулярным выражением:

```
(config)# router community—list <100..500>|(expanded <name>) permit|deny <regexp>
```

<regexp> - регулярное выражение.

56.6.9.3 Расширеный список (extcommunity-list)

Создать расширенный список сообществ можно командой:

```
(config)# router extcommunity—list <1..99>|(standard <name>) permit|deny <([rt <ip>:NN|NN:NN]
        [soo <ip>:NN|NN:NN])>
```

rt <ip>:NN|NN:NN – это (Route Target) идентификатор маршрутизатора источника пути с этим сообществом.

soo <ip>:NN|NN:NN - это (Site of Origin) идентификатор сайта источника пути.

Можно создать список сообществ с регулярным выражением.

```
(config)# router extcommunity—list <100..500>|(expanded <name>) permit|deny <regexp>
```

56.6.9.4 Фильтрация при помощи карт маршрутов

Критерий сравнения атрибутов пути со стандартным списком задается при помощи команды:

```
(config—route—map—myrmap)# match community <num>|<comlist_name> [exact—match]
```

exact-match – точное совпадение.

Критерий сравнения атрибутов пути с расширенным списком списком задается при помощи команды:

```
(config—route—map—myrmap)# match extcommunity <num>|<ecomlist_name> [exact—match]
```

Установка атрибутов стандартного сообщества осуществляется при помощи команды:

```
(config—route—map—myrmap)# set community
        <NN>[:<NN>]|internet|local—AS|no—advertise|no—export|additive|none ...
```

Установка атрибутов расширенного сообщества осуществляется при помощи команды:

```
(config—route—map—myrmap)# set extcommunity [rt <ip>:NN|NN:NN] [soo <ip>:NN|NN:NN] ...
```

Удаление атрибутов стандартного или расширенного сообщества осуществляется при помощи команды:

```
(config—route—map—myrmap)# set comm—list <num>|<comlist_name> delete
```

56.6.10 Дополнительные возможности BGP-маршрутизатора

56.6.10.1 Поддержка протокола BFD

Для включения поддержки протокола BFD необходимо выполнить команду:

```
(config—bgp—65001)# neighbor <ip> bfd
```

56.6.10.2 Согласование возможностей (capability)

При согласовании соединения BGP-маршрутизаторы обмениваются информацией о своих возможностях. За каждой хорошо известной возможностью закреплен определенный код. Например:

1 - Multiprotocol Extensions for BGP-4.

2 - Route Refresh Capability for BGP-4.

3 - Outbound Route Filtering Capability.

4 - Multiple routes to a destination capability.

5 - Extended Next Hop Encoding.

64 - Graceful Restart Capability.

67 - Support for Dynamic Capability (capability specific).

По умолчанию включен минимальный набор хорошо известных возможностей. Для проверки точного совпадения возможностей данного маршрутизатора с соседним можно использовать команду:

```
(config—bgp—65001)# neighbor <ip> strict—capability—match
```

Некоторые старые версии BGP не умеют согласовывать возможности. Для отключения согласования своих возможностей с соседом можно использовать команду:

```
(config—bgp—65001)# neighbor <ip>|<group> dont—capability—negotiate
```

Можно также игнорировать согласованные возможности и принудительно использовать все свои возможности. Сделать так можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> override—capability
```

Включить возможность динамического согласования можно командой:

```
(config—bgp—65001)# neighbor <ip>|<group> capability dynamic
```

56.6.10.3 Удаленное применение фильтров (outbound route filtering)

Если сосед поддерживает возможность удаленного применения фильтров, то можно настроить префиксные списки, для отправки и применения их на стороне соседа. Таким образом сокращается объем анонса. Включить такую возможность можно командой:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> capability orf prefix—list send|receive|both
```

56.6.10.4 Мягкая перезагрузка (graceful-restart)

Механизм «Graceful-restart» описан в RFC 4724 «Graceful Restart Mechanism for BGP». Этот механизм позволяет сохранять состояние BGP-соединений и не трогать IP-маршрутизацию, в процессе перезапуска BGP-маршрутизатора, что снижает нагрузку и позволяет избавиться от временных петель в маршрутизации. Включить механизм «Graceful Restart» можно командой:

```
(config—bgp—65001)# bgp graceful—restart [stalepath—time <t>]
```

<t> - время в секундах, которое BGP-маршрутизатор ждет с момента получения сигнала о перезапуске, прежде чем сбросить BGP-соединения. По умолчанию 360 секунд.

Выключить этот механизм можно командой:

```
(config—bgp—65001)# no bgp graceful—restart [stalepath—time]
```

56.6.10.5 Мягкое применение политик без обрыва сессий (soft-reconfiguration)

При изменении настроек, например входящих фильтров, BGP требуется заново получить от соседа пути, чтобы по-новому их отфильтровать. Сделать это можно, принудительно очистив таблицу BGP командой clear bgp, в этом случае информация удаляется, соединения обрываются и устанавливаются заново. Можно воспользоваться более мягким вариантом clear bgp soft in, при котором изменения

применяются без обрыва сессий. Для того чтобы работал такой вариант BGP требуется хранить информацию о всех путях от соседа (не отфильтрованных) в памяти. При перезагрузке фильтры применятся к путям в памяти, соединение не разорвется.

Для того чтобы включить данный механизм нужно выполнить команду:

```
(config—bgp—65001—af4—unicast)# neighbor <ip>|<group> soft—reconfiguration inbound
```

После этого можно пользоваться clear bgp soft in.

56.6.10.6 Быстрое применение политик с принудительным обновлением анонсов (route refresh)

Более совершенная возможность заменяющая «soft reconfiguration» позволяет принудительно запросить свежий анонс у соседа либо отправить анонс соседу без разрыва сессии.

Чтобы использовать возможность не требуется дополнительных команд конфигурации, Принудительно запросить анонс можно командой:

```
# clear bgp in
```

Отправить анонс можно командой:

```
# clear bgp out
```

56.6.10.7 Подавление мигающих путей (dampening)

Механизм защиты от мигающих (хлопающих) путей описан в RFC 2439 «BGP Route Flap Damping». Когда путь анонсируется и быстро исчезает из анонса, то данному пути начисляется штраф 1000 очков и путь запоминается, как «мигающий». Если он снова появился и исчез, то еще 1000 очков и т.д. за каждое мигание. Если число очков превысило настроенный предел, то путь помечается, как «подавленный». Подавленные пути не анонсируются и не передаются в IP-маршрутизатор.

Включить этот механизм можно командой:

```
(config—bgp—65001)# bgp dampening [<halflife> [<reuse_start> <suppress_thresh> <suppress_dur>]]
```

<halflife> - время в минутах, после которого штраф уменьшается наполовину, (по умолчанию 15 минут).

<reuse_start> - размер штрафа, ниже которого путь перестаёт быть подавленным, (по умолчанию 750 очков).

<suppress_thresh> - размер штрафа, выше которого путь подавляется, (по умолчанию 2000 очков).

<suppress_dur> - время в минутах, в течение которого путь подавляется, обычно в четыре раза больше, чем halflife, (по умолчанию 60 минут).

Пример настройки механизма защиты от мигающих путей со значениями по умолчанию:

```
(config—bgp—65001)# bgp dampening
```

Выключить механизм можно командой:

```
(config—bgp—65001)# no bgp dampening
```

56.7 IS-IS

56.7.1 Введение

IS-IS - протокол маршрутизации, для использования в IGP-сетях. По сравнению с RIP этот протокол обеспечивает лучшую масштабируемость, и более быстрое “схождение” сети. Данный протокол может быть использован как в сетях провайдера, так и в магистральных сетях.

Домен маршрутизации IS-IS можно разделить на зоны (области). В основе иерархии IS-IS лежат уровни взаимодействия маршрутизаторов друг с другом. Таким образом пара IS-IS маршрутизаторов, подключенных друг к другу напрямую могут сформировать два вида взаимодействий - уровень 1(Level 1 или L1) и уровень 2 (Level 2 или L2).

Правила для организации соседств разных уровней следующие:

- соседство уровня 1 формируется только между маршрутизаторами в одной зоне;
- соседство уровня 2 может быть сформировано как между маршрутизаторами одной зоны, так и между маршрутизаторами разных зон.

Таким образом такие уровни взаимодействий позволяют выделить три типа маршрутизаторов в IS-IS домене:

- маршрутизаторы L1 - это те устройства, у которых все взаимодействия с другими маршрутизаторами происходят на 1-ом уровне;
- маршрутизаторы L2 - это те устройства, у которых все соседства организованы на 2-ом уровне;
- маршрутизаторы L1/L2 - это устройства, поддерживающие взаимодействия обоих уровней.

Некоторые особенности маршрутизаторов разных уровней:

- всё множество L2 взаимодействий между маршрутизаторами должно быть непрерывным (таким образом маршрутизаторы L2 представляют из себя ядро сети);
- L2 маршрутизаторы знают всю топологию сети;
- L1 маршрутизаторы знают топологию только своей зоны.

56.7.2 IS-IS - маршрутизатор

DionisNX поддерживает работу всех типов маршрутизаторов (L1, L2, L1/L2).

Первое, что требуется для начала настройки IS-IS, это включить IS-IS-маршрутизатор. Сделать это можно командой:

```
(config)# router isis <AT>
```

<AT> - Идентификатор зоны (Area Tag), в которую входит настраиваемый IS-IS-маршрутизатор. Идентификатор является условным и нужен активации рассылки Hello-пакетов на интерфейсе (*ip router isis <AT>*. См. далее).

После ввода команды система переходит в режим конфигурирования IS-IS, в котором вводятся остальные команды настройки IS-IS-маршрутизатора.

Выключить IS-IS-маршрутизатор можно командой:

```
(config)# no router isis <AT>
```

Конфигурация IS-IS-маршрутизатора при этом удаляется.

56.7.2.1 Основные команды для настройки IS-IS-маршрутизатора

Предположим, что Area Tag соответствует значению "**1**".

Задать заголовок сетевого объекта в формате ISO:

```
(config—isis—1)# isis net <NET>
```

Задать тип маршрутизатора (L1,L2 или L1/L2):

```
(config—isis—1)# isis net <LEVEL>
```

Поддержка динамического имени хоста:

```
(config—isis—1)# isis hostname dynamic
```

Задать пароль домена:

```
(config—isis—1)# domain—password <clear|md5> <STRING>
```

Задать пароль зоны:

```
(config—isis—1)# area—password <clear|md5> <STRING>
```

Отображать в журнале работы изменения состояния смежности:

```
(config—isis—1)# log—adjacency—changes
```

Задать стиль формата пакетов:

```
(config—isis—1)# metric—style <narrow|transition|wide>
```

Использовать бит перезагрузки:

```
(config—isis—1)# set—overload—bit
```

Чтобы импортировать маршруты из других протоколов маршрутизации в IS-IS, необходимо указать опцию(и) «redistribute» в режиме конфигурации IS-IS. Формат опции:

```
(config—isis—1)# redistribute <версия ip—протокола> <тип_маршрута> <уровень маршрутизатора>
        [metric <n>]  [route—map <rmap_name>]
```

Интервал между повторной генерацией LSP:

```
(config—isis—1)# lsp—gen—interval <VAL>
```

Интервал между вычислением SPF (кротчайший путь):

```
(config—isis—1)# spf—interval <VAL>
```

Для того, чтобы IS-IS-маршрутизатор начал взаимодействовать с другими IS-IS-маршрутизаторами сети необходимо включить поддержку данного маршрутизатора на заданном интерфейсе (например *ethernet 0*), а также указать уровень взаимодействия в другими маршрутизаторами. Для этого необходимо выполнить команды:

```
(config—if—ethernet0)# ip router isis <AT>
(config—if—ethernet0)# isis circuit—type <LEVEL>
```

56.7.2.2 Пример настройки IS-IS маршрутизаторов

Ниже приведен пример настройки упрошенной схемы с тремя IS-IS маршрутизаторами R1, R2 и R3. R1 в данном примере является маршрутизатором уровня L1, R2 - маршрутизатором уровня L1/L2, R3 - маршрутизатором уровня L2.

Пример конфигурации маршрутизатора уровня L1 (маршрутизатор R1):

```
router isis 1
 isis net 49.0008.0100.1008.4001.00
 no isis hostname dynamic
 is—type level—1
 lsp—gen—interval 1
 lsp—refresh—interval 50
 max—lsp—lifetime 350
 log—adjacency—changes

interface ethernet 1
 description ”To R2”
 ip address 10.0.3.1/24
 ip router isis 1
 isis circuit—type level—1
 isis psnp—interval 1
 enable
```

Пример конфигурации маршрутизатора уровня L1/L2 (маршрутизатор R2):

```
router isis 3
 isis net 49.0008.0100.1008.4003.00
 no isis hostname dynamic
 lsp—gen—interval 1
 lsp—refresh—interval 50
 max—lsp—lifetime 350
 log—adjacency—changes
```

```
interface ethernet 0
 description ”To R1”
 ip address 10.0.3.3/24
 ip router isis 3
 isis psnp—interval 1
 enable

interface ethernet 2
 description ”To R3”
 ip address 172.16.0.3/24
 ip router isis 3
 isis psnp—interval 1
 enable
```

Пример конфигурации маршрутизатора уровня L2 (маршрутизатор R3):

```
router isis 4
 isis net 49.0008.0100.1008.4004.00
 is—type level—2—only
 no isis hostname dynamic
 lsp—gen—interval 1
 lsp—refresh—interval 50
 max—lsp—lifetime 350
 log—adjacency—changes

interface ethernet 0
 description ”To R2”
 ip address 172.16.0.4/24
 ip router isis 4
 isis circuit—type level—2—only
 isis psnp—interval 1
 enable
```

56.8 BFD

56.8.1 Краткое описание протокола BFD

Bidirectional Forwarding Detection protocol (BFD) — протокол, созданный для быстрого обнаружения неисправностей линков. Два устройства согласовывают и устанавливают BFD сессию, отправляют друг другу hello-сообщения. Если hello-сообщения перестают поступать от соседа, BFD-сессия разрывается и система оповещается о неполадках в коммуникациях. BFD может определить неисправность линка

менее чем за 1 секунду. Когда имеет место неисправность, то BFD оповещает о ней протокол маршрутизации, для которого настроен, и после этого протокол маршрутизации может предпринять необходимое действие.

Ключевые особенности протокола:

- Более быстрое обнаружение неисправности, по-сравнению со средствами протоколов маршрутизации;
- Обеспечение единого метода управления таймерами протокола, что позволяет не менять таймеры каждого протокола в отдельности.

На данный момент в Dionis-NX с протоколом BFD могут работать протоколы OSPF и BGP.

56.8.2 Активация и настройка службы BFD на узле Dionis-NX

Чтобы активировать службу и войти в режим настроек BFD, нужно ввести команду конфигурации:

```
(config)# router bfd
(config—bfd)#
```

В режиме «config-bfd» вводятся общие настройки BFD для данного узла Dionis-NX. Также существуют настройки BFD, относящиеся к конкретным протоколам маршрутизации.

Для останова службы BFD и удаления всех настроек можно использовать команду: (config)# no router bfd

Для работы BFD-маршрутизатора требуется указать IP адреса связных BFD-маршрутизаторов для приема и отправки сообщений протокола. Это можно сделать следующей командой:

```
(config—bfd)# peer <A.B.C.D|X:X::X:X> [{multihop|local—address <A.B.C.D|X:X::X:X>|interface
        IFNAME|vrf NAME}]
```

Для отключения взаимодействия со связным узлом в секции узла необходимо выполнить команду:

```
(config—bfd—peer—A.B.C.D)# shutdown
```

Удалить узел можно командой:

```
(config—bfd)# no peer <A.B.C.D|X:X::X:X> [{multihop|local—address <A.B.C.D|X:X::X:X>|interface
        IFNAME|vrf NAME}]
```

56.8.2.0.1 Настройка BFD Посмотреть информацию обо всех связных узлах можно командой:

```
# show bfd peers
```

Посмотреть информацию о состоянии конкретного узла можно командой:

```
# show bfd peer <A.B.C.D>
```

56.8.2.1 Настройки связных BFD-маршрутизаторов.

Команды, доступные в режиме настройки связных BFD-маршрутизаторов:

Команда	Краткое описание
detect-multiplier <2-255>	Множитель для таймера обнаружения потери соединения
echo-interval <10-60000>	Настройка интервала эхо-запросов
echo-mode	Включение режима эхо-передачи
label	Описание узла
receive-interval <10-60000>	Настройка интервала приема контрольных пакетов
transmit-interval <10-60000>	Настройка интервала отправки контрольных пакетов
shutdown	Отключить взаимодействие со связным BFD-марзрутиратором

56.9 Карты маршрутов

56.9.1 Описание карт маршрутов

Карты маршрутов позволяют изменять атрибуты путей и маршруты по определенным критериям. Они обычно применяются для управления маршрутами в протоколах динамической маршрутизации, чаще всего для BGP.

Карта маршрутов - это набор правил. Каждое правило имеет свой порядковый номер и политику. Политика может быть либо разрешающей вносить изменения и передавать маршрут на дальнейшую обработку, либо запрещающей обрабатывать маршрут.

Правило содержит разделы:

- Описание (description) задает комментарий для правила и служит для удобства;
- Критерии (match) задают условия, при которых срабатывает правило. Если маршрут попал под заданные критерии, то при разрешающей политике этого правила, выполняются остальные разделы: установки (set), вызов другой карты (call) и действие (action). При запрещающей – маршруты, попавшие под правило, отбрасываются. Если маршрут не попал под критерии, то рассматриваются критерии правила со следующим порядковым номером;
- Установки (set) изменяют атрибуты путей и маршруты;
- Вызов другой карты (call) позволяет выполнить правила из другой карты маршрутов;
- Действие (action) определяет поведение после выполнения правила. Если действие отсутствует, то выполнение карты завершается. В качестве действия можно задать переход на другое правило (goto).

Если маршруты не попали ни под одно правило, то они отбрасываются. Чтобы разрешить дальнейшую обработку таких маршрутов, последним в карте должно идти пустое правило с разрешающей политикой.

56.9.2 Создание правил

Создание правил route-map.

Каждое правило в карте маршрутов изменяется отдельно в режиме редактирования. Перейти в режим редактирования правила можно командой:

```
(config)# router route—map <rmap_name> permit|deny <seq>
```

<rmap_name> - имя карты маршрутов, в которую заносится правило.

permit – разрешающая политика правила.

deny - запрещающая политика правила.

<seq> - порядковый номер правила. Рекомендуется задавать порядковый номер через десятки т.е. 10, 20, 30 и т.д. Если в дальнейшем понадобиться вставить новое правило между существующими, можно будет использовать номера 11, 12, 22 и пр. и не придется переписывать всю карту.

Пример создания разрешающего правила под номером 10 для карты с именем myrmap:

```
(config)# router route—map myrmap permit 10
(config—route—map—myrmap)#
```

Удалить правило можно командой:

```
(config)# no router route—map <rmap_name> permit|deny <seq>
```

56.9.3 Просмотр правил

Посмотреть правила, созданные в определенной карте, можно командой:

```
# show router route—map <rmap_name>
```

Команда выдает список правил, разделенный на четыре секции: ZEBRA (статическая маршрутизация), RIP, OSPF и BGP. В каждой секции показан один и тот же набор правил. Критерии и установки правил (match и set), могут меняться в зависимости от секции. Например, критерии и установки, специфические для BGP, будут показаны только в BGP-секции.

Пример вывода правил карты маршрутов:

```
(config—route—map—myrmap)# do show
call  myrmap2
description “first rule”
match ip address myacl
set as—path prepend 65010
set metric 1
# show router route—map myrmap
ZEBRA:
route—map myrmap, permit, sequence 10
  Description:
    ”first  rule”
  Match clauses:
    ip address myacl
  Set clauses:
```

```
  Call clause:
    Call myrmap2
  Action:
    Exit routemap
RIP:
route—map myrmap, permit, sequence 10
  Description:
    ”first  rule”
  Match clauses:
    ip address myacl
  Set clauses:
    metric 1
  Call clause:
    Call myrmap2
  Action:
    Exit routemap
OSPF:
route—map myrmap, permit, sequence 10
  Description:
    ”first  rule”
  Match clauses:
    ip address myacl
  Set clauses:
    metric 1
  Call clause:
    Call myrmap2
  Action:
    Exit routemap
BGP:
route—map myrmap, permit, sequence 10
  Description:
    ”first  rule”
  Match clauses:
    ip address myacl
    peer 192.168.0.3
  Set clauses:
    metric 1
    as—path prepend 65010
  Call clause:
    Call myrmap2
  Action:
    Exit routemap
```

56.9.4 Описание к правилу

Для удобства администрирования можно задать описание к правилу при помощи команды:

```
(config—route—map—myrmap)# description <string>
```

<string> - строка с описанием.

Удалить описание можно командой:

```
(config—route—map—myrmap)# no description
```

56.9.5 Критерии сравнения

56.9.5.1 Интерфейс

Для ZEBRA, RIP, OSPF, BGP.

Сравнение интерфейса, с которого доступен маршрут.

```
(config—route—map—myrmap)# match interface <iface>
```

<iface> - название интерфейса, например ethernet0.

56.9.5.2 IP-адрес назначения

Для: ZEBRA, RIP, OSPF, BGP

Сравнение IP-адреса сети назначения по списку доступа маршрутизатора, либо по префиксному списку:

```
(config—route—map—myrmap)# match ip address <racl_num>|<racl_name>|(prefix—list
        <prlist_name>)
```

<racl_num>|<racl_name> - номер либо имя списка доступа маршрутизатора.

<prlist_name> - имя префиксного списка.

56.9.5.3 IP-адреса шлюза

Для ZEBRA, RIP, OSPF

Сравнение IP-адреса шлюза по списку доступа, либо по префиксному списку:

```
(config—route—map—myrmap)# match ip next—hop <racl_num>|<racl_name\>|(prefix—list
        <prlist_name>)
```

56.9.5.4 IP-адрес маршрутизатора, анонсировавшего маршрут

Для BGP.

IP-адрес маршрутизатора, анонсировавшего маршрут:

```
(config—route—map—myrmap)# match ip route—source <racl_num>|<racl_name>|(prefix—list
        <prlist_name>)
```

56.9.5.5 Метрика маршрута (metric). RIP, BGP

```
(config—route—map—myrmap)# match metric <n>
```

<n> - метрика маршрута.

56.9.5.6 Тег маршрута

Для RIP.

```
(config—route—map—myrmap)# match tag \<n\>
```

<n> - тег маршрута.

56.9.6 Установки

56.9.6.1 Метрика маршрута

Для OSPF, BGP.

Устанавливает метрику, либо изменяет существующую

set metric [+|-]<seq>

<seq> - изменение метрики

56.9.6.2 Источник маршрута

Для ZEBRA.

IP-адрес источника маршрута

set src <ip>

56.9.6.3 Адрес шлюза

Для RIP, BGP.

set ip next-hop <ip>

56.9.6.4 Тег маршрута

Для RIP.

set tag <n>

56.9.7 Вызов другой карты маршрутов

Внутри правила можно вызвать правила из другой карты маршрутов. Сделать это можно командой:

```
(config—route—map—myrmap)# call <rmap_name>
```

Удалить такой вызов можно командой:

```
(config—route—map—myrmap)# no call
```

56.9.8 Переход на другое правило при выполнении условий

Переход на другое правило при выполнении условий (on-match next, goto, continue)

Если совпадений с критериями нет, то просматривается правило со следующим порядковым номером. При выполнении критериев правила, выполняются указанные изменения и просмотр карты маршрутов закачивается. Если требуется продолжить просмотр, следует использовать команду on-match.

Переход на следующее правило при выполнении критериев можно задать командой:

```
(config—route—map—myrmap)# on—match next
```

При выполнении этой команды в конфигурацию вносится строка «on-match goto N», где N – число на единицу большее порядкового номера правила, например для десятого правила N будет 11.

У этой команды существует синоним, команда continue. При выполнении этой команды в конфигурацию все равно запишется «on-match goto N».

Явно указать, на какое правило переходить, можно при помощи команды:

```
(config—route—map—myrmap)# on—match goto <n>
```

<n> - номер правила.

Если правила с таким номером не существует, то переход осуществляется на первое правило с номером большим, чем указанный. Например, есть правила с номерами 10, 20 и 30. Если из десятого правила указать переход на 11-е, то переход произойдет на 20-е.

Внимание! Нельзя переходить на правила выше текущего, например, из десятого правила нельзя переходить на девятое.

У этой команды существует синоним, команда continue <N>. При выполнении этой команды в конфигурацию все равно запишется «on-match goto N».

57. MPLS - многопротокольная коммутация по меткам

57.1 Введение

Dionis-NX имеет базовую поддержку MPLS-маршрутизации. MPLS работает поверх ip-маршрутизации. т.е. для работы MPLS необходимо вначале настроить статическую или динамическую ip-маршрутизацию.

В рамках архитектуры MPLS различают следующие типы устройств:

- LSR-маршрутизатор, поддерживающий коммутацию по меткам и традиционную IP-маршрутизацию. (Маршрутизатор заменяет(swap) или снимает(pop) метки с пакетов).
- LER-маршрутизатор, подключённый к устройствам, не осуществляющим коммутацию по меткам. (Маршрутизатор назначает(push) или снимает(pop) метки с пакетов).
- MPLS-domain - MPLS-домен - группа соединенных устройств осуществляющих коммутацию по меткам, находящихся под единым административным подчинением и функционирующих в соответствии с единой политикой маршрутизации. MPLS домен образуется LSR-ами, а на границе домена размещаются устройства LER.

Для включения поддержки MPLS-маршрутизации необходимо включить MPLS как глобально на маршрутизаторе, так и на необходимых интерфейсах:

```
adm@DionisNX(config)# mpls ip

adm@DionisNX(config)# interface ethernet0
adm@DionisNX(config—if—ethernet0)# mpls ip
adm@DionisNX(config—if—ethernet0)# exit

adm@DionisNX(config)# interface ethernet1
adm@DionisNX(config—if—ethernet1)# mpls ip
adm@DionisNX(config—if—ethernet1)# exit
```

57.2 Статическая MPLS - маршрутизация

Настройка происходит в два этапа:

1. Настройка LER-маршрутизатора (push);
2. Настройка LSR-маршрутизатора (swap/pop).

Настройка LER-маршрутизатора.

```
adm@DionisNX(config)# ip route 192.168.16.0/24 10.1.1.1 label 17
```

Настройка LSR-маршрутизатора.

Замена метки:

```
adm@DionisNX(config)# mpls lsp 17 10.1.1.2 18
```

Снятие метки:

```
adm@DionisNX(config)# mpls lsp 17 10.1.1.2 implicit—null
```

57.3 Динамическая MPLS - маршрутизация

В Dionis-NX динамическое назначение меток происходит при помощи протокола LDP.

Для включения динамической MPLS-маршрутизации необходимо:

1. Настроить динамическую ip-маршрутизацию (например OSPF).
2. Включить поддержку MPLS глобально и на необходимых интерфейсах.
3. Включить поддержку LDP-протокола на необходимых интерфейсах.
4. Указать адрес TCP-соединения для LDP-протокола (адрес одного из интерфейсов, на котором включена поддержка LDP-протокола).

Чтобы войти в режим настроек LDP-протокола в необходимо выполнить команду:

```
adm@DionisNX(config)# router lpd
```

Для включения поддержки LDP-протокола на интерфейсе необходимо выполнить команду:

```
adm@DionisNX(config)# router lpd
adm@DionisNX(router—ldp)# address—family ipv4
adm@DionisNX(router—ldp—af4)# iface ethernet 0
```

Для настройки адреса TCP-соединения для LDP-протокола необходимо выполнить команду:

```
adm@DionisNX(config)# router lpd
adm@DionisNX(router—ldp)# address—family ipv4
adm@DionisNX(router—ldp—af4)# discovery transport—address 192.168.1.1
```

58. VRF - Virtual Routing and Forwarding

58.1 Введение

VRF – технология, позволяющая реализовывать на базе одного физического маршрутизатора несколько виртуальных – каждый со своей независимой таблицей маршрутизации. В основном данная технология применяется для изоляции трафика в сетях провайдера.

Каждый такой виртуальный маршуртизатор — практически является отдельным VPN. Их таблицы маршрутизации, FIB, список интерфейсов и прочие параметры не пересекаются — они строго индивидуальны и изолированы. Ровно так же они обособлены и от самого физического маршрутизатора. Но между ними возможна коммуникация.

Различают два вида VRF: VRF-Lite (VPN без MPLS) и MPLS-L3VPN.

На данный момент в DionisNX реализован поддержка обоих видов:

- VRF-Lite: доступна как статическая маршрутизация, так и динамическая (с использованием OSPF или BGP).
- MPLS-L3VPN: динамическая маршрутизация на основе протокола BGP.

58.2 Базовая настройка

Базовая настройка VRF в DionisNX заключается в создании виртуального интерфейса vrf и добавления к нему других интерфейсов. В данном примере создается виртуальный интерфейс vrf 1 и к нему добавляется реальный интерфейс ethernet 0:

```
adm@DionisNX(config)# interface ethernet 0
adm@DionisNX(config—if—ethernet0)# ip address 192.168.16.2/24
adm@DionisNX(config—if—ethernet0)# enable
adm@DionisNX(config—if—ethernet0)# exit
adm@DionisNX(config)# interface vrf 1
adm@DionisNX(config—if—vrf1)# slave ethernet 0
adm@DionisNX(config—if—vrf1)# enable
```

Примечание: Имя vrf интерфейса имеет значение только в пределах одного устройства.

58.3 VRF-Lite

При создании интерфейса vrf автоматически создается виртуальная таблица маршрутизации. Для редактирования маршрутов в данной таблице необходимо зайти в секцию **router vrf <N>**, где **N** соответствует номеру созданного vrf интерфейса.

```
adm@DionisNX(config)# router vrf 1
adm@DionisNX(config—vrf1)# ip route 10.0.0.0/8 192.168.16.254
```

Для просмотра таблицы маршрутизации для конкретного vrf необходимо выполнить команду:

```
adm@DionisNX# show router vrf 1 ip route
```

Связь между различными vrf и глобальной таблицей маршрутизации возможна через параметр **nexthop-vrf** при создании/редактировании маршрутов. Пример использования маршрутов из глобальной таблицы маршрутизации в таблице маршрутизации vrf 1:

```
adm@DionisNX(config—vrf1)# ip route 10.0.0.0/8 192.168.16.254 nexthop—vrf default
```

Пример использования маршрутов из vrf 2 таблицы маршрутизации в таблице маршрутизации vrf 1:

```
adm@DionisNX(config—vrf1)# ip route 10.0.0.0/8 192.168.16.254 nexthop—vrf 2
```

58.3.1 Пример использования VRF-Lite совместно с OSPF

Ниже приведена схема, реализующая изоляцию сети NET1(NET1_1 + NET1_2) от сети NET2(NET2_1 + NET2_2). Так сеть NET1 использует для передачи своего трафика маршруты VRF1, а сеть NET2 - VRF2. Между узлами настроена динамическая маршрутизация OSPF для каждого VRF. Таким образом на узле запущено одновременно два OSPF процесса (для vrf1 и для vrf2).

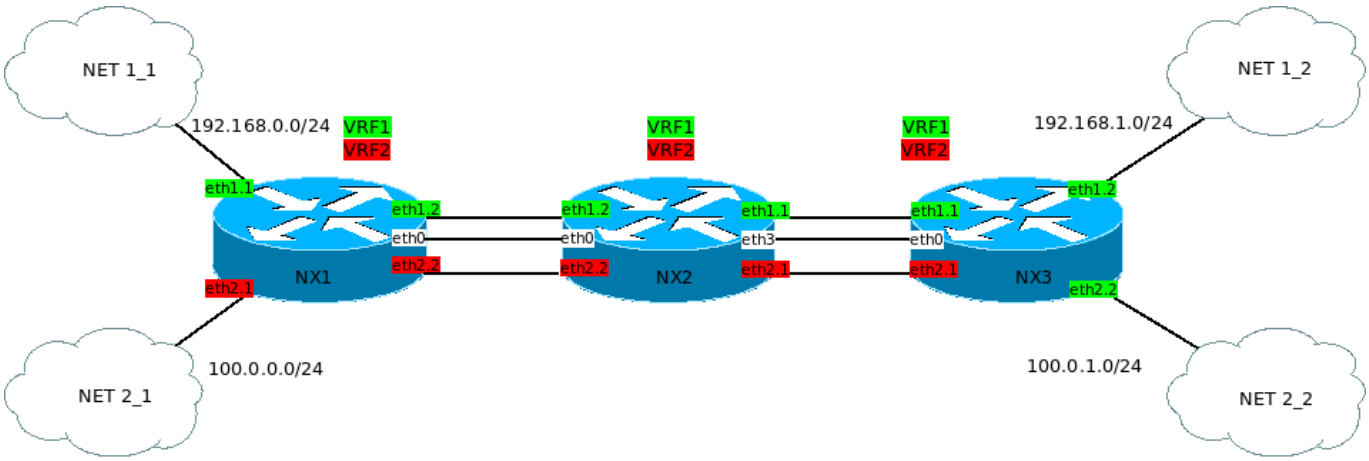

Рис. 58.1: Схема 1: vrf-lite + OSPF

Примеры конфигураций узлов для данной схемы приведены в Приложении.

58.4 MPLS-L3VPN

MPLS L3VPN позволяет избавиться от некоторых этапов настройки VRF-Lite:

- Настройка VRF на каждом узле между точками подключения
- Настройка отдельных интерфейсов для каждого VRF на каждом узле.
- Настройка отдельных процессов IGP для каждого VRF на каждом узле.
- Необходимость поддержки таблицы маршрутизации для каждого VRF на каждом узле.

Процесс настройки можно описать в следующем порядке:

- Настройка базовой связности магистральной сети: IP-адреса, IGP.
- Включение MPLS и LDP
- Создание VRF и привязка к интерфейсам.
- Настройка протокола маршрутизации с CE.
- Настройка BGP и MBGP

Терминология:

CE — Customer Edge router — граничный маршрутизатор клиента, который подключен в сеть провайдера.

PE — Provider Edge router — граничный маршрутизатор провайдера. К нему подключаются CE. На PE начинаются и заканчиваются VPN. На нём расположены интерфейсы, привязанные к VPN. Также PE назначает и снимает сервисные метки. PE являются Ingress LSR и Egress LSR. PE должны знать таблицы маршрутизации каждого VPN, т.к. они принимают решение о том, куда посылать пакет, как в пределах провайдерской сети, так и в плане клиентских интерфейсов.

P — Provider router — транзитный маршрутизатор, который не является точкой подключения — пакеты VPN проходят через него без каких-либо дополнительных обработок, т.е просто коммутируются по транспортной метке. P нет нужды знать таблицы маршрутизации VPN или сервисные метки. На P нет интерфейсов привязанных к VPN.

Для связи PE и P роутеров в MPLS-L3VPN используется семейство адресов ipv4/ipv6 vpn в секции router bgp.

Пример настройки PE роутера для связи с P роутером:

```
router bgp 5226
 neighbor 2.2.2.2 remote—as 5226
 !
 address—family ipv4 unicast
  no neighbor 2.2.2.2 activate
 exit
 !
 address—family ipv4 vpn
  neighbor 2.2.2.2 activate
 exit
!
```

Настройка анонсирования VPN маршрутов на PE роутере происходит в секции address-family ipv4/ipv6 unicast соответствующего VRF:

```
...
router bgp 5224 vrf 1
...
!
 address—family ipv4 unicast
  label vpn export 527
  rd vpn export 5227:1
  rt vpn both 52:27
  export vpn
  import vpn
 exit
!
...
```

Описание команд для анонсирования VPN маршрутов

1.

```
label vpn export <auto|label>
```

Назначение сервисной метки для определения принадлежность к конкретному VPN.

2.

```
rd vpn export <AA:NN>
```

Добавление к анонсируемому маршруту Route Distinguisher для различия маршрутов разных VPN. (В частности, чтобы анонсировать одинаковые префиксы сети от разных клиентов)

3.

```
rt vpn import <AA:NN>
```

Импорт Route Target - для определения к какому VFR относится маршрут

```
rt vpn export <AA:NN>
```

Экспортирование Route Target - для определения к какому VFR относится маршрут

```
rt vpn both <AA:NN>
```

Импорт и экспорт Route Target.

Таким образом на “отдающей” стороне назначается **rt vpn export**, а на “принимающей” должен быть соответствующий **rt vpn import**.

4.

```
export vpn
```

Экспорт маршрутов из address-family unicast в address-family vpn

5.

```
import vpn
```

Импорт маршрутов из address-family vpn в address-family unicast

58.4.1 Пример реализации MPLS-L3VPN

Ниже приведена схема, реализующая изоляцию сети C3PO от сети R2D2. Причем префиксы сетей у клиентов C3PO-1 и R2D2-1 одинаковые.

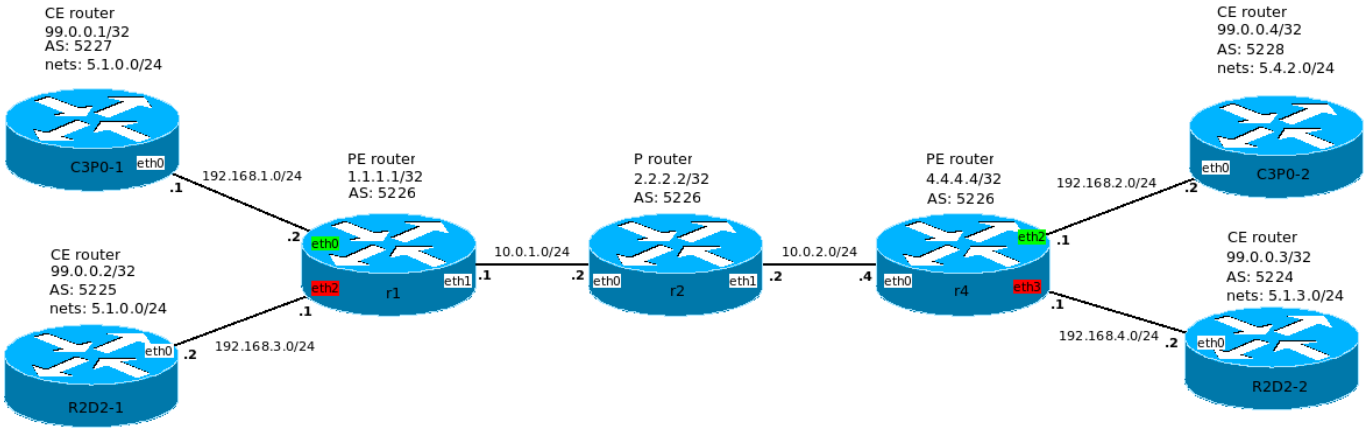

Рис. 58.2: Схема 2: MPLS-L3VPN

Примеры конфигураций узлов для данной схемы приведены в Приложении.

59. Криптография

59.1 Ключ доступа

Для начала работы с криптографическими средствами Dionis-NX необходимо инициализировать датчик случайных чисел (ДСЧ), а также создать ключ доступа (КД).

Начальное заполнение ДСЧ доставляется на внешнем носителе в одном из форматов:

- random.ini - формат хранения симметричных ключей Dionis (необходимы файлы gk.db3, uz.db3, random.ini);
- gk.db3 - формат хранения симметричных ключей Dionis, вариант ДСРФ (необходимы файлы gk.db3, uz.db3);
- PKCS#15 - формат хранения ключевой информации PKCS#15.

Ключ доступа используется для защиты секретов системы, хранящихся на внутреннем носителе, а именно:

- Начальное заполнение ДСЧ для следующей перезагрузки (защита шифрованием);
- Узлы замены для симметричных ключей Disec (защита шифрованием)
- Симметричные ключи Disec (защита шифрованием);
- Предварительно распределённые (pre-shared) ключи (защита шифрованием);
- Закрытые асимметричные ключи (защита шифрованием);
- Корневые X509-сертификаты (защита имитовставкой от подмены).

После генерации ключ доступа должен быть сохранён на внешнем носителе или в постоянной памяти LCD. В случае сохранения на внешнем носителе, данный носитель потребуется при «холодном» перезапуске системы (poweroff/включение). В случае «тёплого» перезапуска (reboot) носитель с КД не потребуется, потому что КД также сохраняется в оперативной памяти LCD.

ВАЖНО: В случае невозможности чтения КД с внешнего носителя из-за технической неисправности следует полностью очистить все данные на носителе без возможности восстановления (или полностью уничтожить носитель). Далее необходимо сгенерировать новый ключ доступа и заново установить в систему все ключи и корневые сертификаты.

ВАЖНО: Утеря или компрометация КД означает компрометацию симметричных и закрытых асимметричных ключей, установленных в данную систему. В этом случае необходимо объявить соответствующие ключи Disec недействительными и уведомить удостоверяющий центр о необходимости отозвать соответствующие сертификаты, закрытые ключи которых были установлены в данную систему. Далее необходимо сгенерировать новый ключ доступа, установить в систему новые ключи и заново установить корневые сертификаты.

Действуют следующие правила:

- Один ключ доступа может относиться только к одному узлу Dionis-NX;
- Один узел Dionis-NX может одновременно иметь только один ключ доступа;

- Ключ доступа может быть сохранён только один раз и на одном носителе (флэш, дискета, LCD, USB-токен/смарт-карта) после генерации или замены;
- На одном носителе может быть только один ключ доступа.

59.1.1 Состояние КД

Чтобы выяснить текущее состояние ключа доступа, нужно выполнить команду (в привилегированном режиме):

```
# show crypto access key status
```

Возможные состояния:

- no key;
- not stored;
- ok.

no key - КД не сгенерирован или не загружен в систему. ДСЧ не инициализирован. Работа криптосистемы невозможна.

not stored - сгенерирован новый КД, но **не** сохранён на **внешнем** носителе. Необходимо выполнить команду ‘crypto access key store’. (См. ниже).

ok - КД загружен в систему. ДСЧ инициализирован.

59.1.2 Генерация нового КД и инициализация ДСЧ

Вставьте внешний носитель с начальным заполнением ДСЧ.

Поддерживаемые внешние носители:

- floppy0 - дискета;
- flashN[.N] - внешний флэш-носитель (пример: flash0, flash1.2);
- token - USB-токен/смарт-карта.

Перейдите в привилегированный режим.

Если на внешнем носителе находится несколько контейнеров PKCS#15, то можно выполнить команду просмотра содержимого внешнего носителя для выяснения имени контейнера:

```
# show crypto access key random—inis flash0:
```

Данная команда выведет доступные контейнеры в корневой директории флэш-накопителя. Примерный формат вывода:

```
subdir1/
subdir2/
...
random.ini    old—dionis
file1         pkcs15
file2         pkcs15
...
```

или

```
subdir1/
subdir2/
...
gk.db3        old—dionis—kb2
file1         pkcs15
file2         pkcs15
...
```

где subdir1, subdir2 - поддиректории; old-dionis, old-dionis-kb2, pkcs15 - форматы контейнеров; random.ini, gk.db3, file1, file2 - имена файлов.

Для просмотра содержимого поддиректорий можно выполнять команды типа:

```
# show crypto access key random—inis flash0:/subdir1/subdir2/subdir3...
```

В случае использования USB-токена/смарт-карты потребуется ввести пин-код пользователя.

ВНИМАНИЕ: При многократном вводе ошибочного пин-кода смарт-карта может заблокироваться. Количество попыток ввода зависит от производителя и настройки смарт-карты.

Для генерации КД следует ввести команду:

```
# crypto access key init flash0:
```

Данная команда выполнит попытку чтения контейнера random.ini/gk.db3 (также необходим файл uz.db3).

Для считывания начального значения ДСЧ из контейнера PKCS#15 необходимо выполнить команду с точным указанием имени файла контейнера. Например:

```
# crypto access key init flash0:/subdir1/file.p15
```

Если контейнер PKCS#15 защищён паролем, то потребуется ввести пароль.

Для считывания начального значения ДСЧ из токена указывать имя PKCS#15 файла не обязательно:

```
# crypto access key init token:
```

ВАЖНО: Если в системе уже присутствовал старый КД, то операция генерации нового КД перезапишет начальное заполнение ДСЧ, сохранённое на **внутреннем** носителе. Новое заполнение ДСЧ будет зашифровано на новом КД. Таким образом, данная операция является необратимой, и использование старого КД становится невозможным. После генерации нового КД необходимо также заново импортировать все необходимые секреты. Для плановой замены КД следует использовать команду crypto access key replace (см. ниже).

Кроме этого, в целях безопасности будет обновлено начальное заполнение ДСЧ на **внешнем** носителе.

Далее обязательно нужно выполнить пункт «Сохранение КД».

59.1.3 Сохранение КД

Если операция генерации прошла успешно, то ключ доступа находится в состоянии «not stored», и его необходимо сохранить на внешнем носителе **или** в постоянной памяти LCD.

Для сохранения в ПЗУ LCD выполните команду:

```
# crypto access key store lcd
```

Для сохранения КД на внешнем носителе **извлеките** носитель с начальным заполнением ДСЧ, вставьте носитель для хранения ключа доступа и выполните команду:

```
# crypto access key store flash0
```

(или “floppy0” для сохранения на дискете).

ВАЖНО: Ключ доступа сохраняется в корневой директории на внешнем носителе в файле «acc-key». Если такой файл уже существует, то он будет заменён (с предварительным уведомлением).

Ключ доступа можно защитить паролем. При сохранении КД будет предложено ввести этот пароль. Если защита КД не требуется, то надо два раза нажать Enter. Если ключ защитить паролем, то его придётся вводить при каждом «холодном» перезапуске. При «тёплом» перезапуске пароль не требуется, так как в ОЗУ LCD ключ доступа сохраняется в открытом виде.

ВНИМАНИЕ: При сохранении КД в ПЗУ LCD рекомендуется всегда защищать КД паролем, чтобы злоумышленник не мог экспортировать КД на внешний носитель с помощью команды ‘crypto access key export’ в случае, когда администратор забыл заблокировать консоль.

Сохранение КД на USB-токене/смарт-карте выполняется командой:

```
# crypto access key store token
```

Ключ доступа, сохранённый на токене, защищается пин-кодом пользователя токена.

Далее необходимо выполнить пункт «Загрузка КД».

59.1.4 Загрузка КД

Сначала необходимо добавить команду «crypto access key load» в сохранённую конфигурацию (startup-config), чтобы ключ доступа загружался каждый раз при перезапуске системы. Для этого нужно выполнить следующие команды:

```
# configure
(config)# crypto access key load
(config)# do write
```

Команда «crypto access key load» осуществляет поиск КД на внешних носителях в следующем порядке: все флэш-носители, дискета, ОЗУ LCD, ПЗУ LCD, USB-токен/смарт-карта. Если требуется иной порядок поиска КД, то можно явно указать список устройств. Например:

```
(config)# crypto access key load lcdram token flash0.1 flash1 lcdrom
```

Примечание: При загрузке КД из токена потребуется ввести пин-код пользователя.

При загрузке КД осуществляется попытка расшифровать начальное заполнение ДСЧ, сохранённое на внутреннем носителе. Неудачное расшифрование означает несоответствие данного КД данному узлу Dionis-NX. При удачном расшифровании происходит инициализация ДСЧ, формируется новое начальное заполнение ДСЧ, которое зашифровывается на КД и помещается на внутренний носитель (для следующей перезагрузки).

Чтобы проверить, успешно ли загрузился КД, нужно выполнить команду «show crypto access key status» (см. выше).

Если при загрузке было выдано сообщение, требующее обновление формата ключевой системы (выполнения команды “crypto access key uprade”), см. раздел “Обновление формата ключевой системы” ниже.

Если по каким-то причинам требуется не осуществлять загрузку КД при перезапуске системы, то нужно удалить команду «crypto access key load» из сохранённой конфигурации, выполнив команды:

```
(config)# no crypto access key load
(config)# do write
```

59.1.5 Обновление формата ключевой системы

При выполнении команды “crypto access key load” система может выдать ошибку:

```
Error: Detected old secrets format. Run ’crypto access key upgrade’ command.
```

Данная ошибка означает, что система была обновлена до новой версии, в которой используется новый формат хранения внутренних секретов.

В этом случае необходимо выполнить команду:

```
# crypto access key upgrade <носитель_с_ключом_доступа>
```

Если ключ доступа защищён паролем, потребуется ввести пароль.

После успешного обновления ключевой системы необходимо:

1) Повторно выполнить загрузку ключа доступа, как описано выше в разделе “Загрузка КД”.
2) Повторно активировать все криптослужбы и криптотуннели.
3) Сохранить конфигурацию.
4) Перезагрузить систему.

59.1.6 Удаление КД

Если по каким-то причинам требуется удалить КД из **оперативной памяти** системы, следует выполнить команду:

```
# crypto access key clear memory
```

Данная команда удаляет ключ доступа из памяти системы и из ОЗУ LCD.

Следующие команды удаляют КД с **внешних носителей** (соответственно ПЗУ LCD, флэш, дискета, токен/смарт-карта):

```
# crypto access key clear lcd
# crypto access key clear flash0
# crypto access key clear floppy0
# crypto access key clear token
```

Для **безопасного** удаления ключей с внешних носителей рекомендуется использовать команду "clear removable" (см. раздел "Обслуживание").

59.1.7 Плановая замена/импорт КД

Для замены ключа доступа необходимо, чтобы старый ключ доступа был загружен в память (см. «crypto access key load» выше).

Для замены КД в ПЗУ LCD выполните команду:

```
# crypto access key replace save—to lcd
```

Для замены КД с сохранением на внешнем носителе вставьте носитель со старым КД и выполните команду:

```
# crypto access key replace save—to flash0
```

(или floppy0 для дискеты, token для USB-токена/смарт-карты).

В случае использования токена КД защищается пин-кодом пользователя токена.

В случае использования флэш/дискеты будет предложено защитить новый КД паролем. Если защита не нужна, нажмите 2 раза Enter.

ВНИМАНИЕ: При сохранении КД в ПЗУ LCD рекомендуется всегда защищать КД паролем, чтобы злоумышленник не мог экспортировать КД на внешний носитель с помощью команды 'crypto access key export' в случае, когда администратор забыл заблокировать консоль.

При замене КД перешифровываются все секреты системы на новом КД.

Если по каким-то причинам замена КД не удалась, то осуществляется откат, и старый КД остаётся в силе.

59.1.8 Экспорт/синхронизация КД в кластерных решениях

В кластерных решениях может потребоваться иметь одинаковый КД на всех узлах кластера. Если на master-узле КД был сгенерирован и сохранён в ПЗУ LCD, то его можно экспортировать на внешний носитель, доставить на slave-узлы и заменить существующий КД на КД с master-узла. Экспорт выполняется командой:

```
[Master]# crypto access key export to flash0
```

Следующая команда выполняет импорт ключа с флэш-носителя в ПЗУ LCD. (При этом все секреты системы перешифровываются на импортированном ключе).

```
[Slave]# crypto access key replace load—from flash0 save—to lcd
```

ВНИМАНИЕ: При сохранении КД в ПЗУ LCD рекомендуется всегда защищать КД паролем, чтобы злоумышленник не мог экспортировать КД на внешний носитель в случае, когда администратор забыл заблокировать консоль.

59.1.9 Плановое обновление начального заполнения ДСЧ

Для обеспечения качества псевдослучайных последовательностей, выдаваемых программным ДСЧ, периодически необходимо инициализировать ДСЧ повторно с помощью нового начального заполнения, доставляемого на внешнем носителе. Периоды повторной инициализации ДСЧ определяются регламентом.

Повторная инициализация ДСЧ внешним начальным заполнением осуществляется командой:

```
# crypto random update <носитель>[:/<путь_к_файлу_pcks15>]
```

Начальное значение ДСЧ может доставляться в форматах “random.ini”, ДСРФ, PKCS15 (см. выше). (См. также описание команд ‘crypto access key init’ и ‘show crypto access key random-inis’ выше).

59.2 Работа с USB-токенами/смарт-картами

В качестве внешних носителей криптографической информации (помимо флэш и дискет) могут использоваться криптографические USB-токены и смарт-карты. Для доступа к USB-токенам/смарт-картам используются специализированные команды.

Примечание: Допускается присутствие одновременно только одного токена или смарт-карты в системе. Если присутствует несколько токенов/смарт-карт, то для обращения будет выбрана один/одна из них непредсказуемым образом.

Для вывода списка файлов, находящихся на токене/смарт-карте, используется команда:

```
# show crypto token
```

При выполнении данной команды будет предложено опционально ввести пин-код пользователя токена. Если пин-код не введён, то будут отображены только файлы, не защищённые пин-кодом. Если пин-код введён, будут отображены все файлы.

ВНИМАНИЕ: При многократном вводе ошибочного пин-кода смарт-карта может заблокироваться. Количество попыток ввода зависит от производителя и настройки смарт-карты.

Удаление файла с токена производится командой:

```
# crypto token rm <имя_файла>
```

Ввод пин-кода для данной команды обязателен.

Следующие команды также могут обращаться к USB-токенам/смарт-картам. Для этого в качестве параметра “внешний_носитель” необходимо указать ключевое слово “token:”.

- show crypto access key random-inis
- crypto access key init
- crypto access key store
- crypto access key load
- crypto random update
- show crypto pki key
- crypto pki import key
- show crypto pki certs
- show crypto pki cert
- crypto pki import cert
- crypto pki import ocsp cert
- crypto pki import ca cert
- crypto pki import root ca cert
- show crypto pki crls
- show crypto pki crl
- crypto pki import crl

Подробное описание данных команд см. в соответствующих разделах.

59.3 Туннели Disec

Dionis-NX имеет возможность создавать туннели DISEC.

59.3.1 Инициализация DISEC

Перед созданием туннеля необходимо инициализировать подсистему DISEC.

Перед инициализацией подсистемы DISEC необходимо загрузить ключ доступа (подробнее см. Криптография. Ключ доступа):

```
# crypto access key init flash0
# crypto access key store lcd
(config)# crypto access key load
```

Первой командой производится инициализация ключа доступа (КД), данные для КД считываются с внешнего ключевого носителя (ВКН), в данном случае с флэшки. Второй командой КД сохраняется в памяти LCD. И,наконец, третья команда определяет, что нужно каждый раз при загрузке системы считывать КД с LCD, куда он ранее был сохранен (см.вторую команду).

После того, как КД был успешно проинициализированван, можно импортировать ключ DISEC:

```
# crypto disec import key flash0:/
```

Если команда успешно выполнилась, будет выведена информация об импортированном ключе:

```
Info: key (serial:55; cn:1) successfully imported
```

В данном случае ключ серии 55 с локальным крипто-номером 1 был успешно импортирован при инициализации DISEC.

Можно импортировать ключи, указав путь хранения ключей на конкретном устройстве, например:

```
# crypto disec import key flash0.1:/path/to/keys
```

Также можно осуществить множественный импорт ключей, указав путь базовой директории на устройстве, например:

```
# crypto disec import key flash0.1:/keys/*
```

Команда произведет импорт ключей (если они есть), находящихся в директориях, которые находятся в директории /keys на устройстве flash0.1. Директории с именами km_k,db1,db2 при этом будут пропущены.

59.3.2 Создание туннеля

Перед описанием процесса создания туннеля DISEC (далее туннеля) введем необходимые понятия, и сопроводим их краткими описаниями:

1. имя туннеля: задает имя туннеля, нужно для идентификации туннеля и для большей наглядности;
2. параметры туннеля:
    - IP-адреса концов туннеля;
    - криптографические параметры туннеля;
3. правила отбора в туннель: задают правила для исходящего сетевого трафика, по которым он попадает в туннель;
4. приоритет правил отбора:
    - чем ниже численно приоритет правила,тем выше в списке правил стоит данное правило;
    - правила просматриваются сверху вниз;

- при попадании исходящей датаграммы в правило, просмотр нижележащих правил данного туннеля прекращается: датаграмма попала в туннель;

5. приоритет туннеля:

   - определяет приоритет набора правил туннеля;
   - чем ниже численно приоритет туннеля,тем выше в списке туннелей стоит данной туннель;
   - наборы правил туннелей просматриваются сверху вних;
   - при попадании исходящей датаграммы в одно из правил в наборе правил туннеля, просмотр нижележащих туннелей прекращается.

Будем считать, что DISEC инициализирован с поддержкой криптографии. Для создания туннеля следует выполнить команды:

```
(config)# crypto disec conn t1
(config—disec—t1)# local 1.1.1.1
(config—disec—t1)# remote 2.2.2.2
(config—disec—t1)# id 1
(config—disec—t1)# serial 55
(config—disec—t1)# local—cn 1
(config—disec—t1)# remote—cn 2
(config—disec—t1)# alg both
```

Туннель добавится в конец списка уже имеющихся туннелей под очередным номером.

Если необходимо поместить туннель в список под другим номером, при создании туннеля можно использовать следующую команду:

```
(config)# 1 crypto disec conn t2
```

Данная команда создаст туннель t2 под номером 1. Ранее созданный туннель t1 переместится ниже под номером 2.

Рассмотрим по порядку параметры туннеля, которые необходимо указать при его создании:

- local: задает IP-адрес локального конца туннеля;
- remote: задает IP-адрес удаленного конца туннеля;
- id: целое число (до 5 цифр), идентифицирующее туннель; значение этого параметра должно совпадать на обоих концах туннеля;
- serial: номер серии ключей - целое десятичное число, равное номеру серии ключей, используемой в данной криптографической сети;
- local-cn: целое число (до 5 цифр), равное номеру данного узла в криптографической сети;
- remote-cn: целое число (до 5 цифр), равное номеру в криптографической сети того узла, с которым будет выполняться обмен информацией по данному туннелю;
- alg both: алгоритм трансформации данных в туннеле; возможные значения:
  - compression: только сжатие данных;
  - encryption: только зашифрование данных;
  - both: и сжатие, и зашифрование данных;
  - none: никакой трансформации данных не производится.

Чтобы удалить ненужный более туннель, следует использовать команду:

```
(config)# no crypto disec conn t1
```

Этой командой удаляется туннель t1.

Чтобы удалить все туннели, следует использовать команду:

```
(config)# no crypto disec conn *
```

Настройка IPv6-туннелей аналогична настройке IPv4-туннелей.

Создание IPv6-туннеля и редактирование его параметров:

```
(config)# 1 crypto disec conn6 t1v6
(config—disec—conn6—t1v6)# local 2001:db8::10:10
(config—disec—conn6—t1v6)# remote 2001:db8::12:11
(config—disec—conn6—t1v6)# id 2
(config—disec—conn6—t1v6)# serial 33
(config—disec—conn6—t1v6)# local—cn 3
(config—disec—conn6—t1v6)# remote—cn 2
(config—disec—conn6—t1v6)# alg both
```

Удаление IPv6 туннеля:

```
(config)# no crypto disec conn6 t1v6
```

59.3.3 Правила отбора туннеля

Правила отбора определяют,какой именно сетевой трафик будет попадать в туннель. Правила отбора туннеля просматриваются сверху вниз: самое приоритетное правило имеет номер 1, следующее правило имеет номер 2 и т.д. Как только для данной датаграммы будет найдено соответствующее правило отбора, принимается решение инкапсулировать датаграмму в туннель. Важную роль в том, в какой именно туннель попадет датаграмма, играет приоритет туннеля. Чем ниже номер туннеля,тем более приоритетен набор правил отбора данного туннеля по отношению к наборам других туннелей. Рассмотрим пример:

```
1 tunnel1
  1 rule1a
  2 rule1b
2 tunnel2
  1 rule2a
  2 rule2b
  3 rule2c
```

В данном примере условными обозначениями определено 2 туннеля tunnel1 и tunnel2, в каждом из которых свой набор правил. При принятии решения, каким именно туннелем пересылать исходящую датаграмму и нужно ли ее вообще пересылать каким-либо туннелем, правила отбора анализируется в следующем порядке: rule1a, rule1b, rule2a,rule2b,rule2c.

При попадании в одно из первых двух правил датаграмма будет пересылаться через tunnel1, при попадании в одно из следующих трех правил - через tunnel2, иначе - датаграмма не будет пересылаться через эти DISEC-тунели.

Теперь перейдем непосредственно к созданию правил отбора.

Формат задания правила следующий: **permit|deny [PROTO] src <SRCIP> dst <DSTIP> [sport <S1> [S2]] [dport <D1> [D2]] [remark <REMARK>]**

Рассмотрим аргументы команды задания правила:

- permit - разрешающее правило: трафик, попадающий в правило, будет обрабатываться данным туннелем;
- deny - исключающее правило: трафик, попадающий в правило, не будет обрабатываться данным туннелем;
- PROTO - протокол, например IP,TCP,UDP,ICMP и др. или номер протокола; по-умолчанию: протокол IP;
- SRCIP - адрес сети или узла отправителя датаграммы;
- DSTIP - адрес сети или узла получателя датаграммы;
- S1,S2 - начальный и, возможно,конечный порт отправителя датаграммы; если указан S2, то значения S1,S2 задают интервал портов; по-умолчанию: любой порт;
- D1,D2 - начальный и,возможно,конечный порт получателя датаграммы; если указан S2, то значения S1,S2 задают интервал портов; по-умолчанию: любой порт;
- REMARK - необязательная пометка(любая строка без пробелов) правила.

Порты S1/S2,D1/D2 можно задать, только если PROTO указывает на tcp- или udp-протокол.

Примеры:

```
(crypto—disec—t1)# permit tcp src 1.1.1.0/24 dst 192.168.2.2/32 sport 20 80 dport 100 200 remark
        GOOD
(crypto—disec—t1)# deny src 2.2.0.0/16 dst 192.168.2.2/32 remark BETTER
(crypto—disec—t1)# 1 permit icmp src 2.2.0.0/16 dst 192.168.2.2/32 remark BEST
(crypto—disec—t1)# no 2
(crypto—disec—t1)# no all
```

Первые три команды задают правила отбора. Четвертая команда удаляет правило под номером 2, помеченное как GOOD. Пятая команда удаляет все правила.

Логика настройки правил отбора IPv6-туннелей практически идентична настройке IPv4-туннелей за исключением используемых адресов для параметров “src” и “dst” (используются IPv6-адреса).

59.3.4 Включение и выключение туннеля

Вышеописанные параметры и правила отбора в туннель не будут действовать,пока вы не включите туннель:

```
(config—disec)# crypto disec enable conn t1
```

Если пристутсвуют лицензионные ограничения на количество включенных туннелей, вы не сможите включить больше определенного числа туннелей. Узнать число доступных к включению туннелей можно по команде show crypto disec license.

Когда туннель включен вы можете менять его параметры и правила отбора. Если эти параметры имеют смысл, они будут автоматически применены к данному туннелю.

Чтобы выключить туннель, следует использовать команду:

```
(config—disec)# crypto disec disable conn t1
```

Чтобы включить все туннели, следует использовать команду:

```
(config—disec)# crypto disec enable conn *
```

Чтобы выключить все туннели, следует использовать команду:

```
(config—disec)# crypto disec disable conn *
```

В случае,если какой-либо туннель А не сможет отключиться, будет предпринята попытка включения успешно отключенных туннелей,если таковые были до момента сбоя в отключении туннеля А. Таким образом, в случае успешности данной команды ВСЕ туннели будут отключены. В случае неуспешности - ничего не изменится.

Чтобы включить определенные туннели, заданные по ID выполните команду:

```
(config—disec)# crypto disec enable id IDSTR
```

Параметр IDSTR имеет формат <N|N-M>[,{N|N-M}] , т.е. это набор из 1 или более подстрок, разделенных запятыми, где каждая подстрока - это число (ID туннеля) или интервал чисел (ID туннелей),заданный через дефис, причем левая граница интервала должна быть меньше либо равна правой. Неверные подстроки пропускаются.

Пример: команда crypto disec enable id 1,4-7,9-9,23-32„,5 - попытается включить туннели со следующими ID: 1,4,5,6,7,9, если указанные ID заданы в туннелях.

Чтобы выключить определенные туннели, заданные по ID выполните команду:

```
(config—disec)# crypto disec disable id IDSTR
```

Включение и выключение IPv6-туннелей идентичны тем же операциям с IPv4-туннелями:

```
(config—disec)# crypto disec enable6 id IDSTR
```

и

```
(config—disec)# crypto disec disable id IDSTR
```

59.3.5 Копирование и перемещение туннеля

crypto disec copy <OLD> <NEW> [id <ID>] [PRF] [force] [rules]

Данная команда осуществляет копирование существующего туннеля в новый вместе со всеми параметрами и,возможно, правилами отбора. Новый туннель в случае успешного копирования будет находиться в состоянии **выключен**.

Параметры:

- OLD : старое имя туннеля;
- NEW : новое имя туннеля;
- ID : id, присваиваемый новому туннелю;
- PRF : приоритет, под которым следует создать новый туннель;
- force : если туннель NEW уже существует, он будет отключен и его параметры станут равными параметрам туннеля OLD, т.е. произойдет замена туннеля;
- rules : копировать также и правила туннеля.

crypto disec move <NAME> <PRF>

Данная команда осуществляет перемещение существующего туннеля NAME: туннелю присваивается новый приоритет PRF. Состояние туннеля (включен/выключен) сохраняется.

Для IPv6-туннелей команда будет иметь вид:

crypto disec move6 \<NAME\> \<PRF\>

59.3.6 Работа с ключами

В данном разделе описывается просмотр,удаление и добавление ключей и абонентов.

59.3.6.1 Просмотр ключей

Чтобы посмотреть установленные в систему крипто-ключи, следует использовать команду:

```
# show crypto disec key
```

Пример вывода команды:

```
Installed keys:
Serial: 55 , locals: 1
```

Из вывода следует, что установлен один ключ с локальным криптономером 1 серии 55.

Чтобы посмотреть установленные в систему крипто-ключи определенной серии, следует использовать команду:

```
# show crypto disec key 55
```

Пример вывода команды:

```
Installed keys for serial 55: 1
```

Из вывода следует, что установлен один ключ с локальным криптономером 1 для серии ключей 55.

59.3.6.2 Просмотр абонентов

Чтобы посмотреть доступность абонента с крипто-номером 2 для доступа по ключу с серией 55 и крипто-номером 1, следует выполнить команду:

```
# show crypto disec abonent 55 1 2
```

Если абонент доступен, выдача команды будет следующей:

```
Access to abonent 2 for key (sn=55;loc=1), check status: GRANTED
```

Если абонент не доступен, выдача команды будет следующей:

```
Access to abonent 2 for key (sn=55;loc=1), check status: DENIED
```

59.3.6.3 Добавление и удаление ключей

В инициализированную подсистему DISEC вы можете добавлять новые и удалять старые крипто-ключи.

Чтобы добавить новый ключ, нужно вставить ВКН, например, флэшку, хранящую новый крипто-ключ, и выполнить команду:

```
# crypto disec import key flash0:/
```

Чтобы удалить установленный ключ, следует использовать команду:

```
# crypto disec remove key 55 1
```

Команда удалит ключ 1 серии 55.

Чтобы удалить все установленные ключи, следует использовать команду:

```
# crypto disec remove key all
```

59.3.6.4 Полное удаление DISEC

Полное удаление из системы всей информации, относящейся к DISEC, состоит в следующих шагах:

- удаление туннелей командой no crypto disec conn <NAME> (режим configure);
- удаление ключей командой: crypto disec remove key <SERIAL> <LOCAL> (режим enable);
- окончательная очистка от DISEC: crypto disec cleanup (режим enable).

Примечание: Для безопасного удаления ключей с внешних носителей рекомендуется использовать команду “clear removable” (см. раздел “Обслуживание”).

59.3.6.5 Плановая смена ключей

При плановой смене ключей (далее ПС) производится замена сетевых ключей одной из серий на ключи новой серии, которая может либо уже иметься в системе, либо быть импортирована с внешнего носителя.

ПС должна быть выполнена на всех узлах, входящих в криптографическую сеть с данной серией ключей.

Рассмотрим процесс ПС на одном из концов туннеля на примере.

Предположим, что у нас имеются следующие входные данные:

- туннель Disec с именем: TUN;
- туннель TUN использует серию ключей с номером SER0;
- туннель TUN использует локальный криптономер LOC0;
- туннель TUN использует удаленный криптономер REM0;
- туннель TUN может быть включен или выключен.

Задача: для туннеля TUN осуществить ПС:

- заменить серию ключей SER0 на серию ключей SER1;
- криптономер LOC0 на LOC1;
- криптономер REM0 на REM1;
- новые криптономера LOC1 и/или REM1 могут остаться прежними, в этом случае LOC1=LOC0 и/или REM1=REM0.

В данном случае алгоритм смены ключей следующий:

1. В случае, если производится удаленная ПС, то необходимо войти в систему, на которой нужно осуществить ПС, по защищенному туннелю. Например, если это туннель Disec, то нужно иметь для этого туннеля специальную серию ключей, используемую только для процедуры ПС.

2. В случае, если производится локальная ПС, то защищенный туннель для ПС не нужен - администратор просто заходит в систему локально, используя свое имя пользователя и пароль.

3. После входа в систему (локально или удаленно), необходимо удостовериться в наличии новой серии ключей SER1, выполнив следующую команду:

```
DionisNX# show crypto disec key SER1
```

Выдача команды покажет, какие локальные криптономера доступны для данной серии ключей. Необходимо удостовериться, что новый локальный криптономер LOC1 туннеля TUN перечислен в выдаче данной команды.

Если LOC1 не найден в выдаче команды, значит ПС не будет выполнена успешно.

4. Далее необходимо проверить, возможен ли доступ по данной серии ключей к нужному удаленному абоненту REM1, выполнив следующую команду:

```
DionisNX# show crypto disec abonent SER1 LOC1 REM1
```

Если удаленный абонент является доступным по данной серии ключей, то выдача данной команды будет следующей:

```
Access to abonent REM1 for key (sn=SER1;loc=LOC1): GRANTED.
```

Если удаленный абонент является недоступным по данной серии ключей, то выдача данной команды будет следующей:

```
Access to abonent REM1 for key (sn=SER1;loc=LOC1): DENIED.
```

Если удаленный абонент недоступен (команда выдала DENIED), значит ПС не будет выполнена успешно.

5. Затем следует войти в настройки туннеля TUN и выполнить следующие команды:

```
DionisNX# configure
DionisNX(config)# crypto disec conn TUN
DionisNX(config—disec—TUN)# serial SER1
```

Если криптономер LOC1 не равен LOC0, то дополнительно необходимо будет выполнить следующую команду:

```
DionisNX(config—disec—TUN)# local—cn LOC1
```

Если криптономер REM1 не равен REM0, то дополнительно необходимо будет выполнить следующую команду:

```
DionisNX(config—disec—TUN)# remote—cn REM1
```

6. На этом плановая смена ключей завершена. Пункты 3 и 4 алгоритма не обязательны и нужны исключительно для проверки того, что туннель будет корректно работать на новой серии ключей.

После проверки связи со всеми удаленными узлами криптографической сети, необходимо удалить старую серию ключей следующей командой:

```
DionisNX# crypto disec remove key SER0 LOC
```

Эту команду следует повторить для каждого локального криптономера в серии SER0 до полного удаления старой серии ключей SER0 из системы.

59.3.6.6 Действия при неплановой смене ключа доступа

Если ключ доступа после импорта ключей был изменен способом, отличным от «Плановой замены КД», то для корректной работы туннелей необходимо:

- удалить установленные ключи;
- выполнить crypto disec cleanup;
- вновь осуществить импорт нужных ключей.

59.3.6.7 Удаление абонентов

Чтобы заблокировать возможность криптографической связи с абонентом, определяемым удаленным крипто-номером, следует удалить его из локального ключа:

```
# crypto disec remove abonent 55 1 10
```

Команда навсегда блокирует абонента 10 для ключа 1 серии 55. Чтобы разблокировать абонента, необходимо удалить и снова добавить ключ 1 серии 55 с ВКН.

59.3.7 Работа с туннелями

59.3.7.1 Просмотр туннелей

Чтобы посмотреть таблицу имеющихся туннелей, следует использовать команду:

```
# show crypto disec conns
```

Пример выдачи команды:

```
[#]NAME       ID   SRC          DST          SN      LOC   REM   A B
t1            4    192.168.2.1  192.168.2.2  —       —     —     N N
#t2           6    192.168.4.1  192.168.4.2  55      1     2     E N
```

Рассмотрим столбцы таблицы:

- [#]NAME - имя туннеля; если перед именем стоит знак #, значит туннель выключен;
- ID - идентификатор туннеля;
- SRC - адрес локального конца туннеля;
- DST - адрес удаленного конца туннеля;
- SN - номер серии ключей;
- LOC - локальный крипто-номер туннеля;
- REM - удаленный крипто-номер туннеля;
- A - тип трансформации данных в туннеле: E - шифрование, C - компрессия, B - шифрование и компрессия, N - нет трансформации;
- B - туннель заблокирован: Y - да, N - нет.

Чтобы посмотреть информацию по конкретному туннелю, следует использовать команду:

```
# show crypto disec conn t1
```

и для IPv6-туннелей:

```
# show crypto disec conn6 t1
```

В дополнение к информации, которая была рассмотрена для предыдущей команды, будут выведены правила данного туннеля, например:

```
[#]NAME        ID    SRC           DST           SN        LOC   REM   A B
id1            11    192.168.0.5   192.168.0.4   1          2     3    E N
1 permit src 0.0.0.0/0 dst 192.168.32.0/24
2 permit src 0.0.0.0/0 dst 192.168.16.0/24
3 permit src 0.0.0.0/0 dst 192.168.3.0/24
```

59.3.7.2 Просмотр правил отбора

Чтобы посмотреть информацию по конкретному туннелю, следует использовать команду:

```
# show crypto disec rules id1
```

Пример выдачи команды:

```
1 permit src 0.0.0.0/0 dst 192.168.32.0/24
2 permit src 0.0.0.0/0 dst 192.168.16.0/24
3 permit src 0.0.0.0/0 dst 192.168.3.0/24
```

59.3.7.3 Блокирование туннеля

Иногда бывает необходимо полностью заблокировать трафик через включенный туннель. Это делается следующей командой:

```
(config—disec—t1)# block
```

Чтоб разблокировать трафик через туннель, следует использовать команду:

```
(config—disec—t1)# no block
```

59.3.8 Прочая работа с DISEC

59.3.8.1 Привязка процедур шифрования к работе на определенных ядрах процессора

Для балансировки шифрования по ядрам процессора используйте команду:

```
(config)# crypto disec affinity 0—1,3
```

где в качестве параметров выступает сумма диапазонов и номеров ядер процессора.

59.3.8.2 Синхронный и асинхронный режим обработки пакетов

По умолчанию ядро пытается распараллеливать обработку входящих и исходящих DISEC-пакетов. Зашифрование/дешифрование пакетов производится на разных процессорах/ядрах. В результате пакеты могут отправляться/попадать в систему фактически не в том порядке, в котором они были отправлены/получены. Если такое поведение неприемлемо, то можно включить синхронный режим обработки:

```
(config)# crypto disec mode sync
```

Вернуть поведение по умолчанию (асинхронный режим) можно с помощью команды:

```
(config)# no crypto disec mode
```

Или:

```
(config)# crypto disec mode async
```

59.3.8.3 Инкапсуляция DISEC в UDP

Для инкапсуляции датаграмм DISEC в UDP-датаграммы в дополнение к основным параметрам туннеля следует использовать следующую команду:

```
(config—disec—t1)# encap sport 500 dport 600
```

Параметры sport и dport - номер UDP-порта отправителя и получателя, соответственно, они не обязательны. По умолчанию равны 500.

59.3.8.4 Блокирование DISEC трафика

Чтоб полностью заблокировать трафик через все включенные туннели, следует использовать команду:

```
(config—disec)# crypto disec block
```

Чтоб разблокировать трафик через все включенные туннели, следует использовать команду:

```
(config—disec)# no crypto disec block
```

Трафик будет разблокирован только для тех туннелей, которые не заблокированны индивидуально командой **block**.

59.3.8.5 Просмотр состояния

Чтобы посмотреть версию подсистемы DISEC, следует использовать команду:

```
# show crypto disec version
```

Чтобы посмотреть статистику обработки пакетов через подсистему DISEC, следует использовать команду:

```
# show crypto disec statistic
```

и для IPv6-туннелей:

```
# show crypto disec statistic6
```

Рассмотрим пример выдачи данной команды:

```
TUNNEL       XP_OUT   XP_IN    XP_FWD   XS_OUT         XS_IN
tun1         17:49:20 17:49:20 17:49:20 1235557106676   108258340540
tun2         17:49:20 17:49:20 17:49:20 1302550192329   609146127844
```

Рассмотрим столбцы данной таблицы:

- TUNNEL : имя туннеля;
- XP_OUT : последнее время попадания исходящей датаграммы в туннель;
- XP_FWD : последнее время попадания в туннель входящей датаграммы, не предназначенной для текущей системы;
- XP_IN : последнее время попадания в туннель входящей датаграммы, предназначенной для текущей системы;
- XS_OUT : число байт исходящего трафика, прошедшего через туннель;
- XS_IN : число байт входящего трафика, прошедшего через туннель.

Чтобы посмотреть лицензионные ограничения на подсистему DISEC, следует использовать команду:

```
# show crypto disec license
```

В настоящее время поддерживается только лицензионное ограничение на количество включенных туннелей DISEC.

59.3.8.6 Режим отладки

При возникновении проблем в работе каких-либо команд подсистемы DISEC можно включить режим отладочных сообщений:

```
(config—disec)# crypto disec debug
```

Более глубокий режим отладки можно включить при помощи команды:

```
(config—disec)# crypto disec debug trace
```

59.4 Туннельные интерфейсы Ditun (ditun, ditap, ip6ditun, ip6ditap)

Туннельные интерфейсы Ditun позволяют создавать криптографически защищенные каналы связи, трафик в которые отбирается по правилам маршрутизации, в отличии от туннельных соединений Disec, трафик в которые отбирается по правилам отбора.

59.4.1 Создание интерфейса

Внимание! Для включения в интерфейсе ditun (ditap) режима шифрования в Dionis-NX должен быть предварительно создан ключ доступа и загружены ключи абонентов Disec (см. Туннели Disec, работа с ключами) или загружены ключи Dikey (см. Работа с ключами Dikey). Для работы интерфейса в открытом режиме (без шифрования) это не является необходимым.

Для создания интерфейса, из режима конфигурации выполните команду:

```
(config)# interface ditun 0
```

В качестве номера интерфейса (в примере – 0) может быть выбрано любое число (натуральное или 0). После этого, в режиме конфигурации интерфейса необходимо задать индентификатор туннеля (id), локальный (local) и удаленный (remote) IP-адреса – концы туннеля, например:

```
(config—if—ditun0)# id 1
(config—if—ditun0)# local 1.1.1.1
(config—if—ditun0)# remote 2.2.2.2
(config—if—ditun0)# enable
```

Параметр enable – делает созданный интерфейс активным.

При использовании схемы Disec необходимо дополнительно ввести параметры аналогичные параметрам туннелей Disec, алгоритм (alg), номер серии ключей (serial), локальный (local-cn) и удаленный (remote-cn) крипто-номера, например:

```
(config—if—ditun0)# id 1
(config—if—ditun0)# alg encrypt
(config—if—ditun0)# local 1.1.1.1
(config—if—ditun0)# remote 2.2.2.2
(config—if—ditun0)# serial 1
(config—if—ditun0)# local—cn 1
(config—if—ditun0)# remote—cn 1
(config—if—ditun0)# enable
```

При использовании схемы Dikey необходимо дополнительно ввести алгоритм (alg) и Dikey-ключи для локального и удаленного абонента (dikey), например:

```
(config—if—ditun0)# id 1
(config—if—ditun0)# alg encrypt
(config—if—ditun0)# local 1.1.1.1
(config—if—ditun0)# remote 2.2.2.2
(config—if—ditun0)# dikey Key1 Key2
(config—if—ditun0)# enable
```

Интерфейс будет создан в тот момент, когда будет задана минимально необходимая информация.

Внимание! При одновременном указании параметров Disec (local-cn, remote-cn, serial) и Dikey (dikey) будет использована схема Disec.

Для создания IPv6-интерфейса, из режима конфигурации выполните команду:

```
(config)# interface ip6ditun 0
```

Настройка IPv6-интерфейсов аналогична настройке IPv4-интерфейсов за исключением используемых адресов (используются IPv6-адреса).

```
(config—if—ip6ditun0)# id 10
(config—if—ip6ditun0)# alg encrypt
(config—if—ip6ditun0)# local 2001:db8::2
(config—if—ip6ditun0)# remote 2001:db8::3
(config—if—ip6ditun0)# serial 7
```

```
(config—if—ip6ditun0)# local—cn 1
(config—if—ip6ditun0)# remote—cn 3
(config—if—ip6ditun0)# enable
```

Для удаления IPv6-интерфейса, из режима конфигурации выполните команду:

```
(config)# no interface ip6ditun 0
```

59.4.2 Версия криптографического алгоритма (протокола)

В новом протоколе Disec используется режим шифрования MGM-Encrypt, основанный на ГОСТ Р 34.15–12 и Р 1323565.1.026–2019. Кроме того, удалось уменьшить размер заголовка по сравнению со старым протоколом. Для включения новой версии криптографического алгоритма используйте команду:

```
(config—if—ditun0)# proto v2
```

Для возвращения на старую версию криптографического алгоритма (протокола) используйте команду:

```
(config—if—ditun0)# proto v1
```

Либо команду:

```
(config—if—ditun0)# no proto
```

Новый криптографического алгоритм несовместим со старым, то есть нельзя создать туннель, у которого на одном конце будет старый криптографический алгоритм (proto v1), а на другом – новый (proto v2). При этом возможна одновременная работа нескольких туннелей, часть из которых использует старый, а часть – новые криптографические алгоритмы (протоколы).

59.4.3 Настройка интерфейса

Настройка автоматического назначения идентификатора туннеля. Возможно использовать автоматическую генерацию id туннеля, с помощью команды:

```
(config—if—ditun0)# id auto
```

При этом, в качестве id будет выбран:

- Номер интерфейса (если интерфейс не в режиме шифрования);
- Локальный крипто-номер + удаленный крипто-номер (если интерфейс в режиме DISEC-шифрования);
- Сгенерированный номер на основании DIKEY ключей (если интерфейс в режиме DIKEY-шифрования);

Какой именно id туннеля используются можно узнать с помощью команды show interface ditun.

Важно! При использовании централизованной системы управления для сбора и анализа конфигурационных файлов не рекомендуется настраивать id auto, так как система управления не сможет определить id туннеля.

Настройка инкапсуляции UDP. Возможно использовать инкапсуляцию в UDP, если провайдер не пропускает протокол инкапсуляции IP в IP. Для создания туннеля с инкапсуляцией в протокол UDP, необходимо выполнить команду encap, с указанием портов концов туннеля (отправитель и получатель), например:

```
(config—if—ditun0)# encap 505 505
```

Значение удаленного порта можно не указывать, если он совпадает с портом источника.

Настройка NAT-traversal. Возможно работать через NAT, когда адрес/порт удаленного конца туннеля заранее неизвестен. Для этого, на одном конце туннеля можно указать значение “any” для параметров удаленного порта и удаленного адреса. В этом случае, туннельный интерфейс будет ожидать входящий пакет от любого IP-адреса с любого порта. Значения порта и адреса, полученные из первого входящего пакета, будут автоматически применены и интерфейс сможет передавать данные. Пример:

```
(config—if—ditun0)# remote any
(config—if—ditun0)# encap 505 any
```

Какой именно IP адрес и порт используются при отправке датаграмм можно узнать с помощью команды show interface ditun.

Настройка keepalive. Интерфейсы ditun поддерживают механизм пинг-проб, при этом, отсутствие ответной пробы меняет состояние несущей и переводит интерфейс в режим no-carrier. Настройки проб аналогичны настройкам проб в GRE туннелях, но имеют небольшие отличия, например:

```
(config—if—ditun0)# keepalive 5 0 src 10.0.0.1 dst 10.0.0.2
(config—if—ditun0)# link—detect
```

В данном примере src и dst устанавливают IP-адрес источника и IP-адрес назначения пинг-проб и являются опциональными. Если они не указаны явно, IP-адресам источника и назначения присваиваются значения IP-адресов концов туннеля.

Дополнительные настройки. Возможно задать IP-адрес (или несколько адресов) интерфейсу ditun и использовать его в маршрутизации/NAT/фильтрах, а также делать другие действия, которые являются допустимы по отношению к интерфейсу, например:

```
(config—if—ditun0)# ip address 10.0.0.1/24
(config—if—ditun0)# ttl 32
(config—if—ditun0)# ip access—group wan in
(config—if—ditun0)# do show interface ditun 0
(config—if—ditun0)# do tcpdump ditun0 numeric
```

Как и в случае с любыми другими интерфейсами, вы можете объединять ditun интерфейсы в bond интерфейс, настраивать политику QoS, применять NAT и фильтры и т.д.

Вы можете менять режим работы обработки пакетов (асинхронный/синхронный) с помощью команды: crypto disec mode, как описано в главе “Туннели Disec”.

Настройка IPv6-интерфейсов аналогична настройке IPv4-интерфейсов за исключением используемых адресов (используются IPv6-адреса).

59.4.4 Маршрутизация

Для направления трафика в интерфейс ditun используются правила маршрутизации. Правила могут быть заданы как и относительно самого интерфейса, так и относительно адреса удаленного интерфейса ditun (не путать с адресом удаленного конца туннеля!), например:

```
(config)# ip route default ditun 0
(config)# ip route 192.168.0.0/24 10.0.0.2
```

При этом во втором случае (маршрутизация через адрес удаленного ditun интерфейса) и примененных настройках: keepalive и link-detect, в случае недостижении пинг пробами удаленной стороны, маршрут будет динамически деактивирован (а при вознобновлении прохождения проб – активирован снова).

При нахождении ditun интерфейса в составе bond интерфейса, попадание трафика в туннель определяется правилами маршрутизации для bond интерфейса и режимом работы bond интерфейса.

59.4.5 Удаление интерфейсов

Для удаления интерфейса выполните команду no interface ditun, например:

```
(config)# no interface ditun 0
```

59.4.6 Работа с ключами Dikey

59.4.6.1 Базовые команды работы с ключами Dikey

Внимание! Для работы с ключами Dikey в Dionis-NX должен быть предварительно создан ключ доступа.

Ключи Dikey, это ключевые пары, которые состоят из закрытого ключа и открытого ключа. Для создания криптографически защищенного канала обмена абонентам необходимо обменяться открытыми ключами способом, исключающим подмену ключей.

Создание (генерация) новой ключевой пары и добавление ее в хранилище Dikey-ключей выполняется командой:

```
# crypto dikey generate <NAME>
```

Удаление ключа выполняется командой:

```
# crypto dikey remove <NAME>
```

Удаление всех неиспользуемых ключей выполняется командой:

```
# crypto dikey remove unused
```

Удалить ключ из хранилища можно только если он не используется. При удалении ключа удаляется как открытый, так и закрытый ключ, при его наличии.

Посмотреть информацию обо всех доступных ключах, а также о том, какие ключи используются можно командой:

```
# show crypto dikey
```

Пример вывода команды

```
+ myKey1 : ditun1
— myKey2
— myKey3 : ditun1
```

Символ "+" перед именем ключа означает, что для данного ключа доступен закрытый ключ. После двоеточия перечислены туннели в которых используется данный ключ.

Посмотреть шестнадцатеричное значение конкретного открытого ключа можно командой:

```
# show crypto dikey <NAME>
```

Кроме того возможно сделать дамп-файл с шестнадцатеричным значением открытого ключа. Для этого необходимо ввести команду:

```
# show crypto dikey <NAME> <FILE>
```

59.4.6.2 Работа с открытыми ключами Dikey

Экспорт (сохранение) открытого ключа/ключей на диск в контейнере, содержащем имя ключа и сам ключ в шестнадцатеричном виде выполняется командой:

```
# crypto dikey export
```

Импорт (загрузка) открытого ключа/ключей из контейнера выполняется командой:

```
# crypto dikey import
```

Добавление (создание) открытого ключа из шестнадцатеричного значения выполняется командой:

```
# crypto dikey add
```

Пример команды для экспорта всех открытых ключей на флеш-диск.

```
# crypto dikey export * flash0:/AllKeys.list
```

Пример команды для экспорта ключа myKey3 на флеш-диск.

```
# crypto dikey export myKey3 flash0:/myKey3.pub
```

Пример команды для импорта открытых ключей с флеш-диска.

```
# crypto dikey import flash0:/AllKeysPub.list
```

Для создания (добавления) только открытого ключа в хранилище ключей необходимо ввести команду:

```
# crypto dikey add <HEX_VALUE | FILE> <NAME>
```

где

<HEX_VALUE> - ключ в виде шестнадцатеричного значения длинной 256 знаков;

<FILE> - дамп-файл, содержащий шестнадцатеричное значение открытого ключа;

<NAME> - будущее имя открытого ключа.

59.4.6.3 Работа с ключевыми парами Dikey

При необходимости работы с ключевыми парами, например при переносе ключевых пар с узла, выводящегося из эксплуатации, на новый узел предназначены команды

```
# crypto dikey store
# crypto dikey load
```

Пример команды для сохранения всех ключевых пар на флеш-диск в общий контейнер.

```
# crypto dikey store * flash0:/AllKeysPriv.list
```

Пример команды для загрузки ключевых пар с флеш-диска и контейнера.

```
# crypto dikey load flash0:/AllKeysPriv.list
```

59.4.6.4 Работа с ключами Dikey в кластерных решениях

При использовании Dikey в кластерных решениях возникает необходимость иметь одинаковые ключи на всех узлах кластера.

Внимание! Для обеспечения такой возможности необходимо предварительно синхронизировать ключ доступа на всех узлах кластера (см. соответствующий раздел документации).

Перенос ключей возможен напрямую от master-узла на slave-узел, для этого можно использовать копирование ключей в файловом пространстве dikey:. Пространство dikey: является скрытым и не отображается клавишей <Tab>, нужно набирать руками полностью.

Например, если собран кластер master-slave с использованием кластерных IP-адресов 192.168.1.1 и 192.168.1.2 и выполнена синхронизация ключа доступа, то для master-узла можно создать ключевую пару с именем nx-1 командой:

```
# crypto dikey generate nx—1
```

Эту ключевую пару можно скопировать на slave по SSH по кластерному линку на порт 922, сначала закрытый ключ командой:

```
# ssh put dikey:/private/nx—1.key adm 192.168.1.2 dikey:/private/nx—1.key port 922
```

Затем открытый ключ командой:

```
# ssh put dikey:/public/nx—1.pub adm 192.168.1.2 dikey:/public/nx—1.pub port 922
```

Аналогичным образом следует скопировать все ключевые пары и открытые ключи, полученные от соседних узлов. После такого копирования при переходе на Slave-узел, крипто-туннели, использующие dikey, продолжат работу.

59.4.6.5 Работа с имитозащитой ключей Dikey

Для удобства работы при импорте/экспорте открытых ключей предусмотрен механизм имитозащиты (электронной подписи) ключей и ее проверки (данный механизм позволяет подписывать и проверять любые файлы). Предполагается, что на узле A уже сформирована ключевая пара и открытый ключ из этой пары также добавлен на узел B.

Для подписи файла используется команда:

```
# crypto dikey sign <FILE_NAME> <KEY> <PATH_TO_SIG>
```

где:

<FILE_NAME> - подписываемый фаил

<KEY> - закрытый ключ

<PATH_TO_SIG> - директория, куда будет помещена полученная подпись. При этом имя файла с подписью будет <FILE_NAME>.sig

Для проверки подписи используется команда:

```
# crypto dikey verify <FILE_NAME> <KEY> <SIG_FILE>
```

<KEY> - открытый ключ

<FILE_NAME> - подписанный файл

<SIG_FILE> - подпись

59.4.6.6 Алгоритм распространения ключей Dikey

1. Создать на узлах новые ключевые пары (crypto dikey generate).
2. Экспортировать открытые ключи (crypto dikey export).
3. Обменяться открытыми ключами.
4. Импортировать полученные открытые ключи (crypto dikey import).

59.4.7 Интерфейсы ditap

Интерфейсы ditap/ip6ditap, в отличии от ditun/ip6ditun, работают не на сетевом, а на канальном уровне модели OSI. Они предназначены для инкапсуляции кадров канального уровня в транспортные пакеты сетевого уровня для организации L2VPN, имеют свой MAC-адрес и обладают возможностями реального сетевого интерфейса. Настройка ditap/ip6ditap полностью аналогична настройке ditun/ip6ditun интерфейсов. Например:

```
(config)# interface ditap 0
(config—if—ditap0)# id 1
(config—if—ditap0)# alg encrypt
(config—if—ditap0)# local 1.1.1.1
```

```
(config—if—ditap0)# remote 2.2.2.2
(config—if—ditap0)# serial 1
(config—if—ditap0)# local—cn 1
(config—if—ditap0)# remote—cn 1
(config—if—ditap0)# enable
```

или

```
(config)# interface ip6ditap 0
(config—if—ip6ditap0)# id 3
(config—if—ip6ditap0)# alg encrypt
(config—if—ip6ditap0)# local 2001:db8::10:9
(config—if—ip6ditap0)# remote 2001:db8::10:3
(config—if—ip6ditap0)# serial 1
(config—if—ip6ditap0)# local—cn 2
(config—if—ip6ditap0)# remote—cn 2
(config—if—ip6ditap0)# enable
```

59.5 PSK (предварительно распределённые ключи)

Предварительно распределённые ключи (Pre-shared Keys, PSK) представляют из себя конфиденциальную последовательность байт (от 8 до 255) и используются для взаимной аутентификации оппонентов IPsec. Для успешной аутентификации pre-shared ключи должны совпадать (по длине и содержимому) у обоих оппонентов. Ключи PSK сохраняются в системе с уникальными именами. Контейнеры сохранённых ключей PSK защищаются шифрованием с помощью ключа доступа (КД).

59.5.1 Ввод/импорт PSK

Pre-shared ключи можно загрузить с внешнего носителя или ввести вручную.

Рекомендуемый способ ввода pre-shared ключа в систему - это импорт ключа из контейнера DSRF. Контейнер DSRF содержит набор 32-байтных симметричных ключей, идентифицирующихся по номеру абонента.

Чтобы просмотреть информацию о DSRF-контейнере на внешнем носителе, необходимо ввести команду в enable-режиме:

```
# show crypto psk keys <носитель>
```

Если контейнер существует на внешнем носителе (в корневой директории), то будет отображена следующая информация:

- Номер зоны (Zone);
- Номер серии ключей (Serial);
- Номер абонента, для которого выпущен данный DSRF-контейнер (Abonent);

- Количество ключей в контейнере (Number of abonents).

Далее можно импортировать нужный ключ из контейнера с помощью команды:

```
# crypto psk set key <имя_ключа_в_системе> dsrf <носитель> <номер_абонента>
```

Номер абонента задаётся в десятичном виде и может быть от 1 до «Number of abonents» включительно.

Пример:

```
# show crypto psk keys flash0
DSRF key container on 'flash0' device:
Zone: 1
Serial: 1234
Abonent: 1
Number of abonents: 9999

# crypto psk set key dsrf_key dsrf flash0 9000
Info: Found possible DSRF container on 'flash0' device.
Info: Read 32 bytes of pre—shared key.
Info: Saving the key with internal name 'dsrf_key'.

# show crypto psk keys
dsrf_key
```

В данном примере в качестве pre-shared ключа загружается ключ номер 9000 из DSRF-контейнера с внешнего флеш-носителя и сохраняется в системе с внутренним именем ‘dsrf_key’. Список ключей, загруженных в систему, можно просмотреть с помощью команды ‘show crypto psk keys’ (без параметра <носитель>).

Также ключ можно загрузить из произвольного файла на внешнем носителе (не рекомендуется). Это можно сделать с помощью команды:

```
# crypto psk set key <имя_ключа_в_системе> file <носитель>:/<путь_к_файлу>
```

Всё содержимое файла воспринимаются как ключ. Файл должен иметь длину от 8 до 255 байт.

Пример:

```
# ls flash0:
total 12
drwxrwxr—x  2  adm  adm  4.0K  Jul 11 17:42  psks/
drwxrwxr—x  2  adm  adm  4.0K  Jul 11 17:42  keys/
drwxrwxr—x  2  adm  adm  4.0K  Jul 11 17:42  certs/

# ls flash0:/psks
total 8
—rwxrwxr—x  1  adm  adm   32  Jul 11 17:42  key1
—rwxrwxr—x  1  adm  adm  255  Jul 11 17:42  key2

# crypto psk set key psk1 file flash0:/psks/key1
```

```
Info: Read 32 bytes of pre—shared key.
Info: Saving the key with internal name ’psk1’.

# crypto psk set key psk2 file flash0:/psks/key2
Info: Read 255 bytes of pre—shared key.
Info: Saving the key with internal name ’psk2’.

# show crypto psk keys
dsrf_key
psk1
psk2
```

В данном примере в качестве pre-shared ключей загружаются файлы ‘key1’ и ‘key2’ с внешнего флеш-носителя и сохраняются в системе с внутренними именами ‘psk1’ и ‘psk2’ соответственно.

Если необходимо ввести ключ вручную, то рекомендуется это делать с помощью команды:

```
# crypto psk set key <имя_ключа> pass
```

При этом будет предложено ввести ключ в текстовом виде два раза (как пароль). Содержимое ключа при вводе отображаться не будет. В качестве ключа сохраняется введённый текст (в кодировке UTF-8) без заключительного символа перевода строки.

Также можно ввести ключ в открытом текстовом или 16-ричном виде, но эти варианты ввода не рекомендуются как небезопасные. После ввода ключей таким способом необходимо удалить журнал командной оболочки.

Пример:

```
# crypto psk set key psk3 text ”123”
# crypto psk set key psk4 hex 0x313233
# rm log:/dish.log
# show crypto psk keys
dsrf_key
psk1
psk2
psk3
psk4
```

59.5.2 Ассоциация ключей с туннелями IPsec

Чтобы ключ мог быть использован для взаимной аутентификации оппонентов IPsec, его необходимо ассоциировать с IP-адресами концов IPsec-туннеля. Это можно сделать с помощью следующей команды режима configure:

```
(config)# crypto psk map <локальный_IP> <удалённый_IP> <имя_ключа>
```

Примеры:

```
(config)# crypto psk map 10.1.0.1 10.2.0.1 psk1
(config)# crypto psk map 192.168.0.1 * psk2
(config)# crypto psk map * 10.3.0.1 psk3
```

Звёздочка означает любой IP. Не допускается использование сочетания «* *».

59.5.3 Удаление ключей и ассоциаций

Чтобы удалить pre-shared ключ, нужно выполнить следующую команду режима enable:

```
# crypto psk clear key <имя_ключа>
```

Удалить все pre-shared ключи можно следующей командой:

```
# crypto psk clear keys
```

Удалить ассоциацию PSK с IPsec можно командой режима configure:

```
(config)# no crypto psk map <локальный_IP> <удалённый_IP>
```

Чтобы удалить все ассоциации, нужно выполнить следующую команду в режиме configure:

```
(config)# no crypto psk maps
```

Для безопасного удаления ключей с внешних носителей рекомендуется использовать команду “clear removable” (см. раздел “Обслуживание”).

59.6 PKI (закрытые ключи, сертификаты, COC, PKCS#10)

59.6.1 Базовые понятия PKI

PKI - Public Key Infrastructure, Инфраструктура открытых ключей - технология аутентификации с помощью открытых ключей, связывающая открытые ключи с личностью пользователя посредством удостоверяющего центра (УЦ).

Закрытый ключ - конфиденциальный компонент пары асимметричных ключей, используемый для **создания** электронно-цифровой подписи (ЭЦП).

Открытый ключ - неконфиденциальный компонент пары асимметричных ключей, используемый для **проверки** электронно-цифровой подписи.

Имя X500 (DN, Distinguished Name) - последовательность типа «имя_параметра1=значение_параметра1, имя_параметра2=значение_параметра2, …», которая однозначно определяет конкретный субъект (человек, организация, маршрутизатор и т.д.). «Имя_параметра» определяется конкретным объектным идентификатором OID, который также определяет синтаксис «значения_параметра».

Удостоверяющий центр (УЦ) - организация, пользующаяся доверием и выпускающая X509-сертификаты.

Сертификат X509 - зафиксированная (неизменяемая) последовательность бинарных данных, которая содержит следующую информацию:

- Серийный номер сертификата;
- X500-имя удостоверяющего центра, выпустившего сертификат;
- Дата начала и конца действия сертификата;
- X500-имя субъекта, кому выдан сертификат;
- Открытый ключ субъекта;
- Дополнительная информация (область применения сертификата, точки распространения списков отзывов сертификата и т.д.);
- Электронно-цифровая подпись по вышеуказанным данным, сформированная закрытым ключом удостоверяющего центра, выпустившего данный сертификат.

Сертификат является неконфиденциальной информацией. Целостность сертификата можно проверить с помощью открытого ключа из сертификата удостоверяющего центра. Сертификат удостоверяющего центра может быть выпущен вышестоящим удостоверяющим центром. В результате для проверки сертификата необходима вся цепочка сертификатов УЦ до корневого.

Корневой (самоподписанный) сертификат - сертификат самого главного удостоверяющего центра, который пользуется абсолютным доверием. Корневой сертификат подписан закрытым ключом этого же УЦ и может быть проверен собственным открытым ключом. Также X500-имя издателя эквивалентно x500-имени субъекта. Корневой сертификат должен доставляться и устанавливаться в систему доверенным способом.

Если закрытый ключ субъекта скомпрометирован до окончания срока действия соответствующего сертификата, то удостоверяющему центру необходимо выпустить (и распространить) список отозванных сертификатов, содержащий серийный номер скомпрометированного сертификата.

Список отозванных сертификатов (COC, Certificate Revocation List, CRL) - зафиксированная бинарная последовательность, содержащая следующую информацию:

- X500-имя удостоверяющего центра, выпустившего данный список;
- Дата выпуска;
- Предельная дата выпуска следующего COC;
- Список отозванных сертификатов (серийный номер, время отзыва, причина отзыва и т.д.);
- Дополнительная информация;
- Электронно-цифровая подпись по выше указанным данным, сформированная закрытым ключом удостоверяющего центра, выпустившего данный COC.

OCSP - (Online Certificate Status Protocol) - протокол немедленного выяснения действительности сертификата. Данный протокол позволяет отзывать сертификаты более оперативно, чем COC. Протокол работает по механизму «запрос-ответ» от клиента к УЦ (или уполномоченному OCSP-серверу). Запрос содержит идентификатор сертификата, статус которого требуется выяснить. Ответ содержит информацию о статусе сертификата (действительный/отозванный/неизвестный). Запрос может быть подписан ключом клиента. Ответ всегда подписан ключом УЦ или уполномоченным OCSP-сервером.

Запрос на выпуск сертификата (PKCS10-запрос) - бинарная последовательность, содержащая следующую информацию:

- X500-имя субъекта;
- Открытый ключ субъекта;
- Атрибут (необязательный), содержащий расширения X.509, которые должны быть включены в сертификат;
- Другие (необязательные) атрибуты;
- Электронная подпись по вышеперечисленным данным, выполненная на ключе субъекта.

PKCS10-запрос генерируется вместе с закрытым ключом. Закрытый ключ сохраняется у субъекта, PKCS10-запрос доставляется доверенным способом в удостоверяющий центр и используется для выпуска сертификата.

Для более подробной информации см. “Руководство администратора DiCert”.

59.6.2 Полная очистка PKI

Если необходимо удалить все закрытые ключи, сертификаты, COC и PKCS10-запросы из системы, следует выполнить команду привилегированного режима:

```
# crypto pki clear all
```

59.6.3 Управление закрытыми ключами

Закрытые ключи устанавливаются в систему с внешних носителей.

(Также см. раздел “Генерация закрытых ключей” ниже).

ВАЖНО: Установка закрытого ключа является доверенной процедурой, и должны быть обеспечены все необходимые административные меры безопасности при доставке и установке ключа, с целью избежания его компрометации или подмены. После установки в систему закрытый ключ защищается ключом доступа.

Поддерживаются закрытые ключи для алгоритмов: ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012 (256 и 512 бит).

Поддержива́емый формат контейнера закрытых ключей: PKCS#15.

59.6.3.1 Импорт ключей

Для импорта закрытого ключа с внешнего носителя необходимо выполнить команду привилегированного режима:

```
# crypto pki import key from <носитель>:/путь_к_ключу
```

Найденный ключ сохранится в системе с тем же именем файла, который имеет контейнер на внешнем носителе. Если необходимо дать импортируемому ключу другое системное имя, то следует выполнить команду с параметром «to»:

```
# crypto pki import key from <носитель>:/путь_к_ключу to <новое_имя>
```

Для вывода списка ключей на внешнем носителе следует использовать команду:

```
# show crypto pki keys <носитель>:[/<путь_к_директории>]
```

Пример:

Вставляем внешний флэш-носитель и выполняем команду просмотра:

```
# show crypto pki keys flash0
certs/
crls/
keys/
```

Видно, что на внешнем носителе в корневой директории контейнеров с ключами не содержится. Поэтому следует вывести содержимое поддиректории keys.

```
# show crypto pki keys flash0:/keys
key1.p15    pkcs15
key2.p15    pkcs15   CN=Иванов Иван Иванович, O=Хорошая организация, C=RU
```

В контейнерах PKCS#15 с закрытым ключом может быть ассоциировано X500-имя субъекта, которому этот ключ принадлежит. В данном примере ключ key2.p15 принадлежит Иванову Ивану Ивановичу.

Подробную информацию о ключе в контейнере PKCS#15 можно вывести с помощью команды:

```
# show crypto pki key flash0:/keys/key2.p15
```

Данная команда выводит следующую информацию о ключе:

- Текстовая метка ключа (если есть).
- X500-имя владельца ключа (если есть).
- Алгоритм ключа и криптопараметры.
- Допустимое использование ключа (флаги KeyUsage).
- Отпечаток ключа (который можно сравнить с отпечатком открытого ключа сертификата).

Принимается решение импортировать оба ключа (с внутренними именами unknown_key1 и ivan_key соответственно):

```
# crypto pki import key from flash0:/keys/key1.p15 to unknown_key1
# crypto pki import key from flash0:/keys/key2.p15 to ivan_key
```

Если контейнер PKCS#15 защищён паролем, то потребуется его ввести.

Закрытые ключи также можно импортировать с USB-токена/смарт-карты.

Для просмотра списка закрытых ключей на токене необходимо выполнить команду:

```
# show crypto pki keys token:
```

Закрытые ключи защищены пин-кодом пользователя токена, поэтому для выполнения данной операции необходимо ввести пин-код.

ВНИМАНИЕ: При многократном вводе ошибочного пин-кода смарт-карта может заблокироваться. Количество попыток ввода зависит от производителя и настройки смарт-карты.

Список закрытых ключей выводится в следующем формате.

```
key1.p15    pkcs15
key2.p15    pkcs15   CN=Иванов Иван Иванович,O=Хорошая организация,C=RU
```

Примечание: Все "файловые" объекты на токене хранятся в "корневой" директории. Понятие "поддиректории" отсутствует.

Просмотреть подробную информацию о ключе можно с помощью команды:

```
# show crypto pki key token:/key1.p15
```

Импорт закрытого ключа из токена выполняется командой:

```
# crypto pki import key from token:/[<имя_файла_ключа>] [to <новое_имя>]
```

Если на токене находится один закрытый ключ, то импорт можно выполнить командой:

```
# crypto pki import key from token:
```

59.6.3.2 Просмотр импортированных ключей

Чтобы вывести список ключей, импортированных в систему, нужно выполнить следующую команду в привилегированном режиме:

```
# show crypto pki keys
```

В примере, описанном выше, данная команда выведет следующую информацию:

```
ivan_key
old_key1
unknown_key1
```

59.6.3.3 Удаление ключей из системы

Чтобы удалить закрытый ключ из системы, нужно выполнить следующую команду в привилегированном режиме:

```
# crypto pki clear key <имя_ключа>
```

Чтобы удалить все закрытые ключи, нужно выполнить следующую команду:

```
# crypto pki clear keys
```

Для безопасного удаления ключей с внешних носителей рекомендуется использовать команду "clear removable" (см. раздел "Обслуживание").

59.6.4 Управление сертификатами

В Dionis-NX различаются следующие виды сертификатов:

- Корневые сертификаты (главных УЦ);
- Сертификаты подчинённых УЦ;
- Клиентские сертификаты;
- Сертификаты для подписи/проверки OCSP запросов/ответов.

Сертификаты импортируются в систему вручную с внешних носителей, за исключением чужих клиентских сертификатов, получаемых автоматически службой IPsec IKE (см. Туннели IPsec ниже). Автоматически полученные сертификаты не сохраняются в локальном хранилище.

Поддерживаемые алгоритмы ЭЦП:

- ГОСТ Р 34.10-2001 с ГОСТ Р 34.11-94;
- ГОСТ Р 34.10-2012 (256 бит) с ГОСТ Р 34.11-2012 (256 бит);
- ГОСТ Р 34.10-2012 (512 бит) с ГОСТ Р 34.11-2012 (512 бит).

Поддерживаемые форматы для импорта: DER, PEM, P7B.

Каждому типу сертификата соответствует отдельное локальное хранилище. Сертификаты идентифицируются по локальным именам, которые им присваиваются при импорте. Имена сертификатов разных типов могут пересекаться. (Однако рекомендуется избегать пересечения имён между сертификатами корневых и подчинённых УЦ, если необходимо настраивать дополнительные точки распространения СОС-командами «crypto ike cainfo» - см. Туннели IPsec ниже).

ВАЖНО: Импорт корневого сертификата является доверенной процедурой, и поэтому должны быть соблюдены все необходимые административные меры по безопасной доставке носителя с корневым сертификатом. Импортированный корневой сертификат защищается от подмены имитовставкой, рассчитанной с помощью ключа доступа.

Для клиентских сертификатов, относящихся к данному узлу, должны быть импортированы соответствующие закрытые ключи (см. выше). Совпадение внутренних системных имён ключей и соответствующих сертификатов необязательно.

Для проверки ЭЦП OCSP-ответов используются сертификаты удостоверяющих центров. Если OCSP-ответы формируются уполномоченным OCSP-сервером, то для проверки OCSP-ответа будет использоваться сертификат OCSP-сервера. Данный сертификат должен содержать OID 1.3.6.1.5.5.7.3.9 («ocspSigning») в поле «Extended Key Usage». Данный сертификат обычно передаётся в теле OCSP-ответа. Если он не передаётся, то его следует импортировать в хранилище OCSP-сертификатов.

Если необходимо подписывать OCSP-запросы, то следует импортировать необходимые сертификаты в хранилище OCSP-сертификатов, а также соответствующие им закрытые ключи. Если OCSP-запрос должен подписываться клиентским сертификатом, то последний можно скопировать в хранилище OCSP-сертификатов командой «crypto pki copy cert» (см. ниже). Подпись OCSP-запроса формируется следующим образом:

1. Допустим, требуется проверка на отзыв некоторого сертификата A;

2. Ищется сертификат удостоверяющего центра, выпустившего сертификат A - сертификат B;
3. Если сертификат B не найден, то сертификат A считается заведомо недействительным;
4. В хранилище OCSP-сертификатов ищется сертификат, подписанный сертификатом B (выпущенный тем же УЦ) - сертификат C;
5. Если сертификат C не найден, OCSP-запрос не будет подписан;
6. Ищется закрытый ключ, соответствующий сертификату C;
7. Если ключ не найден, OCSP-запрос не будет подписан;
8. OCSP-запрос подписывается найденным ключом.

59.6.4.1 Импорт сертификатов

Команды просмотра и импорта сертификатов имеют синтаксис, похожий на команды просмотра и импорта ключей.

Чтобы просмотреть сертификаты на внешнем носителе (или в хранилище “file:”), нужно выполнить команду (в привилегированном режиме):

```
# show crypto pki certs <носитель>:[/<путь_к_директории>]
```

Пример вывода команды:

```
subdir1/
subdir2/
file1    root   CN=Главный удостоверяющий центр, O=Хорошая организация, C=RU
file2    ca     CN=УЦ первого отдела, O=Хорошая организация, C=RU
file3    user   CN=Иван Иванович Иванов, O=Хорошая организация, C=RU
```

Список сертификатов выводится в формате: имя файла, тип, X500-имя субъекта.

Чтобы просмотреть подробную информацию о сертификате на внешнем носителе (или в “file:”), можно выполнить команду:

```
# show crypto pki cert <носитель>:/<путь_к_сертификату>
```

Если в качестве внешнего носителя используется USB-токен/смарт-карта, то список сертификатов и информация о конкретном сертификате выводятся с помощью соответствующих команд:

```
# show crypto pki certs token:
# show crypto pki cert token:/<имя_файла>
```

Примечание: При обращении к токену/смарт-карте будет предложено опционально ввести пин-код пользователя. Ввод пин-кода необходим для просмотра списка корневых сертификатов, защищённых от подмены. Для вывода списка сертификатов пользователей, промежуточных УЦ, незащищённых корневых сертификатов пин-код не требуется.

Следующие команды выполняют импорт соответственно корневых, промежуточных, клиентских и OCSP-сертификатов:

```
# crypto pki import root ca cert from <внешний_носитель>:/<путь_к_файлу> [to <новое_имя>]
# crypto pki import ca cert from <внешний_носитель>:/<путь_к_файлу> [to <новое_имя>]
# crypto pki import cert from <внешний_носитель>:/<путь_к_файлу> [to <новое_имя>]
# crypto pki import ocsp cert from <внешний_носитель>:/<путь_к_файлу> [to <новое_имя>]
```

Примечание: Функция автодополнения (вызываемая клавишей Tab) не работает для защищённых корневых сертификатов, находящихся на токене.

Чтобы скопировать клиентский сертификат в хранилище OCSP-сертификатов, следует использовать команду:

```
# crypto pki copy cert <имя_клиентского_сертификата> to ocsp cert [<новое_имя>]
```

Пример:

Вставляем внешний флэш-носитель и выполняем команду просмотра контейнеров сертификатов:

```
# show crypto pki certs flash0
certs/
crls/
keys/
```

Просмотрим директорию «certs»:

```
# show crypto pki certs flash0:/certs
cacer.p7b/
ca.cer       root   CN=УЦ, O=Правильная организация, C=RU
ocsp.cer     user   CN=OCSP сервер, O=Правильная организация, C=RU
```

В данной директории мы видим корневой сертификат “дружественной нам” организации - ca.cer.

Просмотреть подробную информацию о сертификате можно с помощью команды:

```
# show crypto pki cert flash0:/certs/ca.cer
```

Мы принимаем решение его установить в систему, так как доверяем данной организации и сертификатам, выпущенным её УЦ:

```
# crypto pki import root ca cert from flash0:/certs/ca.cer to right_org_ca_cert
```

Также мы видим сертификат доверенного OCSP-сервера, который будет использоваться для проверки OCSP-ответов от этого сервера. Мы его тоже устанавливаем:

```
# crypto pki import ocsp cert from flash0:/certs/ocsp.cer to right_org_ocsp_cert
```

Также мы видим некую “директорию” “cacer.p7b”. На самом деле в директории “certs” находится файл “cacer.p7b” формата P7B. Контейнеры P7B отображаются как “директории”, так как могут содержать несколько сертификатов и CRL.

Просмотрим контейнер “cacer.p7b”

```
# show crypto pki certs flash0:/certs/cacer.p7b
cert01       user   CN=Петров Пётр Петрович, O=Хорошая организация, C=RU
cert02       root   CN=Главный УЦ, O=Хорошая организация, C=RU
cert03       ca     CN=УЦ первого филиала, O=Хорошая организация, C=RU
cert04       user   CN=Иванов Иван Иванович, O=Хорошая организация, C=RU
```

Мы видим сертификаты удостоверяющих центров нашей организации, а также сертификат Иванова Ивана Ивановича, чей ключ мы уже импортировали.

Просмотреть подробную информацию о сертификатах можно с помощью команд:

```
# show crypto pki cert flash0:/certs/cacer.p7b/cert01
# show crypto pki cert flash0:/certs/cacer.p7b/cert02
...
```

Мы импортируем сертификаты наших удостоверяющих центров и сертификат Иванова Ивана Ивановича:

```
# crypto pki import root ca cert from flash0:/certs/cacer.p7b/cert02 to our_main_ca_cert
# crypto pki import ca cert from flash0:/certs/cacer.p7b/cert03 to our_local_ca_cert
# crypto pki import cert from flash0:/certs/cacer.p7b/cert04 to ivan_cert
```

Мы знаем, что наш локальный удостоверяющий центр также является OCSP-сервером (ответы от него будут проверяться сертификатом «our_local_ca_cert»), но также мы знаем, что наш OCSP-сервер разрешает только подписанные запросы к нему. Мы копируем сертификат Иванова Ивана Ивановича в хранилище сертификатов для OCSP, чтобы запрос OCSP подписывался его закрытым ключом (импортированном в предыдущем примере):

```
# crypto pki copy cert ivan_cert to ocsp cert
```

59.6.4.2 Просмотр импортированных сертификатов

Следующие команды выводят списки сертификатов, находящихся в локальных хранилищах (соответственно, корневые сертификаты, сертификаты подчинённых УЦ, клиентские сертификаты, сертификаты для OCSP):

```
# show crypto pki root ca certs
# show crypto pki ca certs
# show crypto pki certs
# show crypto pki ocsp certs
```

Вывод осуществляется в формате:

```
<внутреннее_имя>   <X500—имя субъекта>
...
```

Пример:

Просмотрим наши хранилища сертификатов после выполнения команд предыдущего примера:

```
# show crypto pki root ca certs
right_org_ca_cert    CN=УЦ, O=Правильная организация, C=RU
our_main_ca_cert     CN=Главный УЦ, O=Хорошая организация, C=RU

# show crypto pki ca certs
our_local_ca_cert    CN=УЦ первого филиала, O=Хорошая организация, C=RU

# show crypto pki certs
ivan_cert            CN=Иванов Иван Иванович, O=Хорошая организация, C=RU

# show crypto pki ocsp certs
ivan_cert            CN=Иванов Иван Иванович, O=Хорошая организация, C=RU
right_org_ocsp_cert  CN=OCSP сервер, O=Правильная организация, C=RU
```

Также можно посмотреть подробную информацию о конкретном сертификате с помощью команд:

```
# show crypto pki root ca cert <имя>
# show crypto pki ca cert <имя>
# show crypto pki cert <имя>
# show crypto pki ocsp cert <имя>
```

59.6.4.3 Удаление сертификатов из системы

Удалить сертификат из системы можно с помощью одной из команд (для соответствующего хранилища):

```
# crypto pki clear root ca cert <имя>
# crypto pki clear ca cert <имя>
# crypto pki clear cert <имя>
# crypto pki clear ocsp cert <имя>
```

Чтобы удалить все сертификаты из соответствующего хранилища, следует выполнить одну из команд:

```
# crypto pki clear root ca certs
# crypto pki clear ca certs
# crypto pki clear certs
# crypto pki clear ocsp certs
```

59.6.5 Управление списками отозванных сертификатов

Обычно списки отозванных сертификатов могут быть получены динамически по сети (например, службой IPsec IKE - см. ниже). Они хранятся в оперативной памяти и динамически обновляются. Но бывают ситуации, когда надо их установить вручную с внешнего носителя. Также службы могут кэшировать СОС в данном хранилище, чтобы они были доступны сразу после перезагрузки системы (см. опции «crl cache» и «crl policy strict» службы IPsec IKE).

Поддерживаются форматы: DER, PEM, P7B.

59.6.5.1 Импорт СОС

Импорт списков отозванных сертификатов похож на импорт закрытых ключей.

Для просмотра СОС на внешних носителях (или в хранилище “file:”) и импорта следует использовать команды привилегированного режима:

```
# show crypto pki crls <носитель>:[/<путь_к_директории>]
# show crypto pki crl <носитель>:/<путь_к_файлу>
# crypto pki import crl from <носитель>:/<путь_к_файлу> [to <новое_имя>]
```

Для просмотра и импорта СОС с USB-токена/смарт-карты используются команды:

```
# show crypto pki crls token:
# show crypto pki crl token:/<имя_файла>
# crypto pki import crl from token:/<имя_файла> [to <новое_имя>]
```

Контейнеры P7B помимо сертификатов могут содержать списки отзыва сертификатов. Если контейнер содержит хотя бы один СОС, то следующая команда отобразит его как «директорию»:

```
# show crypto pki crls flash0
cacer.p7b/
```

Пытаемся импортировать СОС из P7B:

```
# crypto pki import crl from flash0:/cacer.p7b
Error: Multiple CRLs found on flash device. Use ’from’ option to specify a CRL.
```

Мы видим, что наш контейнер содержит более одного СОС. в таком случае следует просмотреть содержимое контейнера:

```
# show crypto pki crls flash0:/cacer.p7b
crl01     CN=Главный УЦ, O=Хорошая организация, C=RU
crl02     CN=УЦ первого филиала, O=Хорошая организация, C=RU
```

Мы видим, что контейнер содержит списки отзывов, выпущенные главным УЦ и УЦ первого филиала.

Просмотреть подробную информацию о списках можно с помощью команд:

```
# show crypto pki crl flash0:/cacer.p7b/crl01
# show crypto pki crl flash0:/cacer.p7b/crl02
```

Импортируем оба списка:

```
# crypto pki import crl from flash0:/cacer.p7b/crl01 to our_main_ca_crl
# crypto pki import crl from flash0:/cacer.p7b/crl02 to our_local_ca_crl
```

59.6.5.2 Просмотр СОС

Чтобы вывести список СОС, находящихся в локальном хранилище, следует использовать команду:

```
# show crypto pki crls
```

В выводе команды будут также показаны имена издателей соответствующих СОС. Пример:

```
our_main_ca_crl     CN=Главный УЦ, O=Хорошая организация, C=RU
our_local_ca_crl    CN=УЦ первого филиала, O=Хорошая организация, C=RU
```

Получить детальную информацию о СОС можно с помощью команды:

```
# show crypto pki crl <имя>
```

59.6.5.3 Удаление СОС из системы

Удалить конкретный СОС из системы можно с помощью команды:

```
# crypto pki clear crl <имя>
```

Удалить все СОС можно с помощью команды:

```
# crypto pki clear crls
```

59.6.6 Проверка сертификатов на действительность

К использованию в системах IPsec и DiCert допускаются только действительные сертификаты. Если сертификат недействителен, его использование запрещено.

Сертификат является действительным при соблюдении всех следующих требований:

- Текущее время попадает в срок действия сертификата (сертификат не просрочен).
- Сертификат подписан действительным сертификатом удостоверяющего центра.
- Успешно выстроена цепочка действительных сертификатов вплоть до корневого.
- Сертификат не отозван с помощью списка отозванных сертификатов (CRL).
- Используемый CRL действителен (не просрочен, подписан действительным сертификатов УЦ).

В Dionis-NX существует возможность вручную проверить сертификаты, загруженные в систему. Для проверки сертификатов используются сертификаты корневых и промежуточных УЦ в соответствующих хранилищах, а также списки отозванных сертификатов в хранилище CRL.

Проверить сертификаты разных типов можно с помощью команд:

```
# crypto pki check cert <имя> [no—strict—crl]
# crypto pki check ocsp cert <имя> [no—strict—crl]
# crypto pki check ca cert <имя> [no—strict—crl]
# crypto pki check root ca cert <имя> [no—strict—crl]
```

В командах задаётся имя сертификата в соответствующем хранилище.

Команды выводят подробную информацию о проверке цепочки сертификатов. В случае недействительности сертификата будет выведена подробная информация о причине недействительности.

По умолчанию (в случае отсутствия параметра ‘no-strict-crl’) будет осуществляться строгая проверка CRL. То есть для каждого издателя в цепочке доверия должен быть найден действительный (не просроченный) CRL. В противном случае проверяемый сертификат будет признан недействительным.

Параметр ‘no-strict-crl’ ослабляет политику проверки CRL и допускает отсутствие CRL или использование просроченного CRL.

59.6.7 Шаблоны PKCS10-запросов

Шаблон PKCS10-запроса представляет собой набор настроек, которые используются при генерации закрытого ключа и PKCS10-запроса.

Для создания/редактирования шаблона необходимо войти в режим редактирования шаблона с помощью команды (режима “configure”):

```
(config)# crypto pki pkcs10 template <имя_шаблона>
(cfg—p10t—tname)# _
```

Для удаления шаблона можно использовать команду:

```
(config)# no crypto pki pkcs10 template <имя_шаблона>
```

Далее описываются настройки шаблона PKCS10-запроса.

59.6.7.1 Субъект

Опция “subject” (в режиме “cfg-p10t”) задаёт DN-имя субъекта будущего PKCS10-запроса. Опция имеет следующие форматы:

Формат (1):

```
(cfg—p10t—tname)# subject dn ”<dn—имя>”
```

Формат (2):

```
(cfg—p10t—tname)# subject from cert <имя_контейнера_пользовательского_сертификата>
```

При использовании формата (1) задаётся X500-имя субъекта. Поддерживаются следующие X500-атрибуты:

Мнемоника	Название	OID	Тип данных	Макс. симв.	Использование
CN	commonName	2.5.4.3	UTF8String	64	ФИО человека, название объекта, и т.д.
S	surname	2.5.4.4	UTF8String	32768	Фамилия человека
G	givenName	2.5.4.42	UTF8String	32768	Имя и отчество человека
INN	INN	1.2.643.3.131.1.1	NumericString	12	ИНН физ./юр. лица (12 цифр)
OGRN	OGRN	1.2.643.100.1	NumericString	13	ОГРН юр. лица
SNILS	SNILS	1.2.643.100.3	NumericString	11	СНИЛС физ. лица
E	emailAddress	1.2.840.113549.1.9.1	IA5String	255	Электронная почта

Мнемоника	Название	OID	Тип данных	Макс. симв.	Использование
pseudonym	pseudonym	2.5.4.65	UTF8String	128	Произвольное название объекта
SN	serialNumber	2.5.4.5	PrintableString	64	Серийный номер объекта
T	title	2.5.4.12	UTF8String	64	Должность
OU	organizationUnitName	2.5.4.11	UTF8String	64	Подразделение организации
O	organizationName	2.5.4.10	UTF8String	64	Название организации
street	streetAddress	2.5.4.9	UTF8String	128	Улица, дом, кв. и т.д.
L	localityName	2.5.4.7	UTF8String	128	Населённый пункт
ST	stateOrProvinceName	2.5.4.8	UTF8String	128	Регион, область
C	countryName	2.5.4.6	PrintableString	2	Код страны (RU)
DC	domainComponent	0.9.2342.19200300.100.1.25	IA5String	63	Часть доменного имени

Подробнее см. “Руководство администратора DiCert”.

Если в хранилище пользовательских сертификатов присутствует сертификат требуемого субъекта, то во избежание ошибок при наборе X500-имени субъекта можно его наследовать из сертификата, используя формат (2). Следует помнить, что данный сертификат должен находиться в хранилище до тех пор, пока будет использоваться данный PKCS10-шаблон. (Имя субъекта извлекается из сертификата в момент выполнения команды ‘crypto pki gen’).

59.6.7.2 Контроль X500-имени субъекта

По умолчанию при генерации PKCS10-запросов на X500-имя субъекта накладываются следующие ограничения:

- Длина значения X500-атрибута не должна превышать предельного значения, определённого типом атрибута (см. таблицу выше).
- Атрибуты одного типа не должны повторяться в X500-имени.

При нарушении данных правил система не позволит выпустить PKCS10-запрос.

Для отключения данных ограничений используются соответствующие опции:

```
(cfg—p10t—tname)# limit x500—attr—len off
(cfg—p10t—tname)# x500—attr unique off
```

Для включения проверки данных ограничений используются следующие команды:

```
(cfg—p10t—tname)# limit x500—attr—len on
(cfg—p10t—tname)# x500—attr unique on
```

59.6.7.3 Алгоритм ключа

Следующая опция (в режиме "cfg-p10t") задаёт криптографический алгоритм и параметры, используемые при генерации ключа и PKCS10-запроса:

```
(cfg—p10t—tname)# algorithm <alg>
```

Возможные алгоритмы/параметры:

Мнемоника	Алгоритм и параметры
gost2001-a	ГОСТ Р 34.10-2001, id-GostR3410-2001-CryptoPro-A-ParamSet
gost2001-b	ГОСТ Р 34.10-2001, id-GostR3410-2001-CryptoPro-B-ParamSet
gost2001-c	ГОСТ Р 34.10-2001, id-GostR3410-2001-CryptoPro-C-ParamSet
gost2001-xcha	ГОСТ Р 34.10-2001, id-GostR3410-2001-CryptoPro-XchA-ParamSet
gost2001-xchb	ГОСТ Р 34.10-2001, id-GostR3410-2001-CryptoPro-XchB-ParamSet
gost2012-256-a	ГОСТ Р 34.10-2012 (256 бит), id-GostR3410-2001-CryptoPro-A-ParamSet
gost2012-256-b	ГОСТ Р 34.10-2012 (256 бит), id-GostR3410-2001-CryptoPro-B-ParamSet
gost2012-256-c	ГОСТ Р 34.10-2012 (256 бит), id-GostR3410-2001-CryptoPro-C-ParamSet
gost2012-256-xcha	ГОСТ Р 34.10-2012 (256 бит), id-GostR3410-2001-CryptoPro-XchA-ParamSet
gost2012-256-xchb	ГОСТ Р 34.10-2012 (256 бит), id-GostR3410-2001-CryptoPro-XchB-ParamSet
gost2012-512-a	ГОСТ Р 34.10-2012 (512 бит), id-tc26-gost-3410-12-512-paramSetA
gost2012-512-b	ГОСТ Р 34.10-2012 (512 бит), id-tc26-gost-3410-12-512-paramSetB

59.6.7.4 Содержимое расширения KeyUsage

Расширение сертификата/PKCS10-запроса KeyUsage содержит флаги, определяющие возможные применения ключа.

Бит	Значение
digitalSignature	Электронная подпись данных. (Не включает в себя keyCertSign и cRLSign)
nonRepudiation/contentCommitment	Юридически не отрекаемая ЭП
keyEncipherment	Возможность использования для KeyTransport-структур
dataEncipherment	Возможность использования для шифрования данных.
keyAgreement	Возможность использования для механизма KeyAgreement
keyCertSign	Возможность использования для подписи сертификатов (только для УЦ)
cRLSign	Возможность использования для выпуска списков отозванных сертификатов (только для УЦ)

Бит	Значение
encipherOnly	Только для зашифрования данных при keyAgreement
decipherOnly	Только для расшифрования данных при keyAgreement

Следующие опции (в режиме “cfg-p10t”) определяют наличие/отсутствие соответствующих флагов в расширении KeyUsage PKCS10-запроса.

```
(cfg—p10t—tname)# ku digital—signature on|off
(cfg—p10t—tname)# ku non—repudiation on|off
(cfg—p10t—tname)# ku data—encipherment on|off
(cfg—p10t—tname)# ku enc—flags [encipherment] [agreement [encipher—only|decipher—only]]
```

По умолчанию действуют настройки:

```
ku digital—signature on
ku non—repudiation off
ku enc—flags
```

59.6.7.5 Содержимое расширения ExtendedKeyUsage

Расширение сертификата/PKCS10-запроса ExtendedKeyUsage содержит OID-ы, более точно описывающие возможные применения ключа. Содержимое данного расширения не должно противоречить содержимому расширения KeyUsage.

Следующая команда (в режиме “cfg-p10t”) переводит интерфейс в режим редактирования набора OID-ов расширения ExtendedKeyUsage (режим “cfg-p10t-eku”):

```
(cfg—p10t—tname)# eku oids
(cfg—p10t—tname—eku)# _
```

Для добавления OID-ов в список используется команда (в режиме “cfg-p10t-eku”):

```
(cfg—p10t—tname—eku)# oid <oid_name>|<oid>
```

Параметры:

<oid> – текстовое представление объектного идентификатора вида “1.2.3.4.5”.

<oid_name> – мнемоническое имя OID-а (для часто используемых OID-ов).

Поддерживаются следующие мнемонические имена:

Имя	OID	Применение
tls-server	1.3.6.1.5.5.7.3.1	TLS-сервер
tls-client	1.3.6.1.5.5.7.3.2	TLS-клиент
code-signing	1.3.6.1.5.5.7.3.3	Подпись исполняемого кода
email-protection	1.3.6.1.5.5.7.3.4	Защита электронной почты
time-stamping	1.3.6.1.5.5.7.3.8	Установка штампа времени
ocsp-signing	1.3.6.1.5.5.7.3.9	Доверенный OCSP-сервер

Имя	OID	Применение
ike-intermediate	1.3.6.1.5.5.8.2.2	Узел IPsec
dicert-reply-signing	1.3.6.1.4.1.13312.503.4.1.3	Сертификат подписи ответов сервера DiCert
any-extended-key-usage	2.5.29.37.0	Любое другое использование ключа

Для удаления OID-а из списка используется команда (в режиме “cfg-p10t-eku”):

```
(cfg—p10t—tname—eku)# no oid <oid_name>|<oid>
```

Для удаления всех OID-ов можно использовать команду (в режиме “cfg-p10t”):

```
(cfg—p10t—tname)# no eku oids
```

59.6.7.6 Содержимое расширения SubjectAltName

Расширение сертификата/PKCS10-запроса SubjectAltName содержит дополнительную информацию о субъекте.

Опции “subjaltname” (в режиме “cfg-p10t”) определяют содержимое расширения SubjectAltName в PKCS10-запросе. Опции имеют следующий формат:

```
(cfg—p10t—tname)# subjaltname email <e—mail_субъекта>
(cfg—p10t—tname)# subjaltname ip <ip—адрес_субъекта>
(cfg—p10t—tname)# subjaltname fqdn <доменное_имя_субъекта>
(cfg—p10t—tname)# subjaltname uri <uri_субъекта>
```

Для удаления соответствующих опций можно использовать следующие команды:

```
(cfg—p10t—tname)# no subjaltname email
(cfg—p10t—tname)# no subjaltname ip
(cfg—p10t—tname)# no subjaltname fqdn
(cfg—p10t—tname)# no subjaltname uri
```

59.6.7.7 OID атрибута расширений X.509 в PKCS10-запросе

Атрибут PKCS10-запроса, содержащий расширения X.509, может идентифицироваться одним из следующих OID-ов:

Параметр	OID
rsa	1.2.840.113549.1.9.14
microsoft	1.3.6.1.4.1.311.2.1.14

Следующая опция определяет, какой OID будет использоваться при генерации PKCS10-запроса:

```
(cfg—p10t—tname)# extensions oid rsa|microsoft
```

По умолчанию используется OID RSA.

Данная опция предназначена исключительно для совместимости с другими PKI. УЦ DiCert распознаёт оба OID-а при чтении PKCS10-запросов.

59.6.8 Генерация ключа и PKCS10-запроса

Генерация закрытого ключа и PKCS10-запроса на Dionis-NX осуществляется командой “crypto pki gen key” (в режиме “enable”) на основании шаблона PKCS10-запроса.

Команда имеет следующие форматы:

Формат (1):

```
# crypto pki gen key int <key_name> pkcs10 <p10_name> from <p10_template>
```

Формат (2):

```
# crypto pki gen key ext <key_path> pkcs10 <p10_name> from <p10_template>
```

Формат (1) используется для сохранения закрытого ключа во внутреннем хранилище ключей Dionis-NX.

Формат (2) используется для сохранения закрытого ключа на внешнем носителе. Закрытые ключи сохраняются в формате PKCS15.

Если в качестве внешнего носителя используется флэш, то контейнер ключа можно защитить паролем. В терминале будет предложено ввести пароль. Если пароль не нужен, следует нажать “Enter”.

Если в качестве внешнего носителя используется USB-токен/смарт-карта, контейнер ключа будет защищён пин-кодом пользователя токена/смарт-карты. Имя файла ключа должно оканчиваться на “.p15”.

Параметры:

- <key_name> – внутреннее имя, с которым сгенерированный закрытый ключ будет сохранён во внутреннем хранилище закрытых ключей Dionis-NX;
- <key_path> – полный путь к файлу контейнера закрытого ключа на внешнем носителе; (пример: “flash0:/some_dir/mykey.p15”, “token:/mykey.p15”);
- <p10_name> – имя контейнера сгенерированного PKCS10-запроса, сохраняемого внутри системы; (см. раздел “Управление PKCS10-запросами” ниже);
- <p10_template> – имя шаблона PKCS10-запроса.

Сгенерированный PKCS10-запрос сохраняется внутри системы в хранилище PKCS10-запросов с указанным именем. Далее он может быть экспортирован на внешний носитель для доверенной доставки в УЦ, либо может быть передан на сервер DiCert по протоколу DCRP, если на данном узле есть старые закрытый ключ и сертификат для данного субъекта (см. раздел “Выпуск сертификата по DCRP” ниже).

59.6.9 Генерация nam-файла

Для старых версий некоторых продуктов Фактор-ТС, использующих закрытые ключи в контейнерах PKCS15, необходимо наличие на внешнем носителе дополнительного файла с расширением ".nam" в одной директории с файлом ".p15". Только в этом случае PKCS15-контейнер становится "видимым" для данного программного обеспечения.

Создать nam-файл на внешнем носителе можно с помощью команды:

```
# crypto pki create nam <имя_контейнера> <путь_к_nam_файлу>
```

<имя_контейнера> - это строка, состоящая строго из 15 ASCII-символов, идентифицирующая контейнер. Данная строка записывается в nam-файл в кодировке ASN.1 DER.

Пример:

Допустим, был выпущен закрытый ключ и записан на внешний носитель с именем "abcd1234.p15". Для создания соответствующего nam-файла необходимо выполнить команду:

```
# crypto pki create nam "A:\\abcd1234.p15" flash0:/some_dir/abcd1234.nam
```

Для создания nam-файла на токене/смарткарте необходимо выполнить команду:

```
# crypto pki create nam "A:\\abcd1234.p15" token:/abcd1234.nam
```

Примечание: Следует обратить внимание на обязательность экранирования символа обратной косой черты дополнительной обратной косой чертой.

59.6.10 Экспорт PKCS10-запроса на внешний носитель

Экспорт PKCS10-запроса из внутреннего хранилища на внешний носитель осуществляется командой (в режиме "enable"):

```
# crypto pki export pkcs10 <p10_name> to <p10_path> [non—private]
```

Параметры:

- <p10_name> – имя контейнера сгенерированного PKCS10-запроса, сохранённого внутри системы; (см. раздел "Управление PKCS10-запросами" ниже);
- <p10_path> – полный путь к файлу контейнера PKCS10-запроса на внешнем носителе; (пример: "flash0:/some_dir/myreq.p10", "token:/myreq.p10");

Экспорт поддерживается в следующие устройства: flash:, token:, file:, share:.

ВНИМАНИЕ: Доставка PKCS10-запроса в удостоверяющий центр должна осуществляться доверенным способом.

Если в качестве внешнего носителя используется "token:" (USB-токен/смарт-карта), запрос экспортируется на токен как приватный объект (защищённый пин-кодом). Чтобы экспортировать запрос как публичный объект, необходимо указать опцию "non-private".

При экспорте запроса на токен/смарт-карту имя файла запроса должно оканчиваться на ".req".

59.6.11 Управление PKCS10-запросами

PKCS10-запросы сохраняются в системе в специальном хранилище и идентифицируются по именам контейнеров.

Для вывода списка имён контейнеров PKCS10-запросов, сохранённых в системе, необходимо выполнить команду (в режиме "enable"):

```
# show crypto pki pkcs10—reqs
```

Также данная команда выводит DN-имена субъектов PKCS10-запросов.

Полная информация о PKCS10-запросе выводится командой:

```
# show crypto pki pkcs10 <имя_контейнера_pkcs10>
```

Удаление PKCS10-запроса из системы производится командой:

```
# crypto pki clear pkcs10 <имя_контейнера_pkcs10>
```

Чтобы удалить все PKCS10-запросы из системы, можно воспользоваться командой:

```
# crypto pki clear pkcs10—reqs
```

Если необходимо вывести список PKCS10-запросов, находящихся на внешнем носителе, это можно сделать с помощью команды:

```
# show crypto pki pkcs10—reqs <носитель>:[/путь_к_поддиректории]
```

Примеры параметра "внешний носитель": "flash0:", "flash0:/subdir1/subdir2", "token:".

Если в качестве внешнего носителя используется "token:" (USB-токен/смарт-карта), то будет предложено опционально ввести пин-код пользователя токена. Если пин-код не введён, то будут отображены только публичные (не защищённые пин-кодом) PKCS10-запросы.

Подробную информацию о PKCS10-запросе на внешнем носителе можно получить с помощью команды:

```
# show crypto pki pkcs10 <носитель>:/<путь_к_файлу_pkcs10>
```

59.6.12 Dionis-NX, как клиент УЦ DiCert

Криптомаршрутизатор Dionis-NX может взаимодействовать с сервером DiCert по протоколу DCRP (DiCert Request Protocol).

Подробную информацию о протоколе DCRP см. в "Руководстве администратора DiCert".

59.6.12.1 Настройка клиентского модуля DCRP

Для взаимодействия с сервером DiCert необходимо выполнение следующих условий:

- В хранилище пользовательских сертификатов должен присутствовать сертификат субъекта, который присутствует в БД сервера DiCert.
- Сертификат субъекта должен быть действительным.
- В сертификате в расширении KeyUsage должен присутствовать флаг digitalSignature.
- В хранилище закрытых ключей должен присутствовать ключ, соответствующий сертификату субъекта.
- В хранилищах сертификатов корневых и промежуточных УЦ должны присутствовать сертификаты, необходимые для построения цепочки доверия для проверки сертификата DCRP, присылаемого сервером DiCert.
- Если действует строгая политика проверки CRL (см. ниже), в хранилище CRL должны присутствовать все CRL, необходимые для проверки цепочки доверия при проверке сертификата DCRP, присылаемого сервером DiCert.
- Должен быть настроен адрес/порт сервера DiCert (см. ниже).

Чтобы войти в режим редактирования настроек DCRP, необходимо выполнить команду (в режиме “configure”):

```
(config)# crypto pki dicert config
(config—dicert)# _
```

Адрес сервера DCRP настраивается с помощью команды (в режиме “config-dicert”):

```
(config—dicert)# server <ip_адрес>|<доменное_имя> [<порт>]
```

По умолчанию используется порт 11504.

Следующая настройка (в режиме “config-dicert”) задаёт тайм-аут ожидания ответа от DCRP-сервера:

```
(config—dicert)# connection timeout <секунды>
```

Тайм-аут по умолчанию – 60 секунд.

Существует ограничение на максимальный размер DCRP-сообщения от сервера. По умолчанию размер не должен превышать 64 КБ. При превышении клиент принудительно закрывает соединение.

Изменить данное ограничение можно с помощью опции (в режиме “config-dicert”):

```
(config—dicert)# max reply size <байты>
```

По умолчанию проверка DCRP-сертификата, присылаемого сервером DiCert, производится с использованием строгой политики проверки CRL, которая заключается в том, что для каждого сертификата в цепочке доверия должен существовать действительный и непросроченный CRL, с помощью которого можно проверить данный сертификат на отзыв.

Строгую политику проверки CRL можно ослабить с помощью команды (в режиме “config-dicert”):

```
(config—dicert)# crl policy strict off
```

При нестрогой политике допускается отсутствие CRL или использование просроченных CRL.

Вернуть строгую политику можно с помощью команды:

```
(config—dicert)# crl policy strict on
```

59.6.12.2 Выпуск сертификата по DCRP

Для успешного удалённого выпуска сертификата по протоколу DCRP необходимы следующие условия:

- Должны быть выполнены все условия, необходимые для корректной работы протокола DCRP. (См. выше).
- Должен быть выпущен PKCS10-запрос на выпуск нового сертификата субъекта. (См. раздел "Генерация ключа и PKCS10-запроса" выше).
- Выпущенный PKCS10-запрос должен соответствовать политике выпуска сертификатов, ассоциированной с данным субъектом, на сервере DiCert.
- Ключ в PKCS10-запросе должен быть уникален (неизвестен серверу DiCert).

Удалённый выпуск сертификата осуществляется командой (в режиме "enable"):

```
# crypto pki issue cert from <p10_name> to <cert_name> req—sign—by <req_sign_cert_name>
```

Параметры:

- <p10_name> – имя контейнера PKCS10-запроса, сохранённого в системе.
- <cert_name> – имя контейнера в хранилище пользовательских сертификатов, в котором будет сохранён присланный сертификат в случае удачного выпуска.
- <req_sign_cert_name> – имя контейнера в хранилище пользовательских сертификатов, содержащего сертификат, используемого для подписи DCRP-запроса к серверу DiCert.

Имя субъекта в PKCS10-запросе может не соответствовать имени субъекта в сертификате подписи DCRP-запроса, если политикой выпуска сертификатов разрешена смена имени субъекта.

Если выпуск сертификата не удался, то будет выдано подробное диагностическое сообщение, описывающее причину отказа.

Если выпуск сертификата был успешен, но по каким-либо причинам выпущенный сертификат не был доставлен (например, из-за проблем со связью), то его можно получить по протоколу DCRP, воспользовавшись командой "crypto pki retrieve cert".

Количество выпусков сертификатов субъекта по протоколу DCRP (без ручной доставки PKCS10-запроса на сервер DiCert) ограничивается регламентом. При исчерпании данного количества регламент может потребовать осуществить ручную доверенную доставку PKCS10-запроса на сервер DiCert для выпуска очередного сертификата.

59.6.12.3 Получение сертификата по DCRP

Протокол DCRP допускает возможность получения пользовательского сертификата из БД DiCert по его отпечатку ключа или соответствующему PKCS10-запросу.

Команда получения сертификата (в режиме "enable") имеет следующие форматы:

Формат (1):

```
# crypto pki retrieve cert by—pkcs10 <p10_name> to <cert_name> req—sign—by
      <req_sign_cert_name>
```

Формат (2):

```
# crypto pki retrieve cert by—keyid <hex_keyid> to <cert_name> req—sign—by <req_sign_cert_name>
```

Параметры:

- <p10_name> – имя контейнера PKCS10-запроса, сохранённого в системе.
- <cert_name> – имя контейнера в хранилище пользовательских сертификатов, в котором будет сохранён присланный сертификат.
- <hex_keyid> – отпечаток ключа субъекта в шестнадцатеричном виде.
- <req_sign_cert_name> – имя контейнера в хранилище пользовательских сертификатов, содержащего сертификат, используемого для подписи DCRP-запроса к серверу DiCert.

59.6.12.4 Получение CRL по DCRP

В протоколе DCRP предусмотрена возможность получения последнего выпущенного CRL от сервера DiCert. Для идентификации CRL необходим отпечаток ключа его издателя, который может быть введён вручную, либо заимствован из сертификата корневого или промежуточного УЦ.

Команда получения CRL (в режиме "enable") имеет следующие форматы:

Формат (1):

```
# crypto pki retrieve crl by—root—ca—cert <cert_name> to <crl_name> req—sign—by
      <req_sign_cert_name>
```

Формат (2):

```
# crypto pki retrieve crl by—ca—cert <cert_name> to <crl_name> req—sign—by <req_sign_cert_name>
```

Формат (3):

```
# crypto pki retrieve crl by—keyid <hex_keyid> to <crl_name> req—sign—by <req_sign_cert_name>
```

Параметры:

- <cert_name> – имя контейнера сертификата издателя CRL в хранилище сертификатов корневых или промежуточных УЦ.
- <hex_keyid> – отпечаток ключа издателя CRL в шестнадцатеричном виде.
- <crl_name> – имя контейнера в хранилище CRL, в котором будет сохранён присланный CRL.

- <req_sign_cert_name> – имя контейнера в хранилище пользовательских сертификатов, содержащего сертификат, используемого для подписи DCRP-запроса к серверу DiCert.

Команда формата (1) использует для идентификации CRL отпечаток ключа корневого сертификата. Команда формата (2) использует отпечаток ключа сертификата подчинённого УЦ. В команде формата (3) отпечаток ключа издателя вводится вручную.

59.6.12.5 Отзыв сертификата по DCRP

Протокол DCRP позволяет субъекту удалённо отозвать собственный сертификат.

Для успешного отзыва сертификата по протоколу DCRP необходимы следующие условия:

- Должны быть выполнены все условия, необходимые для корректной работы протокола DCRP.
- На сервере DiCert в политике, ассоциированной с субъектом, должна присутствовать опция "auto-revoke".

Отзыв сертификата осуществляется командой (в режиме "enable"):

```
# crypto pki revoke cert <cert_name> [<reason>]
```

Примечание: DCRP-запрос подписывается отзываемым сертификатом.

Параметры:

- <cert_name> – имя контейнера отзываемого сертификата в хранилище пользовательских сертификатов.
- <reason> – причина отзыва.

Возможные причины отзыва:

Параметр	Причина отзыва
key-compromise	Скомпрометирован ключ субъекта
affiliation-changed	Субъект больше не входит в данную PKI
superseded	Для субъекта выпущен новый сертификат
cessation-of-operation	Субъект прекратил свою деятельность в PKI
privilege-withdrawn	Субъект лишён привилегий
certificate-hold	Действие сертификата (временно) приостановлено

Причина отзыва по умолчанию: key-compromise.

59.6.12.6 Получение цепочки доверия по DCRP

По протоколу DCRP можно получить главный выпускающий сертификат сервера DiCert, а также набор всех сертификатов УЦ (вплоть до корневого), необходимых для проверки главного выпускающего

сертификата. Также передаются все необходимые CRL. Сертификаты УЦ сохраняются в хранилищах корневых и подчинённых УЦ. CRL сохраняются в хранилище CRL.

Команда получения цепочки сертификатов и CRL (в режиме "enable") имеет следующий формат:

```
# crypto pki retrieve ca chain to <name_prefix> req—sign—by <req_sign_cert_name>
```

- <name_prefix> – префикс имён контейнеров, в которых сохраняются полученные сертификаты и CRL.
- <req_sign_cert_name> – имя контейнера в хранилище пользовательских сертификатов, содержащего сертификат, используемого для подписи DCRP-запроса к серверу DiCert.

Полученные сертификаты сохраняются в соответствующих хранилищах с именами вида "<name_prefix><N>.cer", где N – число (1, 2, 3, …), увеличиваемое на 1 с каждым новым сохраняемым объектом и выбираемое таким образом, чтобы не произошла перезапись существующего контейнера. CRL сохраняются в хранилище CRL с именами "<name_prefix><N>.crl". Перед сохранением проверяется, существует ли уже объект в хранилище, идентичный сохраняемому. Если существует, то объект повторно не сохраняется.

Информация о сохранённых объектах и назначенных именах выдаётся в виде сообщений "Info:".

Установка корневого сертификата, полученного по протоколу DCRP, считается доверенной, так как протокол DCRP является доверенным. Количество доставок корневых сертификатов по протоколу DCRP ограничивается регламентом. По исчерпании данного количества регламент может потребовать ручную доверенную доставку очередного корневого сертификата.

59.7 Туннели IPsec

IPsec представляет собой набор протоколов для обеспечения защиты данных, передаваемых по межсетевому протоколу IP, посредством шифрования и подтверждения подлинности IP-пакетов. Также средствами IPsec обеспечивается взаимная двусторонняя аутентификация сторон, устанавливающих между собой крипто-туннель.

IPsec состоит из двух протоколов:

- IKE (Internet Key Exchange) - протокол взаимной аутентификации сторон и выработки ключевого материала для протокола ESP;
- ESP (Encapsulating Security Payload) - протокол шифрования и проверки подлинности IP-пакетов, передаваемых через крипто-туннель.

В Dionis-NX реализованы протоколы IKE версии 1 (RFC2407-2409) и ESP (RFC4303) на основе стандарта, разработанного ООО «Крипто-Про», с использованием российских криптоалгоритмов ГОСТ 28147-89, ГОСТ Р 34.10-2001, ГОСТ Р 34.10-2012, ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012.

В IKE реализована взаимная аутентификация на основе инфраструктуры открытых ключей PKI (см. выше), а также аутентификация на основе предварительно распределённых ключей PSK.

В Dionis-NX протокол IKE реализован в виде службы, которая при установлении туннеля формирует симметричный ключевой материал и загружает его в подсистему XFRM ядра Linux. Протокол ESP реализуется подсистемой XFRM и подсистемой TCP/IP на уровне ядра Linux.

59.7.1 Базовые понятия протокола IKEv1

Основная цель протокола IKE - аутентифицировать удалённый узел, с которым требуется установить защищённое соединение, и выработать ключевой материал, используемый для симметричного шифрования IP-пакетов в протоколе ESP.

Протокол IKE представляет собой протокол «рукопожатия». Для обмена пакетами между сторонами в качестве транспорта используется протокол UDP (порты 500, 4500). Протокол IKE состоит из двух основных фаз.

В фазе 1 производится первоначальное согласование криптопараметров, используемых при шифровании IKE-пакетов фазы 1 и 2, а также взаимная аутентификация сторон.

В фазе 2 производится согласование криптопараметров для протокола ESP и выработка ключевого материала для шифрования/проверки подлинности IP-пакетов, передаваемых по туннелю.

59.7.1.1 Фаза 1

В протоколе IKE вводятся понятия *инициатора* и *ответчика*.

Инициатор - это тот узел, который пытается первым установить IPsec-туннель. Ответчик - противоположная (слушающая) сторона.

В зависимости от настроек сторон роли инициатора и ответчика могут быть либо жёстко закреплены за каждым из узлов (модель клиент-сервер), либо стороны могут меняться ролями по своему усмотрению (модель peer-to-peer).

На рисунке 59.1 изображён обмен IKE-пакетами между инициатором и ответчиком на фазе 1 (аутентификация по сертификатам, Main Mode, Aggressive Mode не реализован).

Условные обозначения:

- HDR - Header, заголовок пакета IKE. «*» означает, что пакет IKE зашифрован;
- SAi - Security Association, предложение наборов криптографических и технологических параметров IKE от инициатора ответчику;
- VIDs - идентификаторы VendorID, уведомляющие о дополнительных возможностях: Dead Peer Detection (RFC3706), NAT Traversal (RFC3947), GOST - IPsec по стандарту ООО «Крипто-Про»;
- SAr - конкретный набор параметров, выбранный ответчиком из предложенных, и уведомление инициатора о сделанном выборе;
- KEi/r - Key Exchange, обмен временными открытыми ключами для генерации общего секрета;
- Ni/r - Nonce, обмен случайными значениями для усиления криптографической защиты;
- [] - параметр, который может отсутствовать при определённых настройках сторон;
- NAT-D - NAT Detection, хэши реальных IP-адресов концов туннеля для определения ситуации «IPsec через NAT»;
- CRi - Certificate Request, уведомление ответчиком инициатора о доверяемом УЦ;
- CRr - уведомление инициатором ответчика о доверяемом УЦ;
- IDii/r - X500-имена инициатора и ответчика;
- CERTi/r - сертификаты инициатора и ответчика;

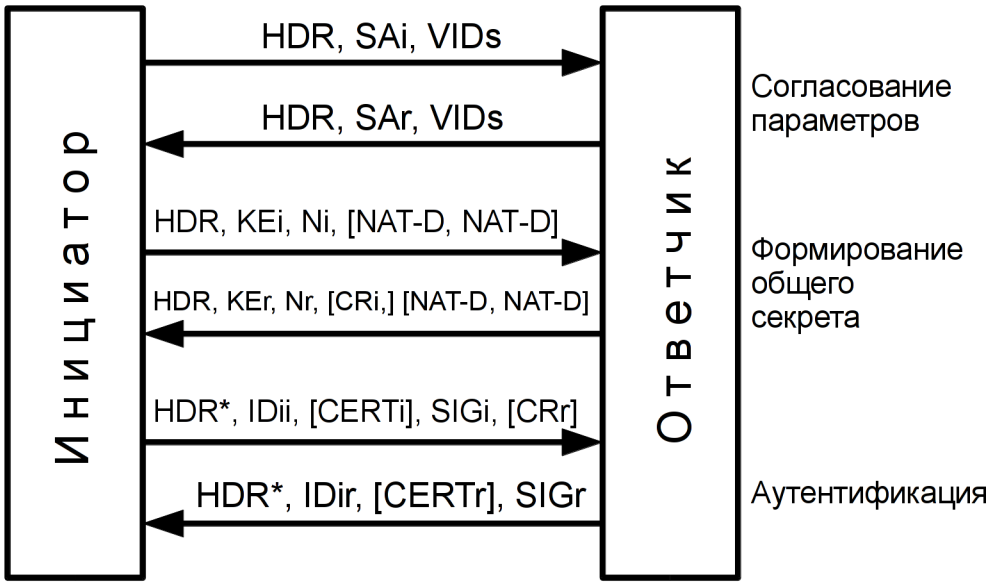

Рис. 59.1: Фаза 1 (PKI)

- SIGi/r - электронно-цифровые подписи инициатора/ответчика, сформированные по ранее переданным параметрам.

Краткое описание фазы 1:

1. Инициатор желает установить защищённое соединение с ответчиком;
2. Инициатор знает IP-адрес ответчика и шлёт ему первый пакет (порт 500), содержащий предлагаемые наборы параметров (криптопараметры для фазы 1 и 2, время жизни фазы 1, и т.д.) и идентификатор, уточняющий реализацию протокола IKE;
3. Если ответчик не поддерживает данную реализацию протокола IKE или не может выбрать ни один набор параметров, соединение не устанавливается;
4. Если ответчик выбрал набор из предлагаемых параметров, он шлёт ответный пакет с выбранным набором;
5. Инициатор генерирует временную пару асимметричных ключей и шлёт открытый ключ ответчику;
6. Ответчик получает временный открытый ключ инициатора, генерирует свою временную асимметричную пару и шлёт открытый ключ инициатору;
7. Ответчик также может послать Certificate Request, то есть X500-имя удостоверяющего центра, чей сертификат должен обязательно присутствовать в цепочке сертификатов при проверки сертификата инициатора;
8. Ответчик вычисляет секрет на основе открытого ключа инициатора и собственного временного закрытого ключа;
9. Инициатор получает временный открытый ключ ответчика и вычисляет секрет на основе полученного открытого ключа и собственного временного закрытого ключа. Секрет инициатора равен секрету ответчика;

10. Инициатор и ответчик обмениваются хэшами своих реальных IP-адресов (NAT-D), чтобы определить ситуацию «IPsec через NAT». Если NAT обнаружен, то последующий обмен (начиная с 3-их пакетов) будет производиться через порт UDP 4500. Также трафик протокола ESP будет инкапсулирован в UDP/4500;
11. Инициатор формирует 3-й пакет из следующих данных:
12. IDii - X500-имя инициатора;
13. Сертификат инициатора (может не передаваться в зависимости от настроек). X500-имя субъекта сертификата должно совпадать с IDii;
14. Электронно-цифровая подпись, вычисленная с помощью закрытого ключа инициатора (соответствующего сертификату) по переданным полям (в том числе IDii), которая удостоверяет, что X500-имя и сертификат действительно принадлежат данному инициатору;
15. Также может быть послан Certificate Request - X500-имя УЦ, сертификат которого должен обязательно присутствовать в цепочке сертификатов при проверки сертификата ответчика;
16. Содержимое 3-го пакета инициатора зашифровывается на симметричном ключе, полученном из общего секрета, значений Ni/r и т.д. Пакет передаётся ответчику;
17. Ответчик расшифровывает пакет;
18. Ответчик анализирует IDii и проверяет (согласно своим настройкам), разрешено ли установить соединение с данным субъектом;
19. Ответчик анализирует полученный сертификат. Если сертификат не передан, то он ищет его в локальном хранилище по имени IDii. Если сертификат не найден, соединение не устанавливается;
20. Если IDii не соответствует X500-имени субъекта сертификата, соединение не устанавливается;
21. Ответчик проверяет срок действия сертификата;
22. Если есть информация о серверах OCSP для данного сертификата, сертификат проверяется на отзыв;
23. Если OCSP недоступен, ищется соответствующий список отзыва для данного сертификата. COC может быть загружен локально или может быть доступен по точкам распространения COC;
24. Если сертификат отозван, соединение не устанавливается;
25. Если OCSP и COC недоступны, и включена строгая политика проверки на отзыв («crl policy strict», см. ниже), то сертификат считается недействительным, и соединение не устанавливается;
26. В локальных хранилищах сертификатов УЦ ищется сертификат УЦ, выпустившего данный сертификат. Если он не найден, соединение не устанавливается;
27. Проверяется подпись сертификата открытым ключом сертификата УЦ. Если подпись не прошла проверку, соединение не устанавливается;
28. Сертификат УЦ проверяется так же, как описано выше;
29. Проверка осуществляется вплоть до корневого сертификата. (На корневой сертификат не распространяется строгая политика проверки на отзыв);
30. Если хотя бы один сертификат оказался недействительным, соединение не устанавливается;
31. Если аутентификация инициатора прошла успешно, ответчик формирует свой 3-й IKE-пакет аналогичным образом и шлёт его инициатору;
32. Инициатор аутентифицирует ответчика аналогично, как описано выше;
33. Если аутентификация ответчика прошла успешно, инициатор переходит к фазе 2.

В случае аутентификации по pre-shared ключам фаза 1 выглядит следующим образом (см. рис. 59.2).

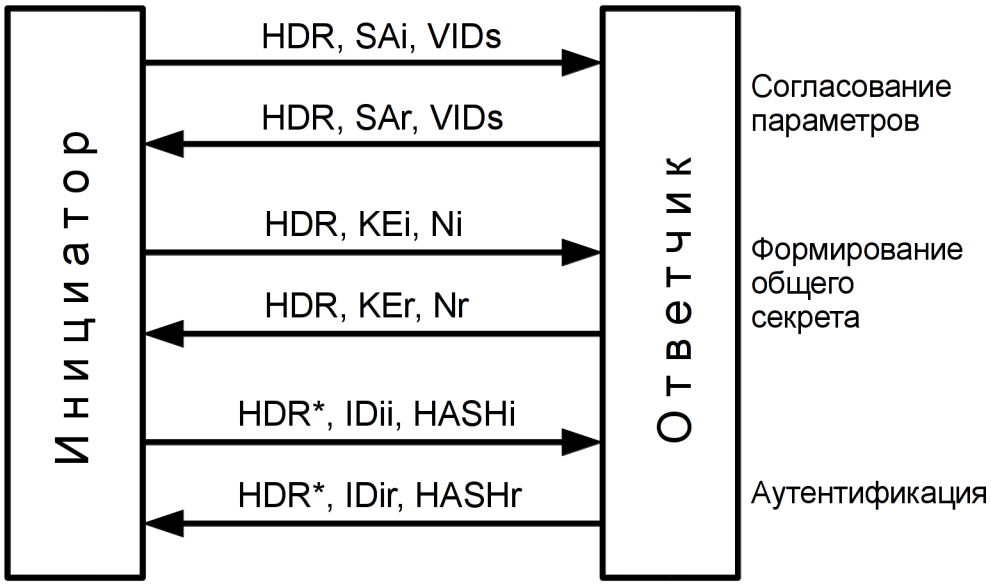

Рис. 59.2: Фаза 1 (PSK)

Условные обозначения:

- IDii/r - IP-адреса концов туннеля (идентификация сторон осуществляется по IP-адресам);
- HASHi/r - хэши, сформированные по pre-shared ключу и по переданным параметрам.

Если pre-shared ключ совпадает на обеих сторонах, то инициатор/ответчик успешно проверят присланные им HASHr и HASHi, соответственно, и аутентификация пройдёт успешно.

59.7.1.2 Фаза 2

Фаза 2 показана на рисунке 59.3.

Условные обозначения:

- HASH(1,2,3) - HMAC по некоторым переданным полям на основе общего секрета (для защиты от подмены);
- SAi - предлагаемые наборы криптографических и технологических параметров ESP ответчику инициатором;
- SAr - выбор ответчиком конкретного набора;
- Ni,r - дополнительные случайные значения для усиления криптографической защиты;
- KEi,r - обмен временными открытыми ключами для формирования дополнительного общего секрета в режиме Perfect Forward Secrecy (PFS);
- IDci - адрес и маска внутренней (защищаемой) подсети, находящейся за инициатором. Также могут быть заданы протокол и порт для конкретизации трафика;

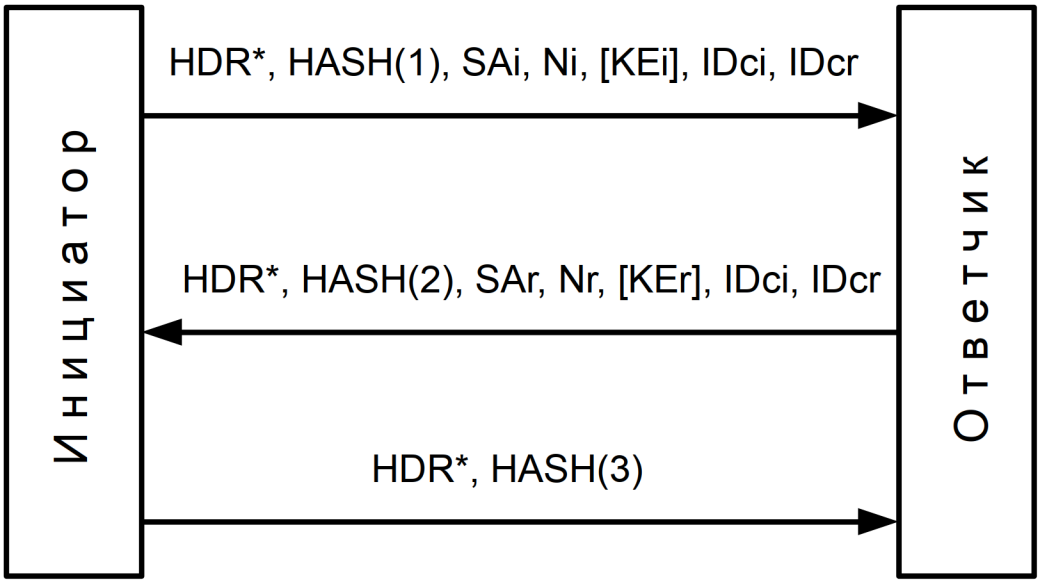

Рис. 59.3: Фаза 2

- IDcr - адрес и маска внутренней (защищаемой) подсети, находящейся за ответчиком. (Опционально - протокол, порт);

Краткое описание фазы 2:

Все пакеты фазы 2 зашифрованы на основе общего секрета, выработанного на фазе 1.

1. Инициатор предлагает наборы параметров (криптопараметры туннеля ESP, время жизни фазы 2/туннеля ESP, и т.д.) ответчику;
2. Также в SAi формируется Security Parameters Index (SPI) туннеля от инициатора к ответчику - SPIir;
3. Также инициатор предлагает адреса/маски внутренних защищаемых подсетей (своей и ответчика). (А также возможна конкретизация протокола и портов. См. пояснение ниже);
4. Если указан протокол, то в туннель будут попадать только трафик данного протокола. Для протоколов TCP/UDP может быть указан порт;
5. Инициатор может передать дополнительный временный открытый ключ (KEi), предложив тем самым режим Perfect Forward Secrecy (PFS). В этом случае ответчик передаёт свой KEr, и производится формирование дополнительного общего секрета (так же, как описано в фазе 1), участвующего в вычислении ключевого материала для ESP;
6. Ответчик получает пакет от инициатора, расшифровывает его и проверяет HASH(1);
7. Ответчик анализирует IDci, IDcr и проверяет, согласуются ли желаемые инициатором подсети/маски/протокол/порт с настройками ответчика. Если нет, соединение не будет установлено;
8. Ответчик анализирует предлагаемые наборы параметров. Если ни один не подходит, соединение не будет установлено;
9. Ответчик выбирает конкретный набор параметров и формирует ответный пакет. В SAr формируется SPI туннеля от ответчика к инициатору - SPIri;

10. Инициатор получает пакет от ответчика, расшифровывает его и проверяет HASH(2);
11. Если успешно, туннель со стороны инициатора считается установленным. Инициатор формирует ключевой материал для туннеля и передаёт его в систему управления протоколом ESP;
12. Инициатор высылает ответчику пакет подтверждения об успешном установлении туннеля;
13. Ответчик получает пакет, расшифровывает и проверяет HASH(3);
14. Если успешно, туннель со стороны ответчика считается установленным. Ответчик формирует ключевой материал для туннеля и передаёт его в систему управления протоколом ESP.

Пояснение к полю протокол/порт в IDci, IDcr:

Действуют следующие правила:

1. Если протокол/порт не заданы в IDci, то их также не должно быть в IDcr (и наоборот). В этом случае весь трафик, идущий из подсети инициатора в подсеть ответчика и обратно, будет идти через туннель. (Исключение составляет трафик UDP/500/4500 - трафик IKE никогда в туннель не попадает);
2. Если задан протокол в IDci, то должен быть задан такой же протокол в IDcr. В этом случае в туннель будет попадать только трафик указанного протокола (из подсети инициатора в подсеть ответчика и обратно);
3. Для протоколов UDP и TCP можно указывать порт;
4. Если порт не указан, в туннель попадает весь трафик TCP/UDP;
5. В IDci/IDcr могут быть указаны разные порты. Также возможна ситуация наличия порта в IDci и отсутствие в IDcr (и наоборот).

В случае указания портов в туннель будет попадать следующий трафик:

- Инициатор ☐ Ответчик: порт отправителя из IDci, порт получателя из IDcr;
- Ответчик ☐ Инициатор: порт отправителя из IDcr, порт получателя из IDci.

Из фазы 1 может быть порождено несколько фаз 2. Обычно следующая фаза 2 порождается незадолго до истечения времени жизни старой фазы 2 (времени жизни туннеля) для обновления ключевого материала туннеля.

59.7.1.3 Фаза ModeConfig

Иногда при использовании модели «клиент - сервер» необходимо назначать клиенту внутренний (виртуальный) IP-адрес из пула сервера. В этом случае параметр фазы 2 IDci не может быть сформирован заранее. Поэтому для назначения клиенту внутреннего IP-адреса используется дополнительная фаза ModeConfig (см. рис. 59.4), которая происходит между фазой 1 и фазой 2.

Условные обозначения:

- ADDR_REQ - запрос клиентского адреса у сервера;
- ADDR_REPLY - назначение адреса клиенту.

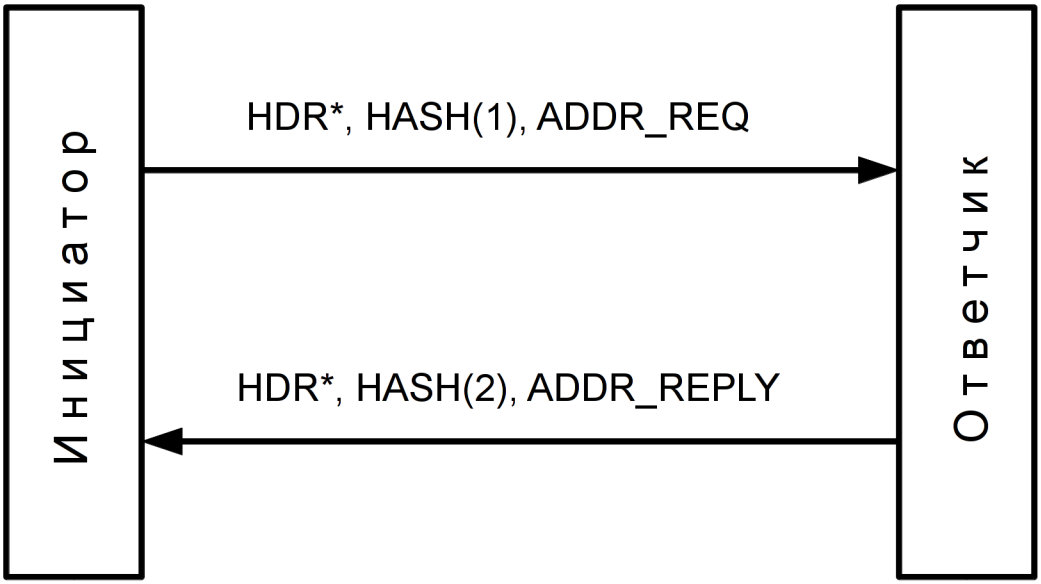

Рис. 59.4: Фаза ModeConfig

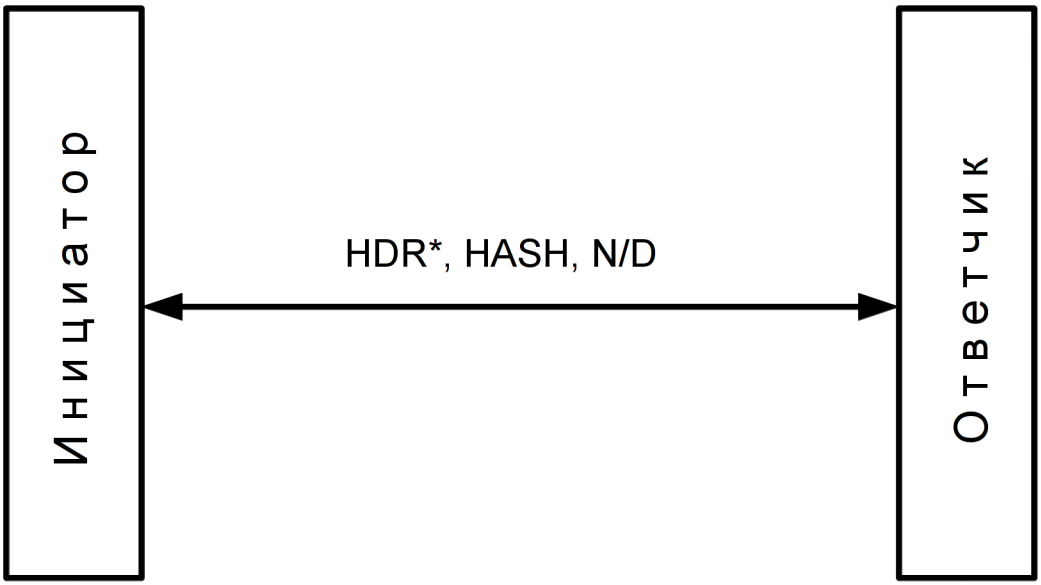

Рис. 59.5: Сообщения уведомлений

59.7.1.4 Уведомление сторон

Стороны могут слать друг другу асинхронные уведомления на любом этапе установления туннеля или в процессе работы туннеля (см. рис. 59.5).

Условные обозначения

- N - Notification. Уведомление (как правило, о причине отторжения пакета, пришедшего от противоположной стороны);
- D - Delete. Требование закрыть активную фазу 1 или 2 (удалить состояние конечного автомата IKE). Удаление фазы 2 равносильно закрытию установленного туннеля.

59.7.2 Базовые понятия протокола ESP

ESP - это IP-протокол с номером 50. В ситуации «IPsec через NAT» протокол ESP инкапсулируется в UDP/4500 (RFC3948).

Протокол ESP инкапсулирует IP-трафик и зашифровывает его на основе симметричных ключей, выработанных протоколом IKE. Также для проверки подлинности передаваемых данных формируется контрольная сумма пакета, выработанная с помощью симметричного ключа.

Содержимое пакета ESP:

- Security Parameters Index (SPI) - идентификатор контекста симметричных ключей (идентификатор туннеля);
- Sequence Number - порядковый номер пакета;
- Init Vector (IV) - синхропосылка для симметричного алгоритма шифрования;
- Зашифрованные данные (выровненные до шифрования);
- Integrity Check Value (ICV) - защищённая контрольная сумма.

При установлении туннеля (протоколом IKE) формируются **два** контекста симметричных ключей (Security Association, SA) для направлений от инициатора к ответчику и от ответчика к инициатору. Эти SA идентифицируются, соответственно, индексами SPIir и SPIri.

Наборы SA для всех туннелей данного узла формируют базу данных SA (SA Database, SAD). Каждый SA-элемент содержит:

- Номер SPI;
- Идентификатор протокола (ESP);
- Идентификаторы криптоалгоритмов и криптопараметров;
- Симметричные ключи для шифрования и проверки подлинности IP-пакетов;
- IP-адреса концов туннеля;
- Режим туннеля;
- Идентификатор соответствия элементу в базе данных IPsec-политик (SPD, см. ниже) - reqid;
- Другую служебную информацию.

Наличие пар SA на обоих криптомаршрутизаторах означает успешно установленный IPsec-туннель.

Политика IPsec (Security Policy, SP) представляет собой набор правил, на основе которых принимается решение о направлении трафика в IPsec-туннель. Совокупность политик на данном узле формирует базу данных политик (SP Database, SPD). Каждый SP-элемент содержит:

- Правило отбора трафика по направлению (in/out/fwd);
- Правило отбора трафика по IP-адресу отправителя;
- Правило отбора трафика по IP-адресу назначения;
- Правило отбора трафика по протоколу/порту (опционально);
- Правило отбора трафика по метке (опционально);
- Приоритет политики;
- Идентификаторы соответствия с SA (reqid, IP-адреса концов туннеля и т.д.);
- Управляющие флаги и другую служебную информацию.

При установленном туннеле в системе присутствуют политики SP и соответствующие им криптоконтексты SA.

К выходящему (открытому) трафику сначала применяются политики SP (направление «out»). Если трафик попадает под критерии определённой политики, то осуществляется попытка поиска соответствующего SA. Если SA не найден, трафик отбрасывается. Если SA найден, то происходит инкапсуляция трафика в пакеты ESP на основе найденного криптоконтекста, и зашифрованный трафик отправляется в систему маршрутизации.

Входящий (зашифрованный) трафик обрабатывается в обратном порядке. Из пакета ESP извлекается SPI, и ищется соответствующий криптоконтекст SA (по SPI и IP-адресам концов туннеля). Если SA не найден, трафик отбрасывается. Если SA найден, производится расшифрование и проверка подлинности инкапсулированного IP-пакета. При неудаче пакет отбрасывается. Ищется политика SP, которая соответствует данному криптоконтексту SA. Анализируются IP-адреса отправителя и назначения и выполняется проверка расшифрованного пакета на соответствие найденной политике (по направлению, IP-адресам, протоколу/порту, метке). Для пакетов, адресованных данному узлу, используется политика с направлением «in». Для транзитных пакетов используется политика с направлением «fwd». Если пакет не удовлетворяет политике SP, то он отбрасывается. Далее выполняется маршрутизация расшифрованного пакета.

Существуют 2 режима инкапсуляции IP-трафика в ESP:

- Транспортный режим;
- Туннельный режим.

В транспортном режиме заменяется заголовок IP-пакета, и шифруется содержимое. В транспортном режиме возможно соединение только типа «точка-точка».

В туннельном режиме инкапсулируется весь исходный IP-пакет. В этом режиме возможны соединения как «точка-точка», так «подсеть-подсеть». Рекомендуется использовать туннельный режим.

59.7.3 Соединения IPsec

Необходимо ввести понятие «соединения IPsec».

Одно соединение IPsec - это одна из следующих конфигураций сети (см. рис. 59.6).

Если требуется направить трафик из нескольких (разных) подсетей в туннель, то необходимо настроить несколько соединений (ограничение протокола IKEv1).

В случае соединения типа «клиенты-сервер» будет порождено несколько подчинённых соединений.

59.7.4 Минимальные настройки IPsec (PKI)

Рассмотрим самый простой пример настройки соединения типа «точка-точка» со взаимной аутентификацией по сертификатам X.509. В данном примере даны минимально необходимые настройки, для установления туннеля IPsec. Более подробная информация будет дана в следующих разделах. Также в следующем разделе показаны минимально необходимые настройки со взаимной аутентификацией по предварительно разложенным ключам (pre-shared keys).

Допустим у нас есть два узла Dionis-NX с IP-адресами 192.168.1.1 и 192.168.2.1.

Настройка узла 1:

Импортируем сертификат узла, сертификат удостоверяющего центра и закрытый ключ узла с внешнего носителя. (См. PKI выше):

```
# crypto pki import key from flash0:/keys/router1.nam
# crypto pki import root ca cert from flash0:/certs/ca.cer
# crypto pki import cert from flash0:/certs/router1.cer
```

Входим в режим конфигурации и запускаем службу IKE:

```
# configure terminal
(config)# crypto ike enable
```

Создаём настройку соединения. Назовём его «t1»:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# _
```

Вид строки приглашения говорит о том, система находится в режиме редактирования настроек соединения «t1».

По умолчанию действует режим аутентификации по сертификатам X.509, что эквивалентно опции:

```
(config—ike—conn—t1)# auth pubkey
```

Задаём IP-адреса концов туннеля - локального и удалённого:

```
(config—ike—conn—t1)# local ip 192.168.1.1
(config—ike—conn—t1)# remote ip 192.168.2.1
```

Задаём имя используемого сертификата:

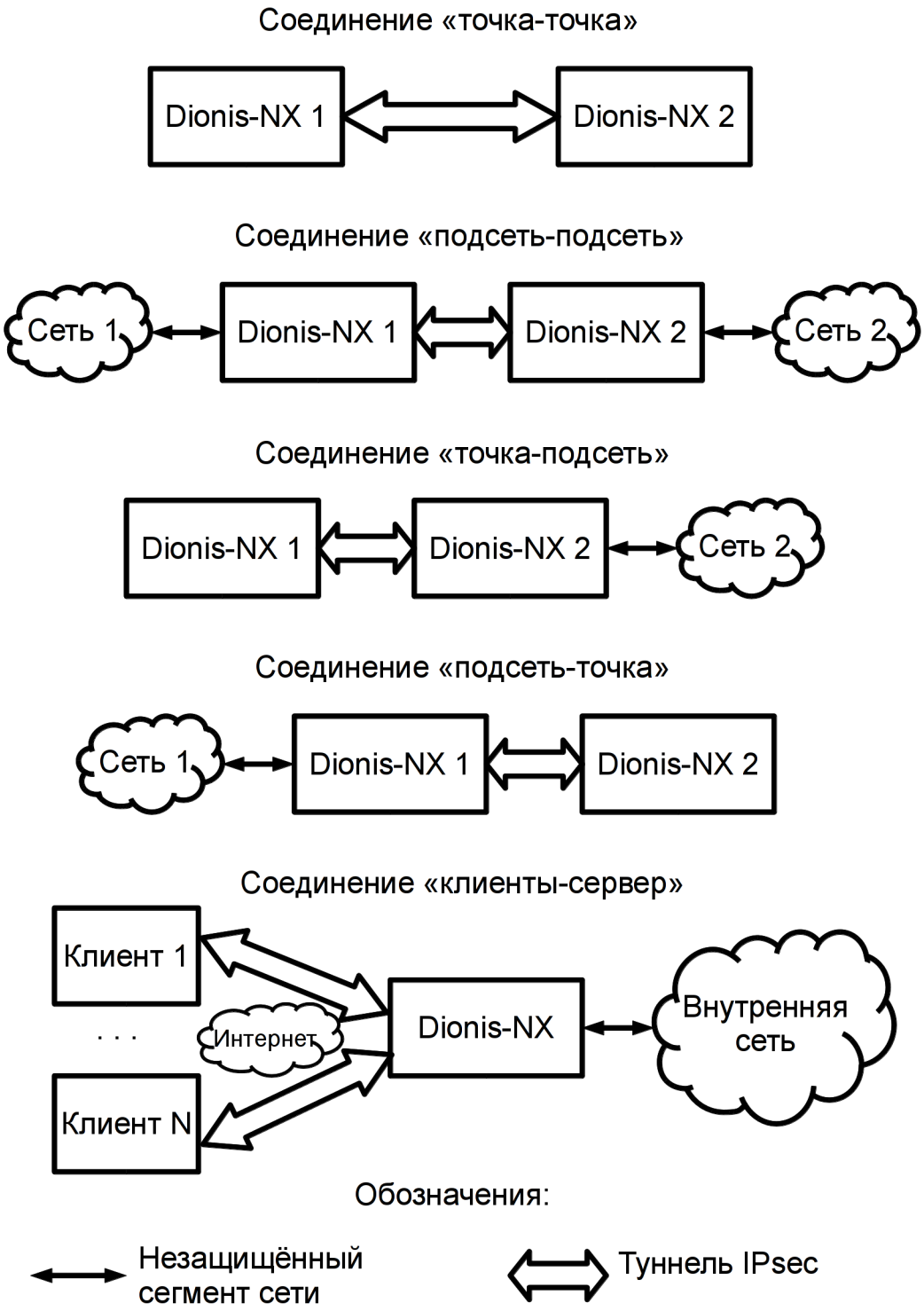

Рис. 59.6: Виды соединений IPsec

```
(config—ike—conn—t1)# local cert router1.cer
```

Задаём X500-имя сертификата нашего оппонента:

```
(config—ike—conn—t1)# remote id ”CN=Узел 2, O=Хорошая организация, C=RU”
```

ПРИМЕЧАНИЕ: Практический опыт показывает, что набор X500-имени вручную является трудоёмкой задачей и часто приводит к опечаткам, из-за которых впоследствии происходит отказ в установлении соединения. Чтобы этого избежать, можно импортировать X500-имя непосредственно из сертификата оппонента. Для этого необходимо предварительно загрузить сертификат оппонента в систему. Пример:

```
(config—ike—conn—t1)# do crypto pki import cert from flash0:/certs/router2.cer
(config—ike—conn—t1)# remote id from cert router2.cer
(config—ike—conn—t1)# exit
(config)# _
```

Минимальная настройка соединения «точка-точка» закончена.

Проверим статус заданного соединения командой режима привилегированного режима «show crypto ike conns»:

```
(config)# do show crypto ike conns
t1      disabled
```

Новые созданные соединения изначально находятся в выключенном состоянии. Чтобы наше соединение смогло стать активным, его необходимо включить:

```
(config)# crypto ike enable conn t1
(config)# do show crypto ike conns
t1      listen
```

Теперь соединение включено и находится в «слушающем» состоянии, то есть оно готово начать установление туннеля IPsec. Установление туннеля может быть инициировано данным узлом (см. ниже), либо может быть инициировано нашим оппонентом.

Теперь выполним настройку узла 2, которая, по сути, будет симметричной настройке узла 1.

```
# crypto pki import key from flash0:/keys/router2.nam
# crypto pki import root ca cert from flash0:/certs/ca.cer
# crypto pki import cert from flash0:/certs/router2.cer
# configure terminal
(config)# crypto ike enable
(config)# crypto ike conn t1
(config—ike—conn—t1)# local ip 192.168.2.1
(config—ike—conn—t1)# remote ip 192.168.1.1
(config—ike—conn—t1)# local cert router2.cer
(config—ike—conn—t1)# remote id ”CN=Узел 1, O=Хорошая организация, C=RU”
(config—ike—conn—t1)# crypto ike enable conn t1
(config)# do show crypto ike conns
t1      listen
```

Теперь оба узла готовы к установлению соединения. В данной конфигурации любой из узлов может стать инициатором.

Инициируем соединение с любого из узлов командой «crypto ike initiate conn» из привилегированного режима:

```
(config)# exit
# crypto ike initiate conn t1
```

Если установление туннеля прошло успешно, то на обоих узлах статус соединения «t1» должен стать «online»:

```
# show crypto ike conns
t1      online
```

Теперь весь трафик (типа «точка-точка») между узлами 1 и 2 будет инкапсулироваться в протокол ESP. Важно помнить, что если к узлу 1, например, подключены другие сети, то проходящий трафик через узел 1 к узлу 2 из этих сетей НЕ будет попадать в туннель и (если не настроены фильтры) будет идти в открытом виде. Ибо данный трафик будет являться трафиком типа «подсеть-точка» и не будет попадать в туннель типа «точка-точка».

59.7.5 Минимальные настройки IPsec (PSK)

Минимальные настройки IPsec с аутентификацией по pre-shared ключам выглядят ещё проще. (За основу взят пример из предыдущего раздела).

Настройка узла 1:

Загружаем pre-shared ключ с внешнего носителя (допустим, ключ №1 из контейнера DSRF):

```
# crypto psk set key psk1 dsrf flash0 1
```

Ассоциируем загруженный ключ с концами туннеля:

```
# configure terminal
(config)# crypto psk map 192.168.1.1 192.168.2.1 psk1
```

Создаём соединение, указываем метод аутентификации по PSK и IP-адреса концов туннеля:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# auth psk
(config—ike—conn—t1)# local ip 192.168.1.1
(config—ike—conn—t1)# remote ip 192.168.2.1
```

Включаем туннель и службу IKE:

```
(config—ike—conn—t1)# crypto ike enable
(config)# crypto ike enable conn t1
(config)# do show crypto ike conns
t1      listen
```

Выполняем симметричные настройки узла 2:

```
# crypto psk set key psk1 dsrf flash0 1
# configure terminal
(config)# crypto psk map 192.168.2.1 192.168.1.1 psk1
(config)# crypto ike conn t1
(config—ike—conn—t1)# auth psk
(config—ike—conn—t1)# local ip 192.168.2.1
(config—ike—conn—t1)# remote ip 192.168.1.1
(config—ike—conn—t1)# crypto ike enable
(config)# crypto ike enable conn t1
```

Инициируем соединение с любого из узлов:

```
(config)# do crypto ike initiate conn t1
(config)# do show crypto ike conns
t1      online
```

59.7.6 IPsec и фильтрация

ВАЖНО помнить, что подсистема IPsec не занимается фильтрацией трафика. Подсистема IPsec только принимает решение, направлять ли его в туннель. Решение принимается на основе настроек соединений (опции «local/remote ip», «local/remote subnet», «local/remote source ip», «local/remote protoport» - см. ниже) и на основе текущего состояния соединения. Например, если соединение неактивно (находится в состоянии «listen»), то трафик не будет направлен в туннель, но будет пропущен, как есть.

Из этого следует, что совместно с настройкой IPsec **необходимо** настроить соответствующую фильтрацию.

Пример настройки фильтрации без использования NAT-traversal:

```
# configure terminal
(config)# ip access—list ipsec_only
(config—acl—ipsec_only)# permit udp sport 500 dport 500
(config—acl—ipsec_only)# permit esp
(config—acl—ipsec_only)# deny
(config—acl—ipsec_only)# interface ethernet 0
(config—if—ethernet0)# ip access—group ipsec_only in
(config—if—ethernet0)# ip access—group ipsec_only out
```

Пример настройки фильтрации с использованием NAT-traversal:

```
# configure terminal
(config)# ip access—list ipsec_in
(config—acl—ipsec_in)# permit udp dport 500
(config—acl—ipsec_in)# permit udp dport 4500
(config—acl—ipsec_in)# permit esp
(config—acl—ipsec_in)# deny
(config—acl—ipsec_in)# ip access—list ipsec_out
```

```
(config—acl—ipsec_out)# permit udp sport 500
(config—acl—ipsec_out)# permit udp sport 4500
(config—acl—ipsec_out)# permit esp
(config—acl—ipsec_out)# deny
(config—acl—ipsec_out)# interface ethernet 0
(config—if—ethernet0)# ip access—group ipsec_in in
(config—if—ethernet0)# ip access—group ipsec_out out
```

В данных примерах мы предполагаем, что интерфейс «ethernet 0» подключён к внешней (опасной) сети. И мы фильтруем весь трафик, проходящий через этот интерфейс, за исключением трафика IKE и ESP.

59.7.7 Фрагментация пакетов IKE, настройки MTU, TCP/MSS

Протокол IKEv1 обладает некоторыми особенностями, которые необходимо учитывать при настройке параметров маршрутизаторов.

В большинстве случаев на каждый пакет IKE ставится флаг DF (Don’t Fragment), если размер данного пакета не превышает значения MTU (Maximum Transfer Unit).

В случае аутентификации с помощью сертификатов X.509 в 5-м и 6-м пакетах фазы 1 могут передаваться сертификаты, а также цепочки сертификатов. Так как сертификаты могут иметь большой размер, 5-й и 6-й пакеты могут превысить стандартное значение MTU 1500 байт, и в этом случае флаг DF на данные пакеты не устанавливается, и пакеты фрагментируются.

Следует отметить, что некоторые маршрутизаторы не позволяют прохождение фрагментированных пакетов IKE и их отфильтровывают. Данную проблему можно решить отключением посылки сертификатов в 5-м/6-м пакетах (с помощью настройки “send cert never”) и использованием сертификатов оппонента из локального хранилища (указав опцию “remote cert”).

Настоятельно не рекомендуется решать проблему прохождения больших сертификатов путём увеличения значения MTU. Значение MTU не должно превышать минимального значения MTU всех маршрутизаторов, через которые будет проходить IKE-пакет. Иначе при больших значениях MTU на большие IKE-пакеты может быть выставлен флаг DF, и впоследствии данный пакет будет отфильтрован маршрутизатором с меньшим MTU.

Для пакетов протокола ESP флаг DF наследуется из оригинального инкапсулированного IP-пакета.

В Dionis-NX существует некоторая особенность инкапсуляции трафика TCP в туннели IPsec для соединений типа “точка-точка”, “точка-подсеть”, “подсеть-точка”. Например, протокол SSH использует DF флаги для своих TCP-пакетов. Если клиентом и/или сервером SSH является узел Dionis-NX, и данный протокол инкапсулируется в туннель IPsec, то необходимо настроить параметр MSS (Maximum Segment Size) для TCP-соединений узла Dionis-NX. Например, в случае MTU 1500 параметр MSS не должен превышать 1400.

Пример настройки MSS:

```
# configure terminal
(config)# ip mangle—list mss
(config—mangle—mss)# mangle adjust—mss 1400 tcp
```

```
(config—mangle—mss)# ip mangle—group mss local—in
(config)# ip mangle—group mss local—out
```

59.7.8 Управление и диагностика службы IKE

Управление туннелями IPsec осуществляется через службу IKE.

59.7.8.1 Запуск/останов службы

Служба IKE может находиться в двух состояниях: остановленном (по умолчанию) и запущенном.

При остановленной службе IKE установление IPsec-туннелей невозможно. Чтобы запустить службу, необходимо выполнить команду режима конфигурации:

```
(config)# crypto ike enable
```

По данной команде служба запускается, загружает все закрытые ключи и сертификаты УЦ из локальных хранилищ, активирует включённые («enabled») соединения (см. ниже) и начинает «слушать» на портах 500/4500 протокола UDP (на всех интерфейсах) входящие IKE-запросы. Если по каким-то причинам запуск службы оказался неудачным (не загружен ключ доступа, фатальная ошибка при активации соединения и т.д.), служба переводится в остановленное состояние.

Остановить службу можно командой режима конфигурации:

```
(config)# crypto ike disable
```

При останове службы IKE закрываются все IPsec-соединения.

59.7.8.2 Диагностика службы

В процессе работы теоретически могут возникать ситуации, когда служба IKE может завершаться аварийно. В этом случае она автоматически переходит в остановленное состояние. В случае возникновения таких ситуаций следует уведомить разработчиков.

Чтобы узнать текущее состояние службы IKE, необходимо выполнить команду режима enable:

```
# show crypto ike status
```

В процессе своей работы служба IKE ведёт журнал. Просмотреть журнал можно с помощью команды режима enable:

```
# show crypto ike log [параметры]
```

Команда без параметров выдаёт последние 25 строк журнала.

Возможные параметры команды «show crypto ike log» (могут использоваться в различных комбинациях):

- nostamp - выводить только текст журнала, без временных заголовков;

- all - вывести весь файл журнала;
- number <n> - вывести последние n строк журнала;
- follow - следить за журналом в реальном времени (Ctrl-C - выход из режима);
- archive <n> - посмотреть архивный файл журнала (чем n больше, тем старше архив).

Чтобы очистить журнал (и все архивы), необходимо выполнить команду:

```
# crypto ike clear log
```

59.7.8.3 Глобальные настройки службы

У службы IKE есть ряд глобальных настроек (см. разделы ниже). Чтобы отредактировать эти настройки, надо войти в режим настроек IKE с помощью команды режима configure:

```
(config)# crypto ike config
```

Следует помнить, что если настройки редактируются при запущенной службе IKE, они **не** применяются немедленно. Чтобы они применились, необходимо перезапустить службу (что повлечёт за собой закрытие всех туннелей):

```
(config)# crypto ike disable
(config)# crypto ike enable
```

59.7.8.4 Особенности обновления PKI/PSK-данных

Следует помнить, что изменения в PKI/PSK (сертификаты, ключи, COC, cainfo и др.) не повлияют на запущенную службу IKE, так как она хранит все PKI/PSK-данные в оперативной памяти. Если требуется перезагрузить PKI-данные без перезапуска службы IKE, нужно выполнить команду режима enable:

```
# crypto ike reload
```

Однако, если были изменены глобальные настройки службы IKE, команда «crypto ike reload» выполнит **полный** перезапуск службы IKE (с предварительным предупреждением).

ВАЖНО: Следует помнить, что команда «crypto ike reload» перезагружает только сертификаты УЦ. Чтобы перезагрузить клиентские сертификаты, следует выключить/включить соответствующие соединения (см. ниже).

59.7.8.5 Удаление всех настроек IKE

Чтобы удалить все настройки службы IKE со всеми настройками соединений, нужно выполнить команду режима configure:

```
# no crypto ike
```

При этом все активные соединения закрываются, и служба IKE останавливается.

59.7.9 Управление и диагностика состояний соединений

59.7.9.1 Создание соединения

Как уже было сказано выше, новое соединение создаётся командой режима configure:

```
(config)# crypto ike conn <имя>
(config—ike—conn—<имя>)# _
```

Данная команда служит как для создания новых соединений, так и для редактирования настроек существующих соединений. Соединение IPsec идентифицируется по имени.

Новое соединение создаётся в выключенном («disabled») состоянии (см. ниже).

59.7.9.2 Удаление соединения

Если требуется удалить соединение со всеми настройками, используется команда режима configure:

```
(config)# no crypto ike conn <имя>
```

Если соединение было активно, то перед удалением оно закрывается.

59.7.9.3 Состояния соединений

Соединения IPsec могут находиться в различных состояниях.

Чтобы вывести состояния всех соединений, нужно выполнить команду режима enable:

```
# show crypto ike conns
```

Пример вывода:

```
t1      disabled
t2      listen
t3      routed
t4 [1] online    CN=Иванов Иван Иванович,O=Хорошая организация,C=RU
t4 [2] online    CN=Петров Пётр Петрович,O=Хорошая организация,C=RU
```

Несколько соединений с одним именем и разными номерами, это порождённые соединения от соединения типа «клиенты-сервер» (см. ниже).

Для вывода состояния одного соединения можно использовать команду:

```
# show crypto ike conn <имя_соединения> [<номер_порождённого_соединения>]
```

Краткое описание состояний соединения:

disabled	Соединение выключено;
enabled	Соединение помечено, как «влючённое», но служба IKE остановлена;

unresolved	Настройки соединения содержат доменные имена, которые не были разрешены в IP-адреса;
invalid	После разрешения IP-адресов выявлены некорректные настройки соединения. Соединение заблокировано;
listen	Соединение включено, но неактивно;
routed	Соединение неактивно, но загружены политики IPsec (SP);
pending1	Соединение пытается стать активным, но не завершилась IKE фаза 1;
pending_mdcfg	Не завершилась фаза ModeConfig, или ожидается начало фазы 2;
pending2	Не завершилась фаза 2;
online	Соединение активно. Установлен IPsec-туннель;
unknown	Внутренняя ошибка.

59.7.9.4 Состояние “disabled”

Состояние «disabled» равносильно отсутствию соединения, как такового. В данном состоянии существуют только настройки соединения, но никакого влияния на систему они оказывать не будут. Настройки соединения следует редактировать именно в этом состоянии. Позволяется редактировать настройки и в других состояниях, но следует помнить, что они не будут применены немедленно. Чтоб они вступили в силу, потребуется сначала перевести соединение обратно в состояние «disabled», а потом снова в состояние «enabled». Если настройки производятся в состоянии «enabled», но при остановленной службе IKE, то выключать/включать соединение не требуется (но требуется запустить службу IKE - см. выше).

В состояние «disabled» соединение переводится командой режима configure:

```
(config)# crypto ike disable conn <имя>
```

Если соединение было активно, то данная команда закрывает IPsec-туннель.

59.7.9.5 Состояние “enabled”

После редактирования настроек соединения, необходимо его включить командой режима configure:

```
(config)# crypto ike enable conn <имя>
```

При остановленной службе IKE соединение будет только помечено, как включённое («enabled»). После запуска службы IKE будет сделана попытка перевести соединение в состояние, согласно настройке «auto» (см. ниже).

При активации соединения при запущенной службе IKE сразу будет сделана попытка перевести соединение в состояние, согласно настройки «auto».

Активация соединения с помощью команды «crypto ike enable conn» также загружает в память соответствующий клиентский сертификат. Если во время включённого соединения сертификат был заменён

с помощью команды «crypto pki import cert», то для того, чтобы он вступил в силу, надо выключить/включить соединение.

59.7.9.6 Состояние “unresolved”

Если соединение содержит доменные имена и/или ссылку на интерфейс (см. «Динамическое разрешение адресов» ниже), и IP-адреса не могут быть разрешены немедленно, то соединение попадёт в состояние «unresolved» и будет находиться в нём до тех пор, пока не будут получены все необходимые IP-адреса. В состоянии «unresolved» инициирование соединения невозможно. При разрешении всех IP-адресов соединение перейдёт в соответствующее состояние согласно настройке «auto».

59.7.9.7 Состояние “invalid”

Если соединение находилось в состоянии «unresolved», и произошло успешное разрешение IP-адресов, но при этом выяснилось, что соединение настроено некорректно, то оно попадает в состояние «invalid». В этом состоянии соединение никогда не станет активным. Для того, чтобы вывести соединение из этого состояния, необходимо его перевести в состояние «disabled» (или остановить службу IKE) и отредактировать настройки соответствующим образом для устранения некорректности. Чтобы диагностировать причину состояния «invalid», необходимо просмотреть журнал службы IKE. В частности, состояние «invalid» может быть вызвано ситуацией, описанной в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

59.7.9.8 Настройка “auto”

Настройка «auto» задаётся для каждого соединения в режиме редактирования его настроек. Например:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# auto route
```

Существуют 4 значения настройки «auto», в зависимости от которых осуществляется попытка перевести соединение в соответствующее состояние автоматически после включения соединения.

Настройка	Желаемое состояние	Состояние при неудаче	Эквивалент ручной команды
auto listen	listen	listen	crypto ike close conn <имя>
auto route	routed	listen	crypto ike route conn <имя>
auto initiate	online	pending, listen	crypto ike initiate conn <имя>
auto route initiate	online	pending, routed	crypto ike route conn <имя>, crypto ike initiate conn <имя>

По умолчанию действует настройка «auto listen».

59.7.9.9 Состояние “listen”

В состоянии «listen» соединение неактивно, но готово к установлению. Соединение может быть инициировано либо со стороны данного узла (командой enable-режима «crypto ike initiate conn»), либо со стороны оппонента. Также это состояние можно назвать «слушающим».

Чтобы принудительно привести соединение в состояние «listen» из других состояний, можно выполнить команду enable-режима:

```
# crypto ike close conn <имя>
```

Данная команда закрывает туннель (удаляет криптоконтексты SA), очищает состояния конечных автоматов соответствующих фаз 1 и 2 и удаляет соответствующие политики SP.

59.7.9.10 Состояние “routed”

Состояние «routed» похоже на состояние «listen» за исключением того, что инсталлируются специальные политики в SPD. Если будет обнаружен трафик, удовлетворяющий правилам отбора в данный туннель, то это приведёт к автоматическому инициированию соединения. Следует отметить, что первые пакеты трафика (до установления соединения) будут отброшены, так как ещё не существует соответствующих криптоконтекстов SA.

Примечание: Повторное инициирование соединения по трафику возможно только по прошествии 60 секунд после начала предыдущего инициирования по трафику.

Чтобы принудительно привести соединение в состояние «routed» из других состояний, можно выполнить команду enable-режима:

```
# crypto ike route conn <имя>
```

Данная команда закрывает туннель (если он был активным) и инсталлирует политики SP для перехвата трафика для инициации соединение.

59.7.9.11 Ручная инициация соединения. Состояние “online”

Чтобы принудительно инициировать соединение, следует выполнить команду enable-режима:

```
# crypto ike initiate conn <имя>
```

В случае успешной инициации соединение переходит в состояние «online», не задерживаясь в состояниях «pending*». В состоянии «online» на обоих концах туннеля существуют пары криптоконтекстов SA.

В случае неуспешной инициации команда «crypto ike initiate conn» будет продолжать попытки установления соединения, блокируя при этом консоль примерно в течение 1 минуты. Далее консоль будет разблокирована, однако соединение будет продолжать попытки установиться (см. раздел «Таймеры»). Разблокировать консоль можно нажатием клавиш Ctrl-C.

Чтобы избежать блокировки консоли при принудительной инициации соединения, следует выполнить команду с параметром «async».

```
# crypto ike initiate conn <имя> async
```

59.7.9.12 Состояния “pending”. Причины неудачного соединения

Состояние «pending1» означает, что IKE-фаза 1 не смогла успешно завершиться. Это может быть вызвано следующими причинами:

- Нет связи с оппонентом;
- Несовместимость систем IPsec между собой (уведомление ATTRIBUTES_NOT_SUPPORTED);
- Оппонент не поддерживает предлагаемые криптопараметры IKE (уведомление NO_PROPOSAL_CHOSEN);
- X500-имя инициатора/ответчика отвергнуто противоположной стороной (уведомление INVALID_ID_INFORMATION);
- Не найден собственный закрытый ключ для формирования подписи (уведомление AUTHENTICATION_FAILED);
- Неправильная ЭЦП от оппонента (уведомление AUTHENTICATION_FAILED);
- Невозможно проверить сертификат оппонента из-за отсутствия, недействительности, отзыва, неполной цепочки сертификатов и т.д. (уведомление AUTHENTICATION_FAILED);
- Неправильная область применения сертификата (уведомление INVALID_CERTIFICATE);
- Оппонент находится в чёрном списке (уведомление SUBJECT_BLACKLISTED);
- Рассинхронизация криптоконтекста из-за выше перечисленных причин или из-за внутренней ошибки (уведомление PAYLOAD_MALFORMED).

Состояние «pending_mdcfg» означает незавершённую фазу выдачи виртуального адреса мобильному клиенту. Также со стороны ответчика данное состояние может означать ожидание начала (или неудачное начало) фазы 2. Это может быть вызвано следующими причинами:

- Неправильные правила пула (см. команду ‘crypto ike pool’). Уведомление ADDRESS_NOTIFICATION.
- Пул не имеет свободных адресов для данного субъекта. Уведомление ADDRESS_NOTIFICATION.
- Невозможно выдать данному клиенту значение серверной подсети. Уведомление ADDRESS_NOTIFICATION.

Состояние «pending2» означает незавершённость фазы 2. Это может быть вызвано следующими причинами:

- Оппонент не поддерживает предлагаемые криптопараметры ESP (уведомление NO_PROPOSAL_CHOSEN);
- Конфигурации правил отбора трафика (подсети, протокол, порт) оппонентов не совпадают (уведомление INVALID_ID_INFORMATION);
- Неправильная область применения сертификата (уведомление INVALID_CERTIFICATE);
- Рассинхронизация криптоконтекста (уведомление PAYLOAD_MALFORMED);
- Мобильный клиент не инициировал фазу ModeConfig, и ему не был назначен виртуальный адрес (уведомление ADDRESS_NOTIFICATION);
- Соединение на данный момент не разрешено расписанием (уведомление PROHIBITED_BY_SCHEDULE).

59.7.9.13 Особенности инициирования соединений из состояний “listen” и “routed”

Существует разница между инициированием соединений из состояний «listen» и «routed». Если соединение было инициировано из состояния «listen», то при закрытии данное соединение будет возвращено в состояние «listen». Если соединение было инициировано из состояния «routed», то при закрытии оно будет возвращено в состояние «routed». Если необходимо автоматически инициировать соединение при запуске службы IKE, и также необходимо, чтобы соединение возвращалось в состояние «routed» при его закрытии, то вместо опции «auto initiate» необходимо указать опцию «auto route initiate».

59.7.9.14 Причины инициирования и закрытия соединений

Причины, вызывающие инициирование соединения:

- Ручная команда «crypto ike initiate conn»;
- Настройка «auto initiate» или «auto route initiate»;
- Инициация соединения со стороны оппонента;
- Исходящий трафик, попадающий в правила отбора (из состояния «routed»);
- Настройка «dpd; action initiate» или «action route initiate» (если соединение было закрыто по Dead Peer Detection, см. ниже).

Причины, вызывающие закрытие соединения:

- Ручная команда «crypto ike close conn»;
- Закрытие соединения со стороны оппонента (см. примечание ниже);
- Истечение времени жизни туннеля и отсутствие инициативы его продления хотя бы с одной стороны (см. опцию «no rekey» ниже);
- Закрытие соединения по Dead Peer Detection (при настройках «dpd action close» или «action route»);
- Внесение оппонента в чёрный список;
- Наступление времени запрета соединения согласно расписанию;
- Деактивация соединения командой «crypto ike disable conn»;
- Останов службы IKE.

ПРИМЕЧАНИЕ: Если соединение было инициировано с нашей стороны (одной из «initiate» настроек/команд), то при закрытии со стороны оппонента оно будет заново инициировано нашей стороной. Если соединение было инициировано с нашей стороны по первому исходящему трафику из состояния «routed», то при закрытии соединения со стороны оппонента оно будет переведено обратно в состояние «routed», и инициации с нашей стороны не последует (до нового исходящего трафика). Во всех остальных случаях при закрытии соединения со стороны оппонента оно будет переведено в исходное пассивное состояние («routed» или «listen»).

59.7.9.15 Диагностика соединений

Информацию о процессе установления соединения можно посмотреть в журнале службы IKE (см. команду «show crypto ike log» выше). Также в журнале записываются принятые уведомления, перечисленные выше. Для вывода в журнал более подробной отладочной информации (кроме криптографической) можно включить опцию:

```
(config)# crypto ike config
(config—ike)# debug control
```

В штатном режиме рекомендуется использовать опцию по умолчанию:

```
(config—ike)# no debug
```

Чтобы вывести подробную информацию о состоянии соединения, можно выполнить команду enable-режима:

```
# show crypto ike conn <имя> [<номер_экземпляра_соединения>] verbose
```

Идентификаторы типа «STATE_MAIN_I4» или «STATE_QUICK_R1» обозначают состояния фаз IKE, где:

- MAIN - фаза 1 (Main Mode);
- QUICK - фаза 2 (Quick Mode);
- I - инициатор;
- R - ответчик;
- цифра - порядковый номер состояния конечного автомата для данной фазы.

Существует возможность вывести время работы туннеля (непрерывного нахождения соединения в состоянии “online”) и количество трафика, переданного/принятого через туннель, с помощью команд:

```
# show crypto ike conns stats
# show crypto ike conn <name> [<n>] stats
```

Также существует возможность ведения дополнительного журнала (упрощённого формата), в который могут записываться следующие события:

- Успешное установление соединения;
- Закрытие соединения;
- Отказ в соединении из-за неудачной аутентификацией оппонента;
- Отказ в соединении из-за ошибки согласования параметров;
- Отказ в соединении из-за нахождения оппонента в “чёрном” списке;
- Отказ в соединении из-за несоответствия расписанию.

Каждое событие записывается в виде одной строки. Запись содержит следующую информацию:

- Дата/время события;
- Имя соединения (и номер экземпляра для шаблонных соединений);

- Тип события;
- IP-адрес оппонента;
- Имя субъекта оппонента;
- Виртуальный адрес, выданный оппоненту (для мобильных клиентов), или адрес внутренней сети, защищаемой оппонентом;
- Адрес внутренней сети, защищаемой данным узлом, если она назначена согласно правилам пула;
- Время нахождения соединения в состоянии “online” (при закрытии соединения);
- Количество переданного/принятого трафика (при закрытии соединения);
- Уведомление (notification), посланное оппоненту (в случае отказа).

Управление типами записываемых событий осуществляется командой:

```
(config)# crypto ike config
(config—ike)# auth—log <параметры>
```

Допустимые параметры: connect, close, auth, negotiation, blacklist, schedule, all (все события).

Остановить запись событий можно с помощью команды:

```
(config—ike)# no auth—log
```

Просмотреть лог событий соединений можно с помощью команды enable-режима:

```
# show crypto ike auth—log [<параметры>]
```

Параметры аналогичны параметрам команды “show crypto ike log”.

Удалить лог событий соединений можно командой:

```
# crypto ike clear auth—log
```

59.7.9.16 Диагностика политик и криптоконтекстов IPsec

Чтобы вывести базу данных политик IPsec (SPD), нужно выполнить команду:

```
# show crypto xfrm policy [verbose]
```

Чтобы вывести базу данных криптоконтекстов (SAD), нужно выполнить команду:

```
# show crypto xfrm state [verbose]
```

59.7.9.17 Вывод всех настроек соединения

Чтобы вывести полную конфигурацию соединения вместе с настройками по умолчанию, нужно выполнить команду:

```
# show crypto ike conn <имя> config
```

59.7.10 Копирование настроек соединений

Так как IKEv1 не поддерживает множественные правила отбора, то часто возникает необходимость создать новое соединение на основе настроек старого. В этом случае рекомендуется создать новое соединение путём копирования из старого. Это делается с помощью команды режима configure:

```
(config)# crypto ike copy conn <старое_соединение> to <новое_соединение>
```

Новое соединение создаётся в выключенном состоянии («disabled»).

59.7.11 Туннельный и транспортный режим

По умолчанию IPsec-туннель будет работать в туннельном режиме. Если необходимо включить транспортный режим, то следует указать опцию в режиме конфигурации соединения:

```
(config)# crypto ike conn <имя>
(config—ike—conn—<имя>)# type transport
```

Вернуть туннельный режим можно командой:

```
(config—ike—conn—<имя>)# type tunnel
```

Без необходимости данную настройку менять не нужно.

Также следует помнить, что данная настройка имеет смысл для соединений типа «точка-точка». Для соединений других типов (с подсетями) будет всегда действовать туннельный режим (например, если присутствуют опции «remote subnet» или «local subnet»).

59.7.12 Соединения “подсеть-подсеть” и “точка-подсеть”

Для того, чтобы соединение работало в режиме «подсеть-подсеть» (см. выше), нужно указать адреса/маски локальной и удалённой подсетей, защищаемых криптомаршрутизаторами. Это делается с помощью опций режима конфигурации соединения:

```
(config—ike—conn—<имя>)# local subnet <IP/mask>
(config—ike—conn—<имя>)# remote subnet <IP/mask>
```

Пример:

Допустим, существуют два криптомаршрутизатора: Dionis1 и Dionis2.

Dionis1 защищает внутреннюю сеть 10.1.0.0/16, а Dionis2 - свою внутреннюю сеть 10.2.0.0/16.

Dionis1 и Dionis2 образуют криптотуннель, IP-адреса концов которого 11.1.0.1 и 11.2.0.1 соответственно.

Настройка Dionis1:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# auto route
(config—ike—conn—t1)# local cert dionis1.cer
(config—ike—conn—t1)# remote id ”CN=Dionis 2, O=Хорошая организация, C=RU”
(config—ike—conn—t1)# local ip 11.1.0.1
(config—ike—conn—t1)# remote ip 11.2.0.1
(config—ike—conn—t1)# local subnet 10.1.0.0/16
(config—ike—conn—t1)# remote subnet 10.2.0.0/16
(config—ike—conn—t1)# exit
(config)# crypto ike enable conn t1
```

Настройка Dionis2:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# auto route
(config—ike—conn—t1)# local cert dionis2.cer
(config—ike—conn—t1)# remote id ”CN=Dionis 1, O=Хорошая организация, C=RU”
(config—ike—conn—t1)# local ip 11.2.0.1
(config—ike—conn—t1)# remote ip 11.1.0.1
(config—ike—conn—t1)# local subnet 10.2.0.0/16
(config—ike—conn—t1)# remote subnet 10.1.0.0/16
(config—ike—conn—t1)# exit
(config)# crypto ike enable conn t1
```

При данных настройках любой трафик из сети 10.1.0.0/16 в 10.2.0.0/16 (и обратно) вызовет инициацию туннеля (опция «auto route»).

Следует отметить, что при такой конфигурации в туннель будет попадать трафик между подсетями, то есть трафик 10.1.0.0/16 □ 10.2.0.0/16. Но следующие типы трафиков в туннель попадать **не** будут:

- 10.1.0.0/16 □ 11.2.0.1 ;
- 11.1.0.1 □ 10.2.0.0/16 ;
- 11.1.0.1 □ 11.2.0.1 .

Если необходимо, чтобы данный трафик тоже попадал в туннель, то нужно настроить дополнительные соединения соответственно типов «подсеть-точка», «точка-подсеть», «точка-точка». Это можно сделать, например, с помощью копирования соединений.

Дополнительная настройка Dionis1:

```
(config)# crypto ike copy conn t1 to t1—net—host
(config)# crypto ike conn t1—net—host
(config—ike—conn—t1—net—host)# no remote subnet
(config—ike—conn—t1—net—host)# crypto ike enable conn t1—net—host
(config)# crypto ike copy conn t1 to t1—host—net
(config)# crypto ike conn t1—host—net
(config—ike—conn—t1—host—net)# no local subnet
(config—ike—conn—t1—host—net)# crypto ike enable conn t1—host—net
(config)# crypto ike copy conn t1 to t1—host—host
```

```
(config—ike—conn—t1—host—host)# no local subnet
(config—ike—conn—t1—host—host)# no remote subnet
(config—ike—conn—t1—host—host)# crypto ike enable conn t1—host—host
(config)# _
```

Дополнительная настройка Dionis2:

```
(config)# crypto ike copy conn t1 to t1—net—host
(config)# crypto ike conn t1—net—host
(config—ike—conn—t1—net—host)# no local subnet
(config—ike—conn—t1—net—host)# crypto ike enable conn t1—net—host
(config)# crypto ike copy conn t1 to t1—host—net
(config)# crypto ike conn t1—host—net
(config—ike—conn—t1—host—net)# no remote subnet
(config—ike—conn—t1—host—net)# crypto ike enable conn t1—host—net
(config)# crypto ike copy conn t1 to t1—host—host
(config—ike—conn—t1—host—host)# no local subnet
(config—ike—conn—t1—host—host)# no remote subnet
(config—ike—conn—t1—host—host)# crypto ike enable conn t1—host—host
(config)# _
```

В данном примере создаётся ещё 3 соединения путём копирования. Чтобы эти соединения стали типа «подсеть-точка», «точка-подсеть» и «точка-точка», из них удаляются соответствующие опции «local/remote subnet» с помощью команд «no».

59.7.13 Динамическое разрешение IP-адресов

Бывают ситуации, когда IP-адреса концов туннеля (задаваемые опциями «local ip» и «remote ip») заранее неизвестны или могут меняться. Например, адрес локального конца туннеля назначается через протокол DHCP, а адрес удалённого конца неизвестен, но известно доменное имя оппонента. В этих случаях опции «local ip» и/или «remote ip» можно задавать в следующем виде:

```
(config)# crypto ike conn <имя_соединения>
(config—ike—conn—<имя>)# local ip from <сетевой_интерфейс>
(config—ike—conn—<имя>)# remote ip <доменное_имя>
```

В этом случае при активации командой «crypto ike enable conn» соединение перейдёт в состояние «unresolved» до тех пор, пока не выяснятся конкретные IP-адреса для «local/remote ip». Попытка разрешить адреса будет повторяться каждые 10 секунд. При успешном разрешении адресов соединение перейдёт в соответствующее состояние («listen»/«routed»/«online») в зависимости от настройки «auto».

Следует отметить, что после удачного разрешения адресов нового разрешения производиться не будет. И если IP-адреса изменятся, то соединение прервётся и будет оставаться в состояниях «listen» или «pending». Для повторного разрешения адресов необходимо выключить и заново включить соединение с помощью команд режима конфигурации:

```
(config)# crypto ike disable conn <имя>
(config)# crypto ike enable conn <имя>
```

См. также «remote ip *» в разделе «Соединения «клиенты-сервер».

59.7.14 Соединения “клиенты-сервер”

59.7.14.1 Шаблонные соединения

В некоторых настройках соединения допустимо использование шаблонов (символ «*»). В этом случае узел может быть только ответчиком (сервером), и самостоятельная инициация соединения становится невозможной.

При использовании шаблонов становится возможным установление соединений типа «клиент-сервер» сразу со множеством клиентов. В этом случае порождается несколько соединений из одного, и команда просмотра состояния соединения выдаст информацию о порождённых соединениях, помеченных индексом. Например:

```
# show crypto ike conn t1
t1  [1]  online    CN=Иванов Иван Иванович,O=Хорошая организация,C=RU
t1  [2]  online    CN=Петров Пётр Петрович,O=Хорошая организация,C=RU
t1  [3]  pending2
```

Существует возможность выводить состояние/статистику одного порождённого соединения с помощью команды:

```
# show crypto ike conn t1 <номер> [verbose|stats]
```

Команда «crypto ike close conn» закрывает все порождённые соединения:

```
# crypto ike close conn t1
# show crypto ike conn t1
t1        listen
```

Существует возможность выборочно закрывать конкретные экземпляры шаблонных соединений по номеру. Чтобы закрыть соединение, например, только от Петрова Петра Петровича, необходимо выполнить команду:

```
# crypto ike close conn t4 2
# show crypto ike conn t1
t1  [1]  online    CN=Иванов Иван Иванович,O=Хорошая организация,C=RU
t1  [3]  pending2
```

59.7.14.2 Шаблон IP-адреса клиента

Следующая опция позволяет устанавливать туннель с любого IP-адреса:

```
(config—ike—conn—<имя>)# remote ip *
```

59.7.14.3 Шаблон X500-имени клиента

Существует возможность задавать шаблон в X500-имени клиентов. Например:

```
(config—ike—conn—<имя>)# remote id ”CN=*, O=Хорошая организация, C=RU”
```

В данном примере любой клиент из «Хорошей организации» может инициировать соединение.

Допускается задавать несколько «*», например:

```
(config—ike—conn—<имя>)# remote id ”CN=*, O=*, C=RU”
```

Примечание: Символ «*» распознаётся как шаблон, если больше нет никаких других символов после «=» в паре «параметр=значение». То есть следующая конструкция воспримется как конкретное X500-имя, содержащее «*», но не как шаблон:

```
CN=Клиент*, O=Хорошая организация, C=RU
```

Также допускается использовать глобальный шаблон в «remote id» (но только в сочетании с «remote ip *»):

```
(config—ike—conn—<имя>)# remote id *
(config—ike—conn—<имя>)# remote ip *
```

В данном примере разрешается установление соединений с любых IP-адресов для любых клиентов, чьи сертификаты выпущены доверяемыми УЦ.

59.7.14.4 “Уникальные” клиенты

По умолчанию в службе IKE действует глобальная настройка «unique ids»:

```
(config)# crypto ike config
(config—ike)# unique ids
```

При включённой настройке «unique ids» происходит автоматическое закрытие старых соединений, если оппонент с тем же самым X500-именем инициировал новое соединение с другого IP-адреса. То есть соблюдается правило «уникальности» оппонентов: один оппонент может одновременно соединяться только с одного IP. Такая настройка является полезной для мобильных клиентов, чтобы избежать множества «висячих» соединений.

Если требуется разрешить подключаться одному субъекту одновременно с нескольких IP, то данную настройку можно отключить:

```
(config—ike)# no unique ids
```

59.7.14.5 Шаблон клиентских правил отбора

Иногда бывают ситуации, когда серверу заранее неизвестны точный адрес и точная маска защищаемой подсети клиента. В этом случае можно указать неточный адрес удалённой подсети с меньшей маской. Это делается с помощью опции режима конфигурации соединения:

```
(config—ike—conn—<имя>)# remote subnet within <ip/mask>
```

В этом случае удалённая подсеть будет «уточнена» при установлении соединения от конкретного клиента. Также эта опция необходима, когда требуется принимать соединения от нескольких клиентов, защищающих свои внутренние подсети.

Пример:

Допустим, нескольким филиалам требуется подключаться к центральному серверу. Точное количество филиалов неизвестно заранее. Есть договорённость, что внутренние сети филиалов должны иметь префикс 10.0.0.0/8. Например, подсеть филиала 1 - 10.1.0.0/16, подсеть филиала 2 - 10.2.0.0/16 и т.д. Также есть договорённость, что X500-имена криптомаршрутизаторов филиалов должны соответствовать шаблону «CN=Шлюз, OU=<название_филиала>, O=Хорошая организация, C=RU».

Настройка центрального сервера:

```
(config)# crypto ike conn filialy
(config—ike—conn—filialy)# local ip 11.1.0.1
(config—ike—conn—filialy)# local subnet 11.2.0.0/16
(config—ike—conn—filialy)# local cert mycert.cer
(config—ike—conn—filialy)# remote ip *
(config—ike—conn—filialy)# remote id ”CN=Шлюз, OU=*, O=Хорошая организация, C=RU”.
(config—ike—conn—filialy)# remote subnet within 10.0.0.0/8
```

59.7.14.6 Мобильные клиенты с внутренним IP-адресом

Для мобильных клиентов может быть полезна возможность создания виртуального внутреннего адреса при соединении с сервером по IPsec. В этом случае клиент не потеряет связь с Интернетом, потому что трафик, не попадающий в туннель, будет отправляться с основного IP-адреса клиента. А трафик, предназначенный для туннеля, - с виртуального. Если клиентом является DionisNX, то виртуальный адрес назначается интерфейсу как вторичный IP, и инсталлируются специальные правила маршрутизации, отправляющие трафик, предназначенный для туннеля, через виртуальный адрес.

Виртуальные адреса управляются с помощью опций «local source ip» на стороне клиента, и «remote source ip» на стороне сервера.

59.7.14.7 Пример 1. Клиент предлагает серверу собственный виртуальный адрес

Настройка клиента:

```
crypto ike conn office—vpn
 local ip 192.168.1.1
 remote ip 192.168.2.1
 local cert client.cer
 remote id ”CN=Сервер доступа, O=Хорошая организация, C=RU”
 local source ip 10.0.0.1 next—hop 192.168.1.100
 remote subnet 10.2.0.0/16
```

Настройка сервера:

```
crypto ike conn office—vpn
 local ip 192.168.2.1
```

```
 remote ip *
 local cert server.cer
 remote id ”CN=*, O=Хорошая организация, C=RU”
 local subnet 10.2.0.0/16
 remote subnet within 10.0.0.0/16
```

В данном примере клиент предлагает серверу свой виртуальный адрес 10.0.0.1. Сервер примет предложение, потому что виртуальный адрес клиента попадает в допустимый диапазон, установленный опцией «remote subnet within 10.0.0.0/16».

При использовании опции «local source ip» необходимо указывать адрес непосредственного следующего маршрутизатора (next hop). Можно указать либо конкретный IP-адрес, либо опцию «default-route», если в системе существует маршрут по умолчанию, и он подходит для данного IPsec-траффика.

59.7.14.8 Пример 2. Сервер назначает клиенту виртуальный адрес (режим ModeConfig)

Настройка клиента:

```
crypto ike conn office—vpn
 local ip 192.168.1.1
 remote ip 192.168.2.1
 local cert client.cer
 remote id ”CN=Сервер доступа, O=Хорошая организация, C=RU”
 local source ip modeconfig next—hop default—route
 remote subnet 10.2.0.0/16
```

Настройка сервера:

```
crypto ike conn office—vpn
 local ip 192.168.2.1
 remote ip *
 local cert server.cer
 remote id ”CN=Иванов Иван Иванович, O=Хорошая организация, C=RU”
 local subnet 10.2.0.0/16
 remote source ip 10.0.0.1
```

В данном примере у клиента действует настройка «local source ip modeconfig», которая означает, что клиент ожидает назначения виртуального адреса сервером (с помощью фазы ModeConfig). Опция сервера «remote source ip 10.0.0.1» предписывает назначить клиенту виртуальный адрес 10.0.0.1.

59.7.14.9 Пример 3. Сервер назначает клиентам виртуальные адреса из пула (режим ModeConfig)

Также реализована возможность назначения разных адресов разным клиентам из пула.

Настройка сервера:

```
crypto ike conn office—vpn
 local ip 192.168.2.1
 remote ip *
```

```
local cert server.cer
remote id ”CN=*, O=Хорошая организация, C=RU”
local subnet 10.2.0.0/16
remote source ip 10.0.0.0/24
```

В данном примере опция «remote source ip 10.0.0.0/24» создаёт пул адресов от 10.0.0.1 до 10.0.0.254. Каждому новому подключающемуся клиенту будет назначаться свой виртуальный адрес.

Просмотреть состояние пула(ов) можно с помощью команды режима enable:

```
# show crypto ike pool [<имя_соединения_или_именованного_пула>]
```

Удалить опции «local/remote source ip» из конфигурации соединения можно с помощью соответствующих команд «no».

59.7.14.10 Сопоставление выдаваемых виртуальных адресов с конкретными субъектами

Существует возможность назначить правила сопоставления выдаваемых виртуальных адресов (групп адресов) с конкретными субъектами (группами субъектов). Данная возможность может использоваться для идентификации трафика от конкретного мобильного клиента внутри защищаемой сети, а также для разграничения доступа различных мобильных клиентов к различным ресурсам защищаемой сети.

Для назначения правил сопоставления адресов с субъектами необходимо создать именованный пул адресов. Следующая команда используется для создания/редактирования именованного пула:

```
(config)# crypto ike pool <имя_пула>
(cfg—ike—pool—<имя>)# _
```

Удалить именованный пул можно с помощью команды:

```
(config)# no crypto ike pool <имя_пула>
```

В режиме редактирования пула необходимо задать опцию “pool” для определения подсети пула.

Далее в данном режиме можно задать одно или несколько правил сопоставления “map”. Правила нумеруются и имеют приоритет применения (1 - старший). При вводе команды “map” без числового префикса правило добавляется в конец списка. При вводе команды “map” с числовым префиксом вводимое правило вставляется на указанную позицию, старые правила, начиная с данной позиции, “сдвигаются” вниз и перенумеровываются.

Удалить правило “map” можно с помощью команды “no ”. Удалить все правила (кроме опции “pool”) можно с помощью команды “no all”.

Далее в соответствующем(их) соединении(ях) необходимо указать использование данного именованного пула с помощью опции “remote source ip”.

Пример:

```
(config)# crypto ike pool p1
(cfg—ike—pool—p1)# pool 10.10.10.0/24
(cfg—ike—pool—p1)# 1 map 10.10.10.100 ”CN=Начальник хорошего отдела,O=Хорошая
        организация,C=RU”
```

```
(cfg—ike—pool—p1)# 2 map 10.10.10.128/25 ”CN=*,OU=Хороший отдел,O=Хорошая
        организация,C=RU”
(cfg—ike—pool—p1)# exit

(config)# crypto ike conn office—vpn
(config—ike—conn—office—vpn)# remote source ip p1
```

В данном примере субъекту “Начальник хорошего отдела” будет всегда выдаваться адрес 10.10.10.100. Субъектам, чьи имена совпадают с шаблоном “CN=*,OU=Хороший отдел,O=Хорошая организация,C=RU”, будут выдаваться адреса в диапазоне от 10.10.10.128 до 10.10.10.254. Остальным субъектам будут выдаваться адреса в диапазоне от 10.10.10.1 до 10.10.10.99 и от 10.10.10.101 до 10.10.10.127.

Действуют следующие правила/ограничения:

- Если пул используется в enabled-соединении, и была изменена опция “pool”, то необходимо перезагрузить данное соединение (иначе служба откажет в выдаче новых виртуальных адресов).
- Изменение правил “map” не требует перезагрузки соединения.
- Если на момент выдачи очередного виртуального адреса в пуле будет обнаружен конфликт правил с опцией “pool”, служба откажет в выдаче адреса.
- Для одного субъекта (одной группы субъектов) можно указать только одно правило.
- Если подпул группы субъектов полностью заполнен (все субъекты “online”), то новому субъекту из данной группы будет выдан адрес из “свободной” области. О данном событии будет сделана запись в журнале.
- Если все адреса “свободной” области заняты (все “online”), то новому субъекту (не попадающему в группы) будет отказано в выдаче адреса.
- Если имя субъекта совпало с шаблоном более приоритетного правила, то остальные (менее приоритетные) правила не рассматриваются. Рекомендуется более частные правила делать более приоритетными, а более общие - менее приоритетными.

59.7.14.11 Сопоставление серверных подсетей с конкретными субъектами

Существует возможность в правилах пула (см. команду ‘crypto ike pool’ выше) помимо выдаваемых виртуальных адресов также указывать серверные подсети и ассоциировать их с конкретными субъектами. В зависимости от того, какой субъект подключился к данному соединению, согласно правилам выбирается соответствующее значение серверной подсети и назначается серверной стороне (как ‘local subnet’). Клиент может выполнить запрос к серверу (на фазе ModeConfig) и получить значение серверной подсети.

Для того, чтобы серверу назначалась локальная подсеть согласно правилам пула, необходимо (на стороне сервера) указать опцию:

```
(config)# crypto ike conn t1
(config—t1)# local subnet from pool
```

При этом опция “remote source ip” должна указывать на соответствующий именованный пул.

Для того, чтобы мобильный клиент запрашивал у сервера информацию о его защищаемой подсети, необходимо (на стороне клиента) указать опцию:

```
(config)# crypto ike conn t1
(config—t1)# remote subnet modeconfig
```

При этом также должна быть указана опция ‘local source ip modeconfig’.

На стороне сервера в правилах пула (см. команду ‘crypto ike pool’) должны быть указаны серверные подсети для соответствующих субъектов (3-й необязательный параметр команды ‘map’), а также желательно, чтобы была указанная опция ‘local subnet’, которая обозначает “подсеть по умолчанию” и используется в том случае, если для данного субъекта не была указана конкретная подсеть.

Пример:

```
(config)# crypto ike pool p1
(cfg—ike—pool—p1)# pool 10.10.10.0/24
(cfg—ike—pool—p1)# local subnet 10.10.20.0/24
(cfg—ike—pool—p1)# 1 map 10.10.10.100 ”CN=Начальник хорошего отдела,O=Хорошая
        организация,C=RU” 10.10.21.0/24
(cfg—ike—pool—p1)# 2 map 10.10.10.128/25 ”CN=*,OU=Хороший отдел,O=Хорошая
        организация,C=RU” 10.10.22.0/24
(cfg—ike—pool—p1)# exit

(config)# crypto ike conn office—vpn
(config—ike—conn—office—vpn)# remote source ip p1
(config—ike—conn—office—vpn)# local subnet from pool
```

В данном примере субъект “Начальник хорошего отдела” будет иметь доступ к подсети 10.10.21.0/24, субъекты из “Хорошего отдела” - к подсети 10.10.22.0/24, а все остальные субъекты - к подсети 10.10.20.0/24.

Чтобы очистить опцию ‘local subnet’ в правилах пула, можно выполнить команду:

```
(config)# crypto ike pool <имя_пула>
(cfg—ike—pool—<имя_пула>)# no local subnet
```

В случае, если для субъекта отсутствует правило назначения серверной подсети, а также отсутствует опция ‘local subnet’, фаза ModeConfig не будет завершена, и клиенту будет отправлено уведомление “ADDRESS_NOTIFICATION”.

59.7.14.12 Уведомление мобильного клиента о внутреннем DNS-сервере

В примерах 2 и 3 сервер может передавать мобильным клиентам адреса внутренних DNS-серверов. Для этого на стороне сервера необходимо указать опцию:

```
crypto ike conn office—vpn
 modeconfig dns <DNS_IP_1> [<DNS_IP_2>]
```

ПРИМЕЧАНИЕ: Возможность передачи внутренних адресов DNS предназначена для мобильных Windows-клиентов DiSEC. Если мобильным клиентом является DionisNX, то он игнорирует адреса DNS, присланные сервером.

59.7.15 Соединения через NAT

Если предполагается, что IPsec-туннель будет проходить через маршрутизаторы, выполняющие трансляцию адресов (NAT), то на обоих концах туннеля необходимо включить поддержку механизма NAT Traversal (RFC3947). Это делается с помощью опции службы IKE:

```
(config)# crypto ike config
(config—ike)# nat traversal
```

По умолчанию NAT Traversal отключён, что равносильно опции:

```
(config—ike)# no nat traversal
```

Как было описано выше, если возникает ситуация «IPsec через NAT», IKE-обмен происходит следующим образом:

- Первая фаза начинается стандартно - через UDP/500;
- После обмена 2-м и 3-м пакетами оппоненты определяют наличие NAT;
- Весь последующий IKE-обмен ведётся через UDP/4500;
- Протокол ESP инкапсулируется также в UDP/4500 (RFC3948).

При использовании «IPsec через NAT» ACL-фильтры должны быть настроены таким образом, чтобы пропускать трафик UDP/4500. При этом следует учитывать, что порт оппонента может быть любым (из-за NAT).

Для поддержания соединения через NAT оппоненты периодически посылают друг другу специальные пакеты «Keep Alive». Интервал по умолчанию - 20 с. Если требуется изменить это значение, то это можно сделать с помощью опции:

```
(config)# crypto ike config
(config—ike)# nat keep—alive interval <secs>
```

«IPsec через NAT» возможен только при выполнении следующих условий:

- Используется туннельный режим ESP. Транспортный режим отключён из-за проблем с безопасностью;
- Если оппонент находится за NAT, то опция «remote ip» для данного узла **должна** быть «*»;
- Если оппонент находится за NAT, то на данном узле **необходимо** наличие одной из опций: «remote subnet [within]» (для «подсеть–клиент==NAT==сервер–…»), «remote source ip» (для «клиент==NAT==сервер–…»);
- Используется аутентификация по сертификатам (не PSK).

Примечание: Если предполагается использование NAT Traversal совместно с режимом ModeConfig (выделение адресов клиентам из пула), то также настоятельно рекомендуется использование механизма Dead Peer Detection (см. раздел «Продление и закрытие туннелей. Таймеры») со стороны сервера для избежания истощения пула в случае обрывов связи.

59.7.16 Принудительная инкапсуляция ESP в UDP

Иногда бывает необходимо инкапсулировать ESP-трафик в UDP, даже если NAT не присутствует. Для этого необходимо на стороне инициатора в настройках соединения указать опцию:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# udp encap force
```

Также необходимо, чтобы на стороне ответчика и инициатора была включена поддержка NAT Traversal:

```
(config)# crypto ike config
(config—ike)# nat traversal
```

Примечание: Для инициации принудительной инкапсуляции в UDP достаточно, чтобы опция «udp encap force» была указана хотя бы у одного из оппонентов (желательно, у инициатора). Если она указана на стороне ответчика, то во избежание ложных срабатываний/несрабатываний данная опция должна быть также указана для всех остальных соединений **между данными оппонентами** (если таких соединений несколько).

Отключение принудительной инкапсуляции осуществляется командой:

```
(config—ike—conn—t1)# no udp encap force
```

ВАЖНО: В текущей реализации Dionis-NX опция принудительной инкапсуляции в UDP работает корректно только для соединений типа «клиент-сервер». То есть на стороне сервера обязательно должна присутствовать опция «remote ip *».

59.7.17 Отбор трафика по протоколу и порту

В предыдущих примерах в туннель попадал трафик любых протоколов (кроме IKE - UDP/500/4500).

Если необходимо направлять в туннель трафик определённого протокола, то следует указать опции «local protoport» и «remote protoport» на обоих сторонах. Например:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# local protoport ospf
(config—ike—conn—t1)# remote protoport ospf
```

Протоколы для «local» и «remote» должны совпадать. Также протоколы можно указывать по номеру:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# local protoport 89
(config—ike—conn—t1)# remote protoport 89
```

Для протоколов TCP и UDP можно указывать порты. В этом случае в туннель будет попадать трафик определённого порта. Порты для «local/remote» могут не совпадать, но настройки на концах должны быть симметричны. (См. раздел «Базовые понятия протокола IKE, Фаза 2»).

Пример (настройка web-доступа через IPsec):

Сервер:

```
local protoport tcp/80
remote protoport tcp
```

Клиенты:

```
local protoport tcp
remote protoport tcp/80
```

В данном примере в IPsec-туннель будет попадать только web-трафик.

Удалить опции «local/remote protoport» из конфигурации соединения можно с помощью соответствующих команд «no».

59.7.18 Исключение трафика из туннеля

Иногда возникает необходимость исключить часть трафика из IPsec-туннеля.

Для этого можно воспользоваться командой «exceptions» в режиме конфигурации соединения.

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# exceptions
(config—ike—conn—t1—exc)# _
```

Данная команда вводит консоль в режим редактирования правил исключений.

Добавить/вставить правило можно с помощью команды:

```
[<n>] deny [<протокол>] [src <локальная_подсеть>] [dst <удалённая_подсеть>] [sport
        <локальный_порт> [<локальный_порт2>]] [dport <удалённый_порт> [<удалённый_порт2>]]
```

Команда «deny» исключает определённый вид трафика из туннеля. Параметры команды:

<n> - номер позиции, куда должно быть вставлено правило. Если правило с таким номером уже существует, то оно сдвигается вниз. Если <n> не указан - правило добавляется в конец списка;

<протокол> - номер или название IP-протокола. Если указан, исключается трафик только данного протокола;

<локальная_подсеть> - IP-адрес/маска диапазона локальной подсети. Если адрес отправителя исходящего трафика (адрес получателя входящего трафика) попадает в данный диапазон, трафик исключается из туннеля;

<удалённая_подсеть> - IP-адрес/маска диапазона удалённой подсети. Если адрес получателя исходящего трафика (адрес отправителя входящего трафика) попадает в данный диапазон, трафик исключается из туннеля;

<локальный_порт> - (только для TCP/UDP). Исключается только трафик указанного порта (для исходящего - порт отправителя, для входящего - порт получателя);

<локальный_порт2> - если указан, то исключается трафик для диапазона портов - от <локальный_порт> до <локальный_порт2>;

<удалённый_порт> - (только для TCP/UDP). Исключается только трафик указанного порта (для исходящего - порт получателя, для входящего - порт отправителя);

<удалённый_порт2> - если указан, то исключается трафик для диапазона портов - от <удалённый_порт> до <удалённый_порт2>.

Добавляемому правилу присваивается номер. Просмотреть номера правил можно с помощью команды «do show» из режима редактирования исключений:

```
(config—ike—conn—t1—exc)# do show
1 deny tcp src 192.168.1.0/24 dst 192.168.2.1/32 dport 80
2 deny src 192.168.1.24/29
(config—ike—conn—t1—exc)# _
```

Удалить правило можно с помощью команды «no <номер>». Оставшиеся правила перенумеровываются. Удалить все правила можно с помощью команды «no all».

Чтобы выключить режим исключения трафика из туннеля и удалить все исключающие правила, нужно выполнить команду режима конфигурации соединения:

```
(config—ike—conn—имя)# no exceptions
```

Примечание: Если использование исключающих правил не требуется, рекомендуется включить опцию "native-policy" (см. ниже) для более оптимальной обработки трафика.

59.7.19 Упрощённый механизм отбора трафика. Ситуация "–NAT–IPsec=="

При настройке сложных схем преобразования трафика совместно с IPsec (например, если требуется изменение IP-адресов (NAT) открытого трафика перед отправкой в IPsec-туннель) из-за особенности реализации внутренних механизмов прохождения трафика по TCP/IP-стеку может наблюдаться неработоспособность IPsec, выражающаяся в непопадании NAT-преобразованного трафика в IPsec-туннель (несмотря на то, что он удовлетворяет правилам отбора).

В этом случае необходимо отключить расширенные возможности отбора трафика с помощью опции настройки соединения:

```
(config—ike—conn—t1)# native—policy
```

Примечание: При включённой опции "native-policy" становится невозможным исключение трафика из туннеля с помощью опции "exceptions". Если использование исключающих правил не требуется, рекомендуется включить опцию "native-policy" для более оптимальной обработки трафика.

ВНИМАНИЕ: При применении NAT к открытому трафику перед отправкой в IPsec-туннель можно использовать только настройку "ip nat-group". Настройка "ip nat-group-xfrm" НЕ ПРИМЕНИМА для IPsec. При этом опция "native-policy" должна быть включена.

Отключить опцию "native-policy" можно командой:

```
(config—ike—conn—t1)# no native—policy
```

59.7.20 Настройка криптопараметров

Как было сказано выше, на IKE-фазе 1 происходит согласование криптографических параметров, определяющих способ шифрования и проверки подлинности пакетов IKE. А на фазе 2 происходит согласование криптографических параметров протокола ESP.

59.7.20.1 Значения по умолчанию

По умолчанию действуют следующие значения:

Криптопараметры IKE:

Предлагается 1 набор криптопараметров;

Шифрование и имитовставка: Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set B;

Выработка общего секрета: Алгоритм ГОСТ Р 34.10-2001, режим VKO, параметры CryptoPro Set XchB;

Политика выбора набора криптопараметров: строгая;

Режим Perfect Forward Secrecy (PFS): предлагать PFS, принимать любой.

Криптопараметры ESP:

Предлагается 1 набор криптопараметров;

Шифрование и имитовставка: Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set B;

Политика выбора набора криптопараметров: строгая.

59.7.20.2 Настройка и согласование криптопараметров. “Строгость” политики согласования

Согласование криптопараметров происходит следующим образом:

1. Инициатор соединения предлагает один или несколько наборов криптопараметров (сортированных по приоритету);
2. Ответчик анализирует предложения (proposals), выбирает первое подходящее и уведомляет инициатора о выборе;
3. Если ничего не выбрано, соединение отвергается.

В настройках соединения можно явно задать предлагаемые наборы и их последовательность в предложениях с помощью команд «ph1 transforms» (криптопараметры для IKE) и «ph2 transforms» (криптопараметры для ESP). Для ответчика различаются две политики выбора набора предлагаемых криптопараметров: строгая и нестрогая. При нестрогой политике выбирается первый попавшийся набор, который поддерживается системой IPsec ответчика (то есть любой, если оба оппонента - узлы Dionis-NX). При строгой политике выбирается первый попавшийся набор, который присутствует в списке наборов, заданном командой «ph1/ph2 transforms».

Чтобы войти в режим редактирования списка предлагаемых наборов криптопараметров IKE, нужно ввести команду «ph1 transforms» в режиме конфигурации соединения:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# ph1 transforms
(config—ike—conn—t1—ph1)# _
```

Если необходимо выключить строгую политику выбора, то следует указать опцию:

```
(config—ike—conn—t1—ph1)# no strict
```

Строгая политика включается противоположной опцией:

```
(config—ike—conn—t1—ph1)# strict
```

Для добавления набора криптопараметров IKE выполняется команда «add». Очередной набор всегда добавляется в конец списка.

59.7.20.3 Криптопараметры фазы 1

Формат команды «add» для «ph1 transforms»:

```
add <алгоритм_шифрования> <алгоритм_выработки_общего_секрета>
```

Возможные алгоритмы шифрования:

gost89-a	Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set A
gost89-b	Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set B
gost89-c	Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set C
gost89-d	Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set D
gost89-z	Алгоритм ГОСТ 28147-89, режим GOST-CFB-IMIT, узел замены CryptoPro Set Z
magma	Алгоритм ГОСТ Р 34.12-2015, режим MAGMA-CFB, MAGMA-OMAC-ACPKM

Возможные алгоритмы выработки общего секрета:

gost2001-vko-a	Алгоритм ГОСТ Р 34.10-2001, режим VKO, параметры CryptoPro Set XchA
gost2001-vko-b	Алгоритм ГОСТ Р 34.10-2001, режим VKO, параметры CryptoPro Set XchB
gost2012-256-vko-a	Алгоритм ГОСТ Р 34.10-2012 (256 бит), режим VKO, параметры CryptoPro Set XchA
gost2012-256-vko-b	Алгоритм ГОСТ Р 34.10-2012 (256 бит), режим VKO, параметры CryptoPro Set XchB
gost2012-512-vko-a	Алгоритм ГОСТ Р 34.10-2012 (512 бит), режим VKO (с выходом 256 бит), параметры TC26A
gost2012-512-vko-b	Алгоритм ГОСТ Р 34.10-2012 (512 бит), режим VKO (с выходом 256 бит), параметры TC26B

Пример:

```
(config—ike—conn—t1—ph1)# add gost89—a gost2001—vko—b
(config—ike—conn—t1—ph1)# add gost89—d gost2001—vko—a
```

Чтобы очистить список наборов и вернуть значение по умолчанию, следует выполнить команду:

```
(config—ike—conn—t1)# no ph1 transforms
```

ПРИМЕЧАНИЕ: При настройке криптопараметров фазы 1 следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

ПРИМЕЧАНИЕ: На фазе 1 также согласуется функция хэширования (ГОСТ Р 34.11-94, ГОСТ Р 34.11-2012 512 бит). Согласуемый тип функции зависит от алгоритма открытого ключа сертификата, указанного в настройке “local cert”. Если алгоритм ключа - ГОСТ Р 34.10-2001, то инициатором предлагается (ответчиком допускается) только одна хэш-функция - ГОСТ Р 34.11-94. Если алгоритм ключа - ГОСТ Р 34.10-2012, то инициатором предлагаются (ответчиком допускаются) обе хэш-функции (функция ГОСТ Р 34.11-2012 (512 бит) имеет больший приоритет). В этом случае инициатор дублирует все пропоузалы. В режиме PSK всегда согласуются оба типа хэша. Также вне зависимости от опции “strict” алгоритмы хэш-функций сравниваются всегда по строгой политике.

59.7.20.4 Криптопараметры фазы 2

Наборы криптопараметров ESP и политика выбора настраиваются аналогично с помощью команды «ph2 transforms»:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# ph2 transforms
(config—ike—conn—t1—ph2)# add <алгоритм_шифрования> [esn]
```

Возможные алгоритмы шифрования для ESP:

gost89-4m-imit-a	Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set A
gost89-4m-imit-b	Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set B
gost89-4m-imit-c	Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set C
gost89-4m-imit-d	Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set D
gost89-4m-imit-z	Алгоритм ГОСТ 28147-89, режим GOST-4M-IMIT, узел замены CryptoPro Set Z
gost89-1k-imit-a	Алгоритм ГОСТ 28147-89, режим GOST-1K-IMIT, узел замены CryptoPro Set A
gost89-1k-imit-b	Алгоритм ГОСТ 28147-89, режим GOST-1K-IMIT, узел замены CryptoPro Set B
gost89-1k-imit-c	Алгоритм ГОСТ 28147-89, режим GOST-1K-IMIT, узел замены CryptoPro Set C
gost89-1k-imit-d	Алгоритм ГОСТ 28147-89, режим GOST-1K-IMIT, узел замены CryptoPro Set D
gost89-1k-imit-z	Алгоритм ГОСТ 28147-89, режим GOST-1K-IMIT, узел замены CryptoPro Set Z
magma-4m-imit	Алгоритм ГОСТ Р 34.12-2015, режим MAGMA-MGM

Если указан параметр «esn», то для данного алгоритма будет согласовываться режим 64-разрядного счётчика пакетов (Extended Sequence Numbers, RFC4304). Также возможно ввести оба варианта для одного алгоритма - с ESN и без. Например:

```
(config—ike—conn—t1)# ph2 transforms
(config—ike—conn—t1—ph2)# add gost89—1k—imit—b esn
(config—ike—conn—t1—ph2)# add gost89—1k—imit—b
```

59.7.20.5 Perfect Forward Secrecy

Для выработки более криптостойкого ключевого материала при использовании протокола ESP на фазе 2 инициатор и ответчик могут обменяться дополнительными временными открытыми ключами (KEi, KEr) и сформировать дополнительный общий секрет. Данный режим называется режимом совершенной прямой секретности (Perfect Forward Secrecy) и может настраиваться командой «pfs mode» в режиме конфигурации соединения.

По умолчанию действует режим:

```
(config—ike—conn—<имя>)# pfs mode propose
```

Если данный режим включён на узле, выступающем в роли инициатора, то он передаёт KEi на фазе 2, инициируя тем самым режим PFS. Если данный режим включён на ответчике, то последний поддерживает оба режима (PFS и Non-PFS) и делает выбор в зависимости от того, прислал ли инициатор KEi или нет.

В режиме

```
(config—ike—conn—<имя>)# pfs mode off
```

инициатор не передаёт KEi, инициируя тем самым режим Non-PFS. Ответчик может принимать оба режима.

Режим Non-PFS можно запретить опцией:

```
(config—ike—conn—<имя>)# pfs mode force
```

В этом случае и инициатор, и ответчик будут работать только в режиме PFS, причём опция «pfs mode force» должна быть включена у обоих оппонентов для успешного согласования фазы 1. (Согласуется атрибут «PFS Control - Disable Non-PFS»).

В таблице представлено поведение при согласовании PFS в зависимости от опции «pfs mode» на инициаторе (I) и ответчике (R):

I\R	off	propose	force
off	Non-PFS	Non-PFS	Отказ
propose	PFS	PFS	Отказ
force	Отказ	Отказ	PFS

ПРИМЕЧАНИЕ: При настройке параметра «pfs mode» следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

По умолчанию при согласовании дополнительного общего секрета используются криптопараметры фазы 1. Если требуется их изменить, то это можно сделать командой:

```
(config—ike—conn—<имя>)# pfs group <алгоритм_выработки_общего_секрета>
```

Если у ответчика действует режимы «ph2 transforms; strict» и «pfs mode propose/force», а также указана «pfs group», то для успешного согласования параметров инициатор должен предложить именно эту группу VKO. Во всех остальных случаях ответчик примет любую (им поддерживаемую) группу VKO, предложенную инициатором, а настройка «pfs group» будет иметь смысл только на стороне инициатора.

59.7.20.6 Максимальное количество фаз 2 из одной фазы 1

Из одной фазы 1 может быть порождено несколько фаз 2. Существует криптографическое ограничение на количество таких фаз 2. (В данном случае «фазой 2» также считается посылка каждого уведомительного сообщения). При режиме «pfs mode force» - максимальное количество фаз 2 - 65536. При режиме «pfs mode off/propose» - 16384. Если необходимо ещё уменьшить данный предел, то это можно сделать с помощью опции:

```
(config—ike—conn—<имя>)# ph2 max <n>
```

Согласование значений «ph2 max» между оппонентами происходит по следующим правилам:

- Если инициатор не пересылает ответчику атрибут «Max-messages», ответчик использует своё значение (без уведомления инициатора);
- Если значение, пересылаемое инициатором, меньше значения ответчика, обе стороны используют значения инициатора;
- Если значение, пересылаемое инициатором, больше значения ответчика, то ответчик использует своё (меньшее) значение БЕЗ уведомления инициатора. Инициатор использует своё значение.

Рекомендуются одинаковые настройки «ph2 max» на обеих сторонах.

ПРИМЕЧАНИЕ: При настройке параметра «ph2 max» следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

59.7.20.7 Максимально допустимое количество ESP-пакетов с неправильной контрольной суммой

Как было сказано выше, если ESP-пакет не прошёл проверку подлинности (не совпала контрольная сумма ICV), то он отбрасывается. По умолчанию узел Dionis-NX готов принимать любой «правильный» пакет (с учётом replay-окна), несмотря на количество предыдущих «неправильных». Существует вероятность «brute-force»-подбора злоумышленником контрольной суммы ICV. Чтобы от этого защититься, можно согласовать с оппонентом атрибут «Max-Integrity-Fails». Он представляет собой максимальное число принятых пакетов ESP с неправильной контрольной суммой ICV (но с правильным значением IVCounter - см. стандарт IPsec от «Крипто-Про»). При превышении данного числа криптоконтекст SA будет удалён и, в случае включённой опции «rekey», инициирован новый. (Опцию «rekey» см. ниже).

Включить данное ограничение можно с помощью опции:

```
(config—ike—conn—<имя>)# esp max icv fails <n> margin <m>
```

где n - максимальное количество «неправильных» пакетов, при превышении которого криптоконтекст будет удалён; m - «отступ» заблаговременного инициирования нового туннеля (при включённой опции «rekey»). Инициация нового туннеля начнётся при достижении счётчика «неправильных» пакетов значения (n - m + 1). Старый туннель будет закрыт согласно настройкам «ph margin …» (см. ниже).

Отключить опцию можно с помощью команды:

```
(config—ike—conn—<имя>)# no esp max icv fails
```

ВНИМАНИЕ: Включение данного ограничения хотя и повышает криптостойкость, но одновременно даёт возможность злоумышленнику организовать DoS-атаку. Злоумышленнику достаточно прослушать хотя бы один «правильный» ESP-пакет, чтобы сгенерировать (n + 1) «неправильных» пакетов, вызвав тем самым удаление криптоконтекста. Слушая ESP-трафик, злоумышленник может вызвать многократную переустановку туннеля.

Согласование атрибута «Max-Integrity-Fails» происходит по следующим правилам:

- Инициатор передаёт ответчику только значение «n» (см. выше). Значение «m» не передаётся;
- Если инициатор не согласует атрибут, то инициатор не использует ограничение «неправильных» пакетов. Ответчик может использовать ограничение без уведомления инициатора;
- Если инициатор согласует атрибут, а у ответчика ограничение не настроено, ответчик будет использовать значение «n» от инициатора и m=0;
- Если значение «n» инициатора меньше значения «n» ответчика, ответчик будет использовать значение «n» от инициатора. Значение «m» ответчика будет пропорционально уменьшено;
- Если значение «n» инициатора больше значения «n» ответчика, ответчик будет использовать собственные значения «n» и «m» без уведомления инициатора.

Рекомендуются одинаковые настройки «esp max icv fails» на обеих сторонах.

59.7.21 Продление и закрытие туннелей. Таймеры

59.7.21.1 Времена жизни туннелей/фаз IKE

Установленные фазы 1 и 2 имеют определенные времена жизни. По умолчанию время жизни фазы 1 - 3 часа, фазы 2 - 1 час.

Изменить время жизни фазы 1 можно командой конфигурации соединения:

```
(config—ike—conn—<имя>)# ph1 life time <секунды>
```

ПРИМЕЧАНИЕ: При настройке параметра «ph1 life time» следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

Изменить время жизни фазы 2 можно командой:

```
(config—ike—conn—<имя>)# ph2 life time <секунды>
```

Время жизни фазы 2 эквивалентно времени жизни текущего криптоконтекста туннеля (SA).

59.7.21.2 Продление туннелей

По умолчанию для соединения действует настройка:

```
(config—ike—conn—<имя>)# rekey
```

Для этой настройки при истечении времени жизни фазы 2 будет произведена попытка установить новую фазу 2 из существующей фазы 1. Если истекает время жизни фазы 1, то будет произведена попытка установления новой фазы 1 и затем новой фазы 2. Таким образом настройка «rekey» означает продление туннеля.

Если продление туннеля не требуется, то его можно отключить опцией:

```
(config—ike—conn—<имя>)# no rekey
```

Данная опция полезна для серверов, потому что обязанность поддерживать соединения обычно возлагается на клиентов. Если клиент не продлит соединение, то оно будет закрыто.

59.7.21.3 Заблаговременное установление нового туннеля

При продлении соединения, чтобы связь по туннелю не прерывалась, осуществляется заблаговременное установление нового криптоконтекста SA еще до истечения времени жизни старого. Данный временной отступ контролируется опциями:

```
(config—ike—conn—<имя>)# ph margin time <секунды>
(config—ike—conn—<имя>)# ph margin fuzz <проценты>
```

Для инициатора временной отступ от момента истечения времени жизни старого туннеля вычисляется по формуле: margin_time * (1 + rnd(0..margin_fuzz) / 100). Привнесение случайной составляющей помогает избежать пиков трафика, если одновременно переустанавливается большое количество туннелей.

Для ответчика временной отступ вычисляется по формуле: margin_time / 2. Настройка «ph margin fuzz» в этом случае не влияет.

ПРИМЕЧАНИЕ: Опции «ph margin …» являются общими для первой и второй фазы IKE. То есть первая фаза будет также заблаговременно продлена согласно вышеприведённым формулам.

По умолчанию действуют значения:

```
ph margin time 540 (9 минут)
ph margin fuzz 100
```

ВНИМАНИЕ: Для обеспечения стабильного продления туннелей (без временных потерь связи) **рекомендуется** использовать одинаковые значения опций «ph1/2 life time», «ph margin time», «ph margin fuzz» на стороне инициатора и на стороне ответчика. Средствами IKE согласуются только «ph1/2 life time», причём только в одну сторону (инициатор □ ответчик). То есть ответчик может уменьшить своё значение «life time» согласно значению от инициатора, но инициатор никогда не уменьшит своё, так как ответчик его не уведомляет о своих значениях «life time». Значения «ph margin time/fuzz» не согласуются вообще.

ПРИМЕЧАНИЕ: При настройке параметров «ph margin …» следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

59.7.21.4 Таймеры pending-состояний. Число попыток установления соединений

Если установление (переустановление) соединения не прошло успешно, то оно задерживается в соответствующем состоянии «pending*», и инициатор пытается повторить попытку установления через некоторое время. Цикл попыток можно описать следующим образом:

1. Первая посылка пакета;
2. Неудача. Ожидание 10 секунд;
3. Вторая попытка посылки того же пакета;
4. Ожидание 20 секунд;
5. Третья попытка посылки того же пакета;
6. Ожидание 40 секунд;
7. Неудачное завершение цикла.

Предельное количество таких циклов попыток можно определить опцией:

```
(config—ike—conn—<имя>)# keying tries <n>
```

Если очередной цикл завершается неудачно, то соединение переводится в состояние «pending1», и в новом цикле начнётся инициация соединения с самого начала фазы 1. Если все n циклов завершились неудачно, соединение переводится в соответствующее состояние «routed» или «listen».

По умолчанию количество циклов не ограничено, что эквивалентно опции:

```
(config—ike—conn—<имя>)# keying tries forever
```

Для серверов рекомендуется настройка «keying tries 1».

59.7.21.5 Обнаружение “мёртвых” оппонентов (Dead Peer Detection)

Бывают ситуации, когда соединение установилось успешно, но потом по каким-то причинам пропала связь с оппонентом. Причины могут быть следующими:

- Временная потеря связи с оппонетом;
- Окончательная (долговременная) потеря связи с оппонетом;
- Аварийная перезагрузка или завершение работы системы оппонента.

Во всех этих случаях на одной стороне продолжит существование криптоконтекст SA, и соединение будет находиться в состоянии «online», хотя пакеты, направляемые в туннель, будут пропадать в «чёрную дыру». Если другая сторона утратила свой криптоконтекст окончательно (без уведомительного сообщения об удалении своего контекста), такая «чёрная дыра» будет продолжать существовать до истечении времени жизни фазы 2.

Данная ситуация может оказаться неприемлемой, и поэтому для её избежания в службе IKE реализован механизм обнаружения «умерших» оппонентов (Dead Peer Detection, DPD, RFC 3706). Суть этого механизма заключается в регулярной посылке специальных сообщений «R_U_THERE» («ты здесь?») оппоненту. Если оппонент способен их получить, и он не утратил соответствующий криптоконтекст, то он шлёт ответное подтверждающее сообщение «R_U_THERE_ACK». Если оппонент не отвечает в течение некоторого времени, то он считается «умершим», и соединение закрывается (или инициируется вновь - см. ниже).

По умолчанию механизм DPD работает в пассивном режиме. То есть данная сторона не шлёт «R_U_THERE» оппоненту, но готова ответить на запросы оппонента.

Чтобы активировать DPD, нужно ввести команду режима конфигурации соединения:

```
(config—ike—conn—<имя>)# dpd
(config—ike—conn—<имя>—dpd)# _
```

Данная команда переводит консоль в режим редактирования опций DPD.

Опция «action» определяет действие с соединением, когда обнаружено, что оппонент «умер». Возможные действия:

- close - Перевести соединение в состояние «listen»;
- route - Перевести соединение в состояние «routed»;
- initiate - Пытаться заново установить соединение (из состояния «listen», минуя состяние «routed»);
- route initiate - Перевести соединение в состояние «routed» и пытаться заново инициировать соединение.

Опция «interval» позволяет установить интервал посылки сообщений «R_U_THERE» (в секундах).

Опция «timeout» устанавливает предельное время, после которого оппонент считается «умершим», если он не ответил ни на одно сообщение «R_U_THERE».

По умолчанию действуют настройки:

```
«action» — close
«interval» — 30
«timeout» — 150
```

ПРИМЕЧАНИЕ: При настройке параметров “dpd” следует учитывать требования, описанные в разделе «Особенности настройки некоторых параметров фазы 1» (см. ниже).

59.7.22 Особенности настройки некоторых параметров фазы 1

Настройки IPsec-соединений **должны** отвечать следующему требованию:

Если существует несколько соединений с **одинаковым** набором следующих опций:

- auth;
- local ip;
- remote ip;

то **необходимо**, чтобы эти соединения также имели **одинаковые** наборы следующих опций:

- ph1 transforms;
- ph1 life time;
- ph margin time;
- ph margin fuzz;
- pfs mode;
- ph2 max;

- dpd (в случае одинаковых или пересекающихся "remote id");
- настройки пулов (в случае одинаковых или пересекающихся "remote id").

Набор считается одинаковым, если значение каждой опции набора равно значению соответствующей опции другого набора. Также считается, что «remote ip *» равен только «remote ip *», но не равен «remote ip A.B.C.D» (или «remote ip <fqdn>»). Для опции «pfs mode» значения «off» и «propose» равны друг другу, но не равны значению «force».

Несоблюдение данного требования не позволит активировать соединение.

59.7.23 Обязательный удостоверяющий центр

Как уже было сказано выше, для проверки сертификата используется вся цепочка сертификатов удостоверяющих центров вплоть до корневого. И соединение разрешается только в том случае, если все сертификаты в цепочке действительны и успешно проверены. Если в систему установлено, например, несколько корневых сертификатов, и нет договорённости между удостоверяющими центрами о единой политике назначения X500-имён субъектам, то может возникнуть ситуация, когда два разных сертификата, выпущенные разными УЦ, будут иметь одинаковые X500-имена. И может возникнуть необходимость различать эти сертификаты (например, принимать соединение от одного, но не принимать от другого).

В этом случае можно с помощью опции «remote ca» (в режиме конфигурации соединения) указать X500-имя удостоверяющего центра, чей сертификат **обязательно** должен присутствовать в цепочке сертификатов при проверке. Также указанное X500-имя передаётся оппоненту в пейлоуде CR (Certificate Request).

Например:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# remote id "CN=*, O=Хорошая организация, C=RU"
(config—ike—conn—t1)# remote ca "CN=УЦ 1, O=Хорошая организация, C=RU"
```

В данном примере соединения будут приниматься от всех субъектов, чьи сертификаты выпущены удостоверяющим центром №1. Соединения от субъектов, чьи сертификаты выпущены, например, УЦ №2, приниматься не будут, даже если имена субъектов удовлетворяют указанному шаблону.

Чтобы не вводить X500-имена удостоверяющих центров вручную, можно их импортировать непосредственно из сертификатов УЦ. Например:

```
(config—ike—conn—t1)# remote ca from root ca cert ca1.cer
```

или

```
(config—ike—conn—t1)# remote ca from ca cert ca2.cer
```

В первом случае X500-имя импортируется из корневого сертификата «ca1.cer», а во втором случае - из сертификата промежуточного УЦ «ca2.cer».

59.7.24 Проверка использования сертификата по назначению

Сертификаты X509 могут содержать в себе дополнительные поля «Key Usage» и «Extended Key Usage», в которых описывается область применения данного сертификата и соответствующего ему закрытого ключа. Поле «Key Usage» представляет из себя набор предопределённых флагов (см. RFC5280, п. 4.2.1.3), а поле «Extended Key Usage» может содержать в себе произвольное количество OID-ов, описывающих область применения сертификата и ключа.

По умолчанию в Dionis-NX проверяются клиентские сертификаты (локальные и присланные от оппонента) на наличие следующих флагов/OID-ов:

- Флаг digitalSignature;
- Флаг nonRepudiation;
- OID 1.3.6.1.5.5.8.2.2 {iso(1) identified-organization(3) dod(6) internet(1) security(5) mechanisms(5) ipsec(8) certificate(2) iKEIntermediate(2)}.

Данное поведение можно изменить с помощью настройки проверки области применения. Для этого необходимо войти в режим конфигурации «cert usage» для соответствующего соединения:

```
(config)# crypto ike conn t1
(config—ike—conn—t1)# cert usage
(config—ike—conn—t1—cu)# _
```

В данном режиме можно включать/выключать проверку соответствующих флагов/OID-ов с помощью команд:

[no] digital-signature	Проверять/не проверять наличие флага digitalSignature
[no] non-repudiation	Проверять/не проверять наличие флага nonRepudiation
[no] key-encipherment	Проверять/не проверять наличие флага keyEncipherment
[no] data-encipherment	Проверять/не проверять наличие флага dataEncipherment
[no] key-agreement	Проверять/не проверять наличие флага keyAgreement
[no] key-cert-sign	Проверять/не проверять наличие флага keyCertSign
[no] crl-sign	Проверять/не проверять наличие флага cRLSign
[no] encipher-only	Проверять/не проверять наличие флага encipherOnly
[no] decipher-only	Проверять/не проверять наличие флага decipherOnly
[no] ike-intermediate	Проверять/не проверять наличие OID-а 1.3.6.1.5.5.8.2.2

Также можно добавить проверку наличия других OID-ов, вводя команды вида:

```
(config—ike—conn—t1—cu)# oid <n.n.n.n.n>
```

Удалить проверку OID-а можно с помощью соответствующей команды:

```
(config—ike—conn—t1—cu)# no oid <n.n.n.n.n>
```

Удалить все OID-ы из списка проверки можно с помощью команды:

```
(config—ike—conn—t1—cu)# no oids
```

(Данная команда не влияет на флаг «ike-intermediate»).

Если локальный сертификат не удовлетворяет заданной области применения, произойдёт ошибка при попытке перевода соединения в состояние «enabled».

Если присланный сертификат не удовлетворяет области применения, то будет послано уведомление INVALID_CERTIFICATE, и соединение будет задержано в следующих состояниях в зависимости от ситуации:

	Состояние инициатора	Состояние ответчика
Плохой инициатор	pending2	pending_mdcfg
Плохой ответчик	pending1	pending_mdcfg

59.7.25 Чёрный список субъектов

Существует возможность задать чёрный список X500-имён субъектов, соединения от которых будут отвергаться. Также, если в чёрный список были добавлены новые субъекты, то активные соединения с данными субъектами будут закрыты в течение минуты.

Для входа в режим редактирования чёрного списка необходимо выполнить команду:

```
(config)# crypto ike blacklist
```

Добавление X500-имя заблокированного субъекта в чёрный список осуществляется командой:

```
(config—ike—blacklist)# id <X500_имя>
```

Просмотр чёрного списка осуществляется командой “do show” (в режиме редактирования списка):

```
(config—ike—blacklist)# do show
1 id ”CN=Иванов Иван Иванович,O=Хорошая организация,C=RU”
2 id ”CN=Петров Пётр Петрович,O=Хорошая организация,C=RU”
```

Удалить элемент из списка можно по порядковому номеру с помощью команды “no”:

```
(config—ike—blacklist)# no 1
```

Очистить весь список можно командой “no all” (в режиме редактирования списка), либо глобальной командой “no crypto ike blacklist”.

59.7.26 Расписание соединений

Соединениям можно назначать расписания. Расписание - это набор правил, определяющих временные промежутки, во время которых соединение запрещено или разрешено.

Примечание: Не рекомендуется назначать расписания соединениям, которые могут выступать в роли инициатора, для которых действует опция “auto initiate” или “auto route initiate”. В случае закрытия по расписанию данные соединения теряют “инициативу” и переходят соответственно в состояния “listen” или “routed”.

Создание/редактирование расписания осуществляется командой:

```
(config)# crypto ike schedule <имя_расписания>
```

Удалить расписание можно с помощью команды:

```
(config)# no crypto ike schedule <имя_расписания>
```

В режиме редактирования расписания доступны команды “permit” и “deny”. Команда “permit” представляет собой разрещающее правило, команда “deny” - запрещающее. Расписание представляет собой нумерованный список данных команд (начиная с 1). Если команда вводится без номера, она добавляется в конец списка. Если с номером - осуществляется сдвиг существующих правил вниз, а затем вставка данной команды на указанную позицию. (Аналогично спискам ACL).

В командах “permit/deny” можно указывать временные критерии:

- year - конкретный год или диапазон годов;
- month - конкретный месяц или диапазон месяцев (от 1 до 12);
- day - конкретное число месяца или диапазон дней (от 1 до 31);
- weekday - конкретный день недели или диапазон дней недели (1 - понедельник, 7 - воскресенье);
- time - временной диапазон (с точностью до минуты, время местное);
- dn - X500-имя оппонента или шаблон имён (в случае аутентификации по PKI);
- ip - IP-адрес оппонента (в случае аутентификации по PSK).

Отсутствие критерия означает “любой”. Указанные критерии объединяются по принципу “И”.

Правила проверяются последовательно, начиная с первого до первого подходящего по критериям. Если не подошло ни одно, действует правило “permit”.

Правила можно удалять по номеру с помощью команды “no <n>”. Команда “no *” удаляет все правила.

Расписание назначается соединению(ям) с помощью команды:

```
(config)# crypto ike conn <имя_соединения>
(config—ike—conn—...)# schedule <имя_расписания>
```

Отменить ассоциацию с расписанием можно командой “no schedule”.

Пример:

```
(config)# crypto ike schedule s1
(cfg—ike—schedule—s1)# no *
(cfg—ike—schedule—s1)# 1 permit year 2016 month 7 weekday 1 5 time 9:00 19:00
(cfg—ike—schedule—s1)# 2 permit year 2016 month 7 weekday 6 7 dn
        CN=*,OU=Hackers,O=Company,C=RU
(cfg—ike—schedule—s1)# 3 deny
(cfg—ike—schedule—s1)# exit
(config)# crypto ike conn t1
(config—ike—conn—t1)# schedule s1
```

В данном примере соединение t1 разрешается только в июле 2016 года с понедельника по пятницу с 9:00 до 19:00. По выходным дням (июля 2016 года) соединение разрешено только для субъектов отдела “Hackers”. Во всех остальных случаях соединение будет отвергнуто.

59.7.27 СОС и OCSP

Как было сказано выше, проверка сертификата, помимо проверки подписи и срока действия, включает в себя проверку на отзыв. Проверка на отзыв может производиться как с помощью протокола OCSP, так и с помощью списков отозванных сертификатов (СОС). Протокол OCSP имеет больший приоритет, как более оперативный.

59.7.27.1 Ручная загрузка СОС

Списки отозванных сертификатов могут быть загружены в систему вручную с помощью команды «crypto pki import crl» (см. выше). При запуске службы IKE, все СОС, находящиеся в локальном хранилище, загружаются в оперативную память. Если при работающей службе IKE в локальное хранилище были загружены дополнительные СОС, то для того, чтобы они вступили в силу, требуется выполнить команду режима enable:

```
# crypto ike reload
```

59.7.27.2 "Строгость" политики проверки сертификата на отзыв

Различаются две политики проверки сертификатов на отзыв: строгая и нестрогая (по умолчанию).

При строгой политике сертификат обязательно должен быть проверен на отзыв (либо по протоколу OCSP, либо по СОС). Если OCSP-информация недоступна и нет соответствующего действительного СОС, то сертификат считается заведомо недействительным.

При нестрогой политике сертификат проверяется на отзыв, если есть соответствующая информация (OCSP или СОС). Если её нет, то сертификат на отзыв не проверяется.

Для корневых сертификатов всегда действует нестрогая политика.

Строгую политику проверки на отзыв можно включить с помощью глобальной опции службы IKE:

```
(config)# crypto ike config
(config—ike)# crl policy strict
```

Отключить строгую политику можно соответствующей командой «no»:

```
(config—ike)# no crl policy strict
```

59.7.27.3 Загрузка новых СОС по сети

Помимо статически загруженных СОС, служба IKE может динамически получать СОС по протоколам HTTP/FTP/LDAP.

ВАЖНО: По умолчанию динамическая загрузка СОС и запросы к OCSP выключены. Чтобы их включить, необходимо задать глобальную опцию службы IKE:

```
(config—ike)# crl fetch interval <секунды>
```

Также данной опцией задаётся интервал между попытками загрузки COC и OCSP-запросами. Следует отметить, что данный интервал используется только тогда, когда нет загруженного действительного COC и OCSP-статуса. Если COC и OCSP-статус действительны, обновления не производится.

Следующая опция позволяет настроить временной порог заблаговременной загрузки нового свежего COC:

```
(config—ike)# crl fetch margin <проценты>
```

Временной порог указывается в процентах от срока действия текущего COC. Например, если срок начала действия COC 16.03.2017 19:00, срок конца действия 18.03.2017 19:00, и указана опция 'crl fetch margin 50', то служба IKE начнёт пытаться скачать свежий COC начиная с 17.03.2017 19:00.

По умолчанию действует настройка:

```
(config—ike)# crl fetch margin 0
```

При данной настройке скачивания свежих CRL начнётся за 2 интервала 'crl fetch interval' до конца срока действия текущего COC.

Отключить динамическую загрузку COC и запросы по OCSP можно командой:

```
(config—ike)# no crl fetch
```

Адреса загрузки COC и OCSP-серверов получаются из двух источников:

1. Настройки «cainfo»;
2. Информация о точках распространения COC и OCSP-серверах, записанная в сертификатах.

59.7.27.4 Ручная настройка точек распространения COC и OCSP

Если необходимо задать точки распространения COC и OCSP вручную, то это можно сделать, создав настройки «cainfo». Настройки «cainfo» имеют больший приоритет, чем точки распространения, записанные в сертификатах.

«cainfo» представляет собой объект, содержащий адреса точек распространения COC и OCSP, и ассоциированный с внутренним системным именем соответствующего сертификата УЦ. COC и OCSP-статусы для сертификатов, выпущенных данным УЦ, будут пытаться загрузиться с указанных адресов.

Чтобы создать/отредактировать объект «cainfo», необходимо ввести команду режима конфигурации:

```
(config)# crypto ike cainfo <имя_сертификата_УЦ>
(config—ike—cainfo—<имя>)# _
```

Данная команда вводит консоль в режим редактирования настроек «cainfo».

Для вывода списка имён сертификатов УЦ, установленных в систему, можно использовать команды режима enable:

```
# show crypto pki root ca certs
# show crypto pki ca certs
```

В режиме редактирования «cainfo» доступны следующие опции:

crl uri main <uri>	Основная точка распространения COC (HTTP/FTP/LDAP)
crl uri alt<n> <uri>	Дополнительные точки распространения COC (HTTP/FTP/LDAP) (до 9 штук)
ldap host <hostname>|<ip>	Сервер LDAP (если не указано в URI)
ocsp uri <uri>	Адрес сервера OCSP
ocsp mode factor-ts|crypto-pro	Режим совместимости OCSP (см. ниже)
source ip <ip>	Альтернативный адрес отправителя запроса (только для HTTP/FTP, см. ниже)

Пример:

```
(config)# crypto ike cainfo ca.cer
(config—ike—cainfo—ca.cer)# crl uri main ”ldap:///O=Хорошая организация,
        C=RU?certificateRevocationList”
(config—ike—cainfo—ca.cer)# crl uri alt ”ftp://ftp.good—org.ru/my.crl”
(config—ike—cainfo—ca.cer)# crl uri alt2 ”http://server2.good—org.ru/my.crl”
(config—ike—cainfo—ca.cer)# ldap host ”ldap.good—org.ru”
(config—ike—cainfo—ca.cer)# ocsp uri ”http://ocsp.good—org.ru:8880”
```

Объекты «cainfo» можно копировать командой режима конфигурации:

```
(config)# crypto ike copy cainfo <имя_сертификата_УЦ_1> to <имя_сертификата_УЦ_2>
```

Если нужно удалить объект «cainfo», то это делается командой режима конфигурации:

```
(config)# no crypto ike cainfo <имя_сертификата>
```

Если объекты «cainfo» редактируются во время запущенной службы IKE, то новые настройки не применяются немедленно. Чтобы они вступили в силу, необходимо выполнить команду «crypto ike reload».

Чтобы вывести список объектов «cainfo», нужно выполнить команду режима enable:

```
# show crypto ike cainfos
```

Чтобы вывести подробную информацию об объектах «cainfo», нужно выполнить команду:

```
# show crypto ike cainfos verbose
```

Данная команда работает только при запущенной службе IKE. Если некоторые объекты «cainfo» отсутствуют в выводе, это означает, что не были найдены соответствующие сертификаты УЦ.

59.7.27.5 Диагностика загрузки COC и OCSP

Чтобы вывести подробную информацию об известных точках распространения COC и OCSP, о попытках загрузки, о времени окончания действия COC и OCSP-статусов, и т.д., нужно выполнить команду режима enable:

```
# show crypto ike revocation
```

59.7.27.6 Кэширование СОС

Иногда бывает полезно сохранять динамически загруженные СОС в локальном хранилище (например, при использовании строгой политики проверки на отзыв). Для этого надо включить глобальную опцию службы IKE:

```
(config)# crypto ike config
(config—ike)# crl cache
```

Управлять сохранёнными СОС можно таким же образом, как и загруженными вручную, то есть с помощью команд «*crypto pki * crl*» (см. выше).

Выключить сохранение загружаемых СОС можно соответствующей опцией «no».

```
(config—ike)# no crl cache
```

59.7.27.7 Настройка альтернативного адреса отправителя

По умолчанию запрос на скачивание СОС отправляется с интерфейса/IP-адреса, согласно правилам маршрутизации.

Если по каким-то причинам необходимо отправлять запрос с другого интерфейса/IP-адреса, то данный IP-адрес можно указать принудительно в настройках соответствующей секции “cainfo” с помощью опции “source ip”:

```
(config)# crypto ike cainfo <ca_cert_name>
(config—ike—cainfo—ca.cer)# source ip <ip>
```

Примечание: Данная настройка влияет только на протоколы HTTP и FTP.

Чтобы вернуть автоматический выбор адреса отправителя, можно удалить данную опцию с помощью команды:

```
(config—ike—cainfo—ca.cer)# no source ip
```

59.7.27.8 Настройка протокола OCSP

В Dionis-NX реализованы два режима совместимости протокола OCSP:

- Режим совместимости «Крипто-Про» (с использованием хэш-функции SHA1);
- Режим совместимости «Фактор-ТС» (с использованием только криптографии ГОСТ).

Этот режим устанавливается глобальной опцией службы IKE, а также на уровне объектов «cainfo». Если для объекта «cainfo» не задана опция, то для данного удостоверяющего центра действует глобальная опция. По умолчанию действует режим «Фактор-ТС».

Команды переключения режима совместимости OCSP:

```
(config)# crypto ike config
(config—ike)# ocsp mode default crypto—pro
(config—ike)# ocsp mode default factor—ts
(config—ike)# exit
(config)# crypto ike cainfo ca.cer
(config—ike—cainfo—ca.cer)# ocsp mode crypto—pro
(config—ike—cainfo—ca.cer)# ocsp mode factor—ts
(config—ike—cainfo—ca.cer)# no ocsp mode
```

В Dionis-NX имеется настройка максимального количества запросов статусов сертификатов в одном OCSP-запросе:

```
(config)# crypto ike config
(config—ike)# ocsp max certs <n>
```

Данный параметр означает, какое максимальное количество сертификатов будет обработано в одном OCSP-запросе (к одному OCSP-серверу). Если системе требуется выяснить статус большего количества сертификатов, то будет сформировано несколько последовательных OCSP-запросов. По умолчанию этот параметр равен 10, и без особой надобности менять его не нужно. Данный параметр нужно уменьшать, если OCSP-сервер отвергает длинные OCSP-запросы; и нужно увеличивать, если узлу Dionis-NX требуется обрабатывать большое количество соединений от разных оппонентов.

59.7.27.9 Очистка информации по отзывам

Если требуется очистить текущие статусы сертификатов, полученных по OCSP, то можно выполнить команду режима enable:

```
# crypto ike clear ocsp cache
```

Динамически загруженные COC очищаются только перезапуском службы IKE.

59.7.27.10 Особенности онлайн-проверки на отзыв

Для защиты от DoS-атак динамическая загрузка COC и обмен по OCSP имеют следующие особенности. При проверки сертификата, пришедшего от оппонента, не производится немедленного запроса к OCSP или немедленной загрузки COC. Запрос к OCSP и инициирование загрузки COC производятся в параллельном потоке, чтобы не блокировать логику конечного автомата IKE. Из этого следует, что при строгой политике проверки на отзыв, первая попытка установления соединения от нового оппонента может оказаться неудачной, так как соответствующий статус OCSP и динамический COC ещё не получены. Для устранения данной проблемы можно указать опцию «crl cache» и/или заранее указать точки распространения COC в объектах «cainfo» для соответствующих удостоверяющих центров. В этом случае COC будут загружены сразу после запуска службы IKE.

59.7.28 Политика пересылки сертификатов

Сертификат оппонента может пересылаться по протоколу IKE, а может и не пересылаться. В последнем случае он должен быть загружен заранее в локальное хранилище клиентских сертификатов,

и в конфигурации соединения должна присутствовать опция (на принимающей стороне):

```
(config)# crypto ike conn <имя>
(config—ike—conn—<имя>)# remote cert <имя_сертификата_оппонента>
```

Данную опцию можно очистить соответствующей командой «no»:

```
(config—ike—conn—<имя>)# no remote cert
```

Политика пересылки собственного сертификата задаётся опцией в конфигурации соединения:

```
(config)# crypto ike conn <имя>
(config—ike—conn—<имя>)# send cert <политика>
```

Возможные политики:

- always - всегда пересылать свой сертификат;
- never - никогда не пересылать свой сертификат;
- ifasked - (по умолчанию) пересылать свой сертификат только по требованию.

В случае политики «ifasked» сертификат будет пересылаться только тогда, когда от оппонента получен запрос Certificate Request (CR).

При необходимости можно отключить посылку CR глобальной опцией службы IKE:

```
(config)# crypto ike config
(config—ike)# no send cert req
```

Включить посылку CR (вернуть поведение по умолчанию) можно опцией:

```
(config—ike)# send cert req
```

59.7.29 Плановая смена ключей

Как было сказано выше, взаимная аутентификация узлов IPsec может осуществляться либо с помощью симметричных предварительно распространённых ключей (pre-shared keys, PSK), либо с помощью асимметричных ключей и сертификатов X.509. Когда согласно регламенту истекает срок действия ключей/сертификатов, то необходима их плановая смена.

Смена симметричных ключей (PSK)

Плановая смена ключей PSK проводится в несколько этапов:

1. Установка новых ключей PSK на каждый узел IPsec.
2. Переключение IPsec-туннелей на новые ключи.
3. Удаление старых ключей.

Каждый этап должен быть выполнен для всех узлов перед переходом к следующему этапу.

Этап 1 выполняется только локально. Этапы 2 и 3 могут выполняться как локально, так и удалённо. При удалённом выполнении этапов канал управления должен защищаться отдельным туннелем IPsec на отдельных ключах. Удалённая смена ключей для туннеля канала удалённого управления имеет особый порядок (см. ниже), отличающийся от порядка смены ключей других туннелей.

Во избежание частых выездов на удалённый объект на этапе 1 можно установить сразу несколько ключей для нескольких последующих удалённых смен.

Этап 1:

Данный этап одинаков для обычных туннелей и для туннеля канала удалённого управления.

На каждый узел импортируется новый pre-shared ключ с помощью команды ‘crypto psk set key’. Новый ключ сохраняется с новым именем.

Пример:

Существующая конфигурация узла 1:

```
# show crypto psk keys
psk1
# configure
(config)# do show
...
crypto psk map 10.1.0.1 10.2.0.1 psk1
...
crypto ike conn t1
 auth psk
 auto initiate
 local ip 10.1.0.1
 remote ip 10.2.0.1
...
```

Существующая конфигурация узла 2:

```
# show crypto psk keys
psk1
# configure
(config)# do show
...
crypto psk map 10.2.0.1 10.1.0.1 psk1
...
crypto ike conn t1
 auth psk
 auto listen
 local ip 10.2.0.1
 remote ip 10.1.0.1
...
```

Действия на узле 1:

```
# show crypto psk keys flash0
DSRF key container on ’flash0’ device:
Zone: 1
Serial: 1234
Abonent: 1
Number of abonents: 9999

# crypto psk set key psk2 dsrf flash0 9000
Info: Found possible DSRF container on ’flash0’ device.
Info: Read 32 bytes of pre—shared key.
Info: Saving the key with internal name ’psk2’.

# show crypto psk keys
psk1
psk2
```

Действия на узле 2:

```
# crypto psk set key psk2 dsrf flash0 9000
# show crypto psk keys
psk1
psk2
```

В данном примере новый симметричный pre-shared ключ сохранён с новым именем psk2.

Этап 2 (для обычных туннелей):

На данном этапе необходимо выполнить следующие действия:

1. Установить новую ассоциацию IP-адресов концов IPsec-туннеля с новым ключом с помощью команды ‘crypto psk map’.
2. Перезагрузить секреты службы IKE с помощью команды ‘crypto ike reload’.
3. Перезагрузить IPsec-туннель(и), использующий(е) данный PSK, с помощью команд ‘crypto ike disable conn <имя_туннеля>’ и ‘crypto ike enable conn <имя_туннеля>’.
4. Убедиться, что туннель успешно установился, с помощью команды ‘show crypto ike conn <имя_туннеля>’. (Должен иметь статус ‘online’). Если необходима ручная инициация туннеля, выполнить команду ‘crypto ike initiate conn <имя_туннеля>’.

Примечание: Пункты 2 и 3 можно заменить (для простоты) на выполнение команд ‘crypto ike disable’ и ‘crypto ike enable’ (но при этом будут прерваны все остальные туннели).

Пример:

Действия на узле 1:

```
# configure
(config)# crypto psk map 10.1.0.1 10.2.0.1 psk2
(config)# do crypto ike reload
```

```
(config)# crypto ike disable conn t1
(config)# crypto ike enable conn t1
(config)# do write
(config)# do show crypto ike conn t1
t1     pending1
```

Связь по t1 прервана, потому что на втором узле ещё не заменён ключ. Узел 1 продолжит пытаться установить соединение, так как действует настройка ‘auto initiate’.

Действия на узле 2:

```
# crypto psk set key psk2 dsrf flash0 9000
# configure
(config)# crypto psk map 10.2.0.1 10.1.0.1 psk2
(config)# do crypto ike reload
(config)# crypto ike disable conn t1
(config)# crypto ike enable conn t1
(config)# do write
(config)# do show crypto ike conn t1
t1     listen

... Узел 1 ещё не успел инициировать соединение. Ждём 40 секунд...

(config)# do show crypto ike conn t1
t1     online
```

Этап 2 (для туннеля канала удалённого управления):

Действия:

1. Установить новую ассоциацию IP-адресов концов IPsec-туннеля с новым ключом с помощью команды ‘crypto psk map’.

2. Сохранить текущую конфигурацию командой ‘write’.

3. Перезагрузить узел командой ‘reboot’.

4. Сменить ключ на терминале удалённого управления.

Если всё было сделано верно, то после перезагрузки узла Dionis-NX канал с ним должен восстановиться.

ВНИМАНИЕ: Удалённое переключение ключа узла Dionis-NX является необратимой операцией, и поэтому она должна осуществляться с особой внимательностью. Неправильное выполнение процедуры может привести к потере связи и необходимости выезда администратора на удалённый объект.

Пример:

Конфигурация узла Dionis-NX:

```
# show crypto psk keys
psk1
# configure
(config)# do show
...
crypto psk map 10.2.0.1 10.1.0.1 psk1
...
crypto ike conn t1
 auth psk
 auto listen
 local ip 10.2.0.1
 remote ip 10.1.0.1
 local protoport tcp/ssh
 remote protoport tcp
...
```

Действия (после этапа 1):

```
(config)# crypto psk map 10.2.0.1 10.1.0.1 psk2
(config)# do write
(config)# do reboot
...
```

Этап 3:

Данный этап одинаков для всех туннелей.

На данном этапе на всех узлах удаляются старые pre-shared ключи с помощью команды ‘crypto psk clear key’.

Пример:

```
# crypto psk clear key psk1
```

Смена асимметричных ключей (PKI)

При использовании PKI (в отличие от PSK) не предусматривается предварительный выпуск нескольких ключей и сертификатов для их удалённых последовательных смен в будущем. Поэтому удалённая смена ключей PKI невозможна.

В данном разделе рассматривается ситуация, когда новые сертификаты X.509 выпускаются с помощью старого ключа подписи удостоверяющего центра (то есть цепочку сертификатов УЦ менять не нужно). О смене сертификатов удостоверяющих центров см. ниже.

Во избежание долговременного вывода из эксплуатации IPsec-туннелей новые сертификаты должны выпускаться за некоторое время до окончания срока действия старых.

Порядок действий:

1. Импортировать новый закрытый ключ с новым именем (команда ‘crypto pki import key’).
2. Импортировать новый сертификат узла с новым именем (команда ‘crypto pki import cert’).
3. Создать копию действующего туннеля IPsec (команда ‘crypto ike copy conn’).

4. Изменить настройку ‘local cert’ в новом туннеле. Указать имя нового сертификата.
5. Вывести из эксплуатации старый туннель командой ‘crypto ike disable conn’.
6. Перезагрузить секреты службы IKE (команда ‘crypto ike reload’).
7. Активировать новый туннель командой ‘crypto ike enable conn’. Если необходимо ручное установление соединения, выполнить команду ‘crypto ike initate conn’.
8. Проверить установления соединения командой ‘show crypto ike conn’.
9. Если соединение установилось успешно, удалить старые ключ, сертификат и туннель соответственно командами ‘crypto pki clear key’, ‘crypto pke clear cert’, ‘no crypto ike conn’.

Данные действия выполняются на каждом узле. Причём выполнять их можно последовательно (поэтапно). (Т.е. допускаются длительные перерывы между изменениями конфигураций разных узлов). При этом связь не будет потеряна на долгое время, так как протокол IKE позволяет пересылать сертификат оппонента. При установлении нового туннеля будет передан новый сертификат. Причём возможно установление соединения “новый-со-старым”, так как новые сертификаты выпущены старым ключом подписи УЦ, и новый сертификат успешно пройдёт проверку на узле со старой конфигурацией.

Пример (для одного узла):

Существующая конфигурация узла 1:

```
# show crypto pki keys
user1.key
# show crypto pki certs
user1.cer   CN=user1,O=Фактор—TC,C=RU
# show
...
crypto ike conn t1
 auth pubkey
 auto listen
 local ip 10.1.0.1
 remote ip 10.2.0.1
 local cert user1.cer
 remote id ”CN=user2,O=Фактор—TC,C=RU”
...
```

Действия на узле 1:

```
# show crypto pki keys flash0
user1.p15/
# crypto pki import key from flash0:/user1.p15 to user1.key.2
# show crypto pki certs flash0
user1.cer   user  CN=user1,O=Фактор—TC,C=RU
# crypto pki import cert from flash0:/user1.cer to user1.cer.2
# configure
(config)# crypto ike copy conn t1 to t1—2
(config)# crypto ike conn t1—2
```

```
(config—ike—conn—t1—2)# local cert user1.cer.2
(config—ike—conn—t1—2)# exit
(config)# crypto ike disable conn t1
(config)# do crypto ike reload
(config)# crypto ike enable conn t1—2
(config)# do crypto ike initiate conn t1—2
(config)# do show crypto ike conn t1—2
t1—2    online
(config)# no crypto ike conn t1
(config)# exit
# crypto pki clear key user1.key
# crypto pki clear cert user1.cert
# write
```

Смена сертификата удостоверяющего центра

Как и любой сертификат, сертификат удостоверяющего центра имеет срок действия. По истечении данного срока все сертификаты, подписанные данным сертификатом, становятся недействительными. Чтобы избежать долговременного вывода IPsec-туннелей из эксплуатации, рекомендуется выпустить новый сертификат удостоверяющего центра за некоторое время до окончания срока действия старого. Также необходимо выпустить новые закрытые ключи и сертификаты, подписанные новым сертификатом УЦ, для всех узлов IPsec. Далее необходимо установить новый сертификат УЦ, новый ключ, новый сертификат узла на каждый IPsec-узел. (При этом туннели продолжат работать на старых сертификатах). Когда новые сертификаты/ключи будут установлены на всех узлах, можно начинать поэтапную смену настройки ‘local cert’ IPsec-туннелей. При этом связь не будет прерываться на долгое время, так как для проверки сертификатов узлов может использоваться как старый сертификат УЦ, так и новый. После того, как на всех узлах будут задействованы новые туннели (с новой настройкой ‘local cert’), старые сертификаты УЦ/узлов, старые ключи и старые туннели можно будет удалить.

Для импорта сертификатов УЦ используются команды ‘crypto pki import [root] ca cert’. Для удаления - ‘crypt pki clear [root] ca cert’.

Пример (для одного узла):

Существующая конфигурация узла 1:

```
# show crypto pki keys
user1.key
# show crypto pki certs
user1.cer   CN=user1,O=Фактор—ТС,C=RU
# show crypto pki root ca certs
ca.cer      CN=УЦ,O=Фактор—ТС,C=RU
# show
...
crypto ike conn t1
 auth pubkey
 auto listen
 local ip 10.1.0.1
 remote ip 10.2.0.1
 local cert user1.cer
```

```
 remote id ”CN=user2,O=Фактор—TC,C=RU”
...
```

Импорт нового сертификата УЦ/узла, нового закрытого ключа:

```
# show crypto pki certs flash0
ca.cer       root   CN=УЦ,O=Фактор—TC,C=RU
user1.cer    user   CN=user1,O=Фактор—TC,C=RU
# crypto pki import root ca cert from flash0:/ca.cer to ca.cer.2
# crypto pki import cert from flash0:/user1.cer to user1.cer.2
# show crypto pki keys flash0
user1.p15/
# crypto pki import key from flash0:/user1.p15 to user1.key.2
# configure
(config)# crypto ike copy conn t1 to t1—2
(config)# crypto ike conn t1—2
(config—ike—conn—t1—2)# local cert user1.cer.2
(config—ike—conn—t1—2)# do write
```

Далее, когда на всех узлах данная процедура будет выполнена, можно начинать деактивацию старых туннелей и активацию новых:

```
(config)# crypto ike disable conn t1
(config)# do crypto ike reload
(config)# crypto ike enable conn t1—2
(config)# do crypto ike initiate conn t1—2
(config)# do show crypto ike conn t1—2
t1—2    online
(config)# do write
```

После полного перехода всех узлов на новые туннели, старые туннели/ключи/сертификаты можно удалить:

```
(config)# no crypto ike conn t1
(config)# exit
# crypto pki clear key user1.key
# crypto pki clear cert user1.cert
# crypto pki clear root ca cert ca.cer
# write
```

59.7.30 Внеплановая смена ключей

Процедуры внеплановой смены ключей аналогичны плановой. В случае внеплановой смены закрытых ключей (PKI) также необходимо уведомить (по доверенному каналу) соответствующий удостоверяющий центр (выпустивший сертификат для данного ключа) для того, чтобы сертификат скомпрометированного ключа был включён с список отозванных сертификатов.

59.7.31 Защита от DoS-атак

Протокол IKE подвержен DoS-атакам. Основная уязвимость заключается в том, что при получении первого пакета от инициатора (возможно, злоумышленника) службе IKE необходимо создать структуру в памяти, чтобы адекватно ответить на последующие пакеты. Это происходит до аутентификации и до выработки общего секрета, и поэтому невозможно заранее отличить злоумышленника от «честного» клиента. Поэтому злоумышленник может сгенерировать поток пакетов, которые приведут к возникновению большого количества структур (состояний соединений) в памяти, что может привести к замедлению работы системы и к аварийному завершению службы IKE из-за нехватки памяти.

Чтобы этого избежать, в службе IKE введено ограничение на количество полуоткрытых соединений (неаутентифицированных фаз 1). Если службе необходимо создать новое состояние, и превышен лимит полуоткрытых соединений, то удаляется наиболее старое полуоткрытое соединение.

По умолчанию лимит полуоткрытых соединений равен 10000. Это адекватное значение для системы с ОЗУ объёмом 512 МБ. Данное значение можно регулировать опцией службы IKE:

```
(config)# crypto ike config
(config—ike)# anti—dos max states <n>
```

Следует помнить, что увеличение лимита состояний потребует большего объёма ОЗУ и приведёт к замедлению работы системы. Однако, значительное уменьшение лимита хотя и повышает реакцию системы, но может привести к отказам в установлении соединений от «честных» клиентов, если данный узел в данный момент находится под DoS-атакой.

Также при превышении лимита полуоткрытых соединений вырабатывается сигнал тревоги (см. show log alert).

59.7.32 Защита от replay-атак

В реализации протокола ESP в Dionis-NX предусмотрена защита от replay-атак. Она заключается в учёте порядковых номеров ESP-пакетов и отбрасывании пакета, если уже был принят пакет с таким же номером, либо если номер пакета слишком «стар». Так как учитывать все номера пришедших пакетов невозможно, реализовано «окно» номеров пришедших пакетов. Данное окно представляет из себя битовый массив, элементы которого показывают, был ли принят пакет с таким номером или нет. При принятии очередного пакета взводится соответствующий бит в окне. Если номер пакета новее, чем самый «новый» элемент окна, то окно продвигается вперёд. «Старые» пакеты, которые оказались позади окна считаются «принятыми», и если придёт пакет, номер которого оказался позади окна, то он будет отброшен вне зависимости от того, был он принят раньше или нет.

По умолчанию размер anti-replay окна равен 512 пакетам.

На высокоскоростных сетевых интерфейсах этот размер может оказаться недостаточным (из-за большой степени неупорядоченности пришедших пакетов), и его можно увеличить с помощью глобальной опции службы IKE:

```
(config)# crypto ike config
(config—ike)# anti—replay window <n>
```

59.7.33 Лицензии на соединения

В некоторых конфигурациях Dionis-NX возможно лицензионное ограничение на максимальное количество IPsec-соединений. Чтобы узнать максимальное и оставшееся количество лицензий на соединения, необходимо ввести команду:

```
# show crypto ike license
```

Примеры вывода команды:

1)
```
Connection licenses: UNLIMITED
```

```
Connection licenses: 100
```

2)
```
Connection licenses left: 95 of 100
```

В примере (1) количество соединений не ограничено. В примере (2) показан вывод команды при отключённой службе IKE. В примере (3) показан вывод команды при включённой службе IKE. В данном примере израсходовано 5 лицензий на данный момент (активно 5 соединений).

Правила блокировки/освобождения лицензий:

В случае нешаблонного соединения (когда данный узел может быть инициатором) расходуется одна лицензия при включении данного соединения (с помощью команды “crypto ike enable conn”). Лицензия освобождается при выключении соединения с помощью команды “crypto ike disable conn”.

В случае шаблонного соединения (когда данный узел не может быть инициатором) при включении данного соединения лицензия не расходуется. Шаблонные соединения могут порождать множества частных соединений. Лицензия расходуется при установлении очередного частного соединения (от клиента) и освобождается при закрытии очередного частного соединения. Лицензий будет израсходовано ровно столько, сколько клиентов будет подключено одновременно к данному узлу.

Примечание: В случае шаблонного соединения для успешного установления очередного частного соединения необходимо, чтобы в запасе было не менее 2 свободных лицензий. (Особенности архитектуры).

59.7.34 Синхронный и асинхронный режим обработки ESP-пакетов

По умолчанию ядро пытается распараллеливать обработку входящих и исходящих ESP-пакетов. Зашифрование/дешифрование пакетов производится на разных процессорах/ядрах. В результате пакеты могут отправляться/попадать в систему фактически не в том порядке, в котором они были отправлены/получены. Если такое поведение неприемлемо, то можно включить синхронный режим обработки ESP-пакетов:

```
(config)# crypto ike config
(config−ike)# esp sync
```

Вернуть поведение по умолчанию (асинхронный режим) можно с помощью команды:

```
(config—ike)# no esp sync
```

ПРИМЕЧАНИЕ: Если до изменения опции “esp sync” был передан какой-либо трафик по IPsec-туннелю, то при выполнении “crypto ike enable” будет выдано предупреждение о невозможности немедленного применения опции “esp sync”. В этом случае для применения нового режима “esp sync” необходимо перезагрузить устройство.

59.7.35 Другие настройки

Фаза ModeConfig может работать в двух режимах (см. draft-dukes-ike-mode-cfg-02):

- Запрос/ответ (pull);
- Предложение/подтверждение (push).

По умолчанию включён режим «pull», и без особой надобности его менять не нужно.

Если по каким-то причинам необходимо поменять данный режим, то это можно сделать с помощью опции конфигурации соединения:

```
(config—ike—conn—<имя>)# modeconfig mode push|pull
```

Настройки «modeconfig mode» должны быть одинаковыми у обоих оппонентов.

59.7.36 Ограничения

Для нормальной работы службы IKE **не допускается** назначение сетевым интерфейсам в общей сложности более 300 локальных IP-адресов. Служба IKE не контролирует данное ограничение. В случае нарушения данного ограничения, некоторые IP-адреса могут быть игнорированы службой IKE, и установление IPsec-туннелей через данные адреса будет невозможно. Выбор игнорируемых адресов осуществляется непредсказуемым образом.

60. Служба TLSPROXY

Служба tlsproxy предназначена для трансляции HTTPS/HTTP-трафика между клиентом и сервером, а также для аутентификации клиента и защиты трафика посредством протокола TLS-GOST.

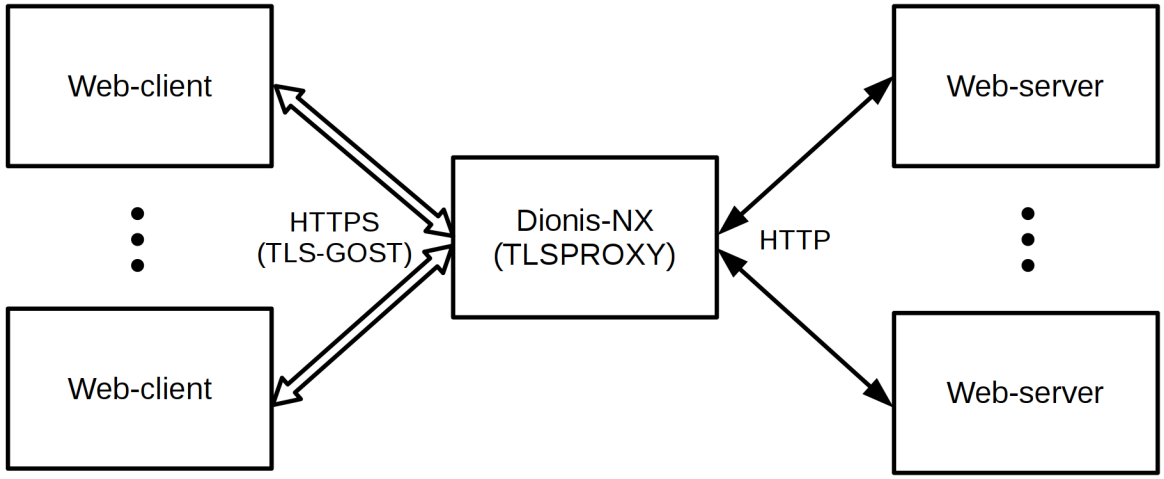

Рис. 60.1: Использование службы tlsproxy

Служба tlsproxy обладает следующими возможностями:

- Обеспечение конфиденциальности HTTP-трафика между клиентами и криптомаршрутизатором Dionis-NX посредством протокола TLS 1.0 с использованием криптоалгоритма ГОСТ 28147-89.
- Обязательная аутентификация клиентов посредством X509-сертификатов с использованием алгоритмов ГОСТ Р 34.10-2012, ГОСТ Р 34.11-2012.
- Обеспечение доступа к web-серверу(ам) для избранных клиентов. (“Белый” список).
- Предоставление web-серверу(ам) информации об аутентифицированном TLS-клиенте в HTTP-заголовке.
- Балансировка нагрузки между несколькими HTTP-серверами.
- Привязка конкретных клиентов (и групп клиентов) к конкретным серверам.
- Возможность приёма HTTPS-трафика на нескольких интерфейсах/портах.
- Распараллеливание обработки TLS-трафика на несколько процессорных ядер.
- Ограничение по количеству одновременных соединений.
- “Нормализация” TCP/HTTP-трафика. Закрытие соединений по тайм-аутам.
- Ведение журнала о соединениях.

60.1 Базовая настройка службы

Служба tlsproxy использует криптографическую подсистему, поэтому для работы службы необходим инициализированный датчик случайных чисел (ДСЧ) и загруженный ключ доступа (КД). Операции инициализации ДСЧ и создания/загрузки КД описаны в разделе “Криптография”.

Для настройки службы tlsproxy необходимо перейти в соответствующий режим конфигурации:

```
# configure terminal
(config)# service crypto tlsproxy
(config—tlsproxy)# _
```

Далее необходимо выполнить следующие обязательные настройки.

60.1.1 Адреса/порты приёма HTTPS-трафика

Адреса и порты приёма HTTPS-трафика задаются командами ‘listen’ (для адресов IPv4) и ‘listen6’ (для адресов IPv6).

Можно использовать специальные адреса ‘0.0.0.0’ (для IPv4) и ‘::’ (для IPv6) для указания службе слушать трафик на всех интерфейсах.

Если в команде не указан порт, то по умолчанию используется порт 443.

Необходимо указать хотя бы одну команду ‘listen’ или ‘listen6’.

Можно указывать несколько команд ‘listen’ и ‘listen6’ для приёма трафика с нескольких интерфейсов/портов.

Пример:

```
(config—tlsproxy)# listen 10.1.0.1 443
(config—tlsproxy)# listen 10.2.0.1 4443
(config—tlsproxy)# listen6 ffc0::1
```

Удалить команды ‘listen’ и ‘listen6’ можно введя соответствующую команду с префиксом ‘no’:

```
(config—tlsproxy)# no listen 10.1.0.1 443
(config—tlsproxy)# no listen6 ffc0::1
```

60.1.2 Группы HTTP-серверов

Для создания группы серверов нужно войти в режим данной группы с помощью команды:

```
(config—tlsproxy)# server—group <имя>
(config—tlsproxy—server—group—<имя>)# _
```

Если нужно удалить группу серверов, то это можно сделать с помощью команды:

```
(config—tlsproxy)# no server—group <имя>
```

С помощью команды “server” можно добавить настройку очередного HTTP-сервера:

```
(config—tlsproxy—server—group—<имя>)# server <адрес> [<порт>]
```

где:

- <адрес> – IPv4 или IPv6 адрес HTTP-сервера.

- <порт> – порт HTTP-службы. По умолчанию используется порт 80.

Если настройка для сервера с данным адресом в заданной группе уже существует, то данная команда её обновит.

Удалить HTTP-сервер из группы можно с помощью команды:

```
(config—tlsproxy—server—group—<имя>)# no server <адрес> [<порт>]
```

В каждой группе должен быть указан хотя бы один HTTP-сервер.

Также обязательно присутствие группы с именем default.

При указании нескольких HTTP-серверов в одной группе нагрузка между ними будет распределяться равномерно по принципу "round-robin".

60.1.3 Сертификат сервера TLS

Служба TLS-прокси ведёт себя как TLS-сервер и поэтому нуждается в сертификате (и закрытом ключе) сервера TLS.

Сертификат сервера TLS должен удовлетворять следующим требованиям:

1) Сертификат должен присутствовать в хранилище пользовательских сертификатов (см. раздел "Криптография").
2) Сертификат должен быть действительным (срок действия, наличие сертификатов необходимых УЦ в соответствующих хранилищах, наличие необходимых действительных CRL).
3) В хранилище закрытых ключей должен находиться ключ, соответствующий сертификату сервера TLS.
4) В сертификате сервера TLS в расширении KeyUsage должны быть указаны флаги digitalSignature и keyEncipherment. Также в расширении ExtendedKeyUsage должен присутствовать OID 1.3.6.1.5.5.7.3.1 (TLS server).

В пункте (2) можно ослабить требование наличия CRL с помощью опции "crl policy strict off" (см. ниже).

Проверку требований пункта (4) можно отключить с помощью опции "cert usage check off" (см. ниже).

Сертификат сервера TLS указывается опцией:

```
(config—tlsproxy)# local cert <имя_сертификата_в_хранилище>
```

В случае недействительности сертификата можно воспользоваться командой "crypto pki check cert" для получения подробной диагностики. (См. раздел "Криптография").

60.1.4 “Белый” список клиентов

Служба tlsproxy работает в режиме “белого” списка. То есть доступ к web-серверу разрешается только тем клиентам, DN-имена субъектов которых присутствуют в “белом” списке.

Для успешного запуска службы необходимо указать хотя бы одного клиента в “белом” списке.

Вход в режим редактирования списка осуществляется командой:

```
(config—tlsproxy)# clients
(config—tlsproxy—clients)# _
```

Добавление DN-имени субъекта в список осуществляется командой “dn”. Например:

```
(config—tlsproxy—clients)# dn ”CN=Субъект 1,O=Хорошая организация,C=RU”
(config—tlsproxy—clients)# dn ”CN=Субъект 2,O=Хорошая организация,C=RU”
(config—tlsproxy—clients)# do show
1 dn ”CN=Субъект 1,O=Хорошая организация,C=RU”
2 dn ”CN=Субъект 2,O=Хорошая организация,C=RU”
(config—tlsproxy—clients)# _
```

Также можно вставлять субъектов на определённое место в списке, указывая порядковый номер перед командой “dn”.

Для оптимизации быстродействия рекомендуется имена наиболее часто подключающихся клиентов указывать в начале списка.

Удалить элемент списка можно командой:

```
(config—tlsproxy—clients)# no <номер>
```

Удалить весь список можно командой:

```
(config—tlsproxy—clients)# no all
```

Также допустимо вместо конкретных DN-имён указывать шаблоны вида:

```
(config—tlsproxy—clients)# dn ”CN=*,O=Одна хорошая организация,C=RU”
(config—tlsproxy—clients)# dn ”CN=*,OU=*,O=Другая хорошая организация,C=RU”
```

Символ “*” соответствует любому значению атрибута. Шаблоны вида “*текст*”, “текст*”, “текст*текст” (и т. п.) не допускаются.

Ввод длинных DN-имён неудобен и часто приводит к ошибкам в конфигурации. Для упрощения процесса настройки предусмотрена возможность импорта DN-имени из сертификата с помощью команды “dn-cert”:

```
(config—tlsproxy—clients)# dn—cert user3.cer
(config—tlsproxy—clients)# do show
1 dn ”CN=Субъект 1,O=Хорошая организация,C=RU”
2 dn ”CN=Субъект 2,O=Хорошая организация,C=RU”
3 dn—cert user3.cer
(config—tlsproxy—clients)# _
```

В команде “dn-cert” указывается имя сертификата в хранилище сертификатов пользователей.

60.1.4.1 Привязка клиентов к группам серверов

В командах “dn” и “dn-cert” в качестве второго параметра можно указывать имя группы . В этом случае запросы от данного клиента (или группы клиентов в случае шаблона DN) будут всегда перенаправляться на указанную группу HTTP-серверов. Если группа не указана, то клиенты обслуживаются группой default

Пример:

```
(config—tlsproxy)# do show
listen 0.0.0.0 443
server—group first
 server 10.0.0.1
 server 10.0.0.2
server—group second
 server 10.0.0.3
 server 10.0.0.4
server—group default
 server 127.0.0.1 1234
clients
 1 dn ”CN=Субъект 1,O=Хорошая организация,C=RU”
 2 dn ”CN=Субъект 2,O=Хорошая организация,C=RU” first
 3 dn—cert user3.cer second
```

В данном примере субъект 2 всегда будет перенаправляться в группу first, а субъект 3 – всегда в группу second . Запросы от субъекта 1 будут обслуживаться группой default

60.1.5 Совместимость с клиентами Крипто-Про

По умолчанию протокол TLS-GOST 1.0 использует расширенный мастер-секрет (Extended Master Secret), который является более защищённым по сравнению со старым способом формирования мастер-секрета TLS 1.0. Использование старого мастер-секрета запрещено.

TLS-клиенты Крипто-Про требуют поддержки старого мастер-секрета. Поэтому для совместимости с клиентами Крипто-Про необходимо указать опцию:

```
(config—tlsproxy)# master secret allow—old
```

Данная опция разрешает использование обоих способов формирования мастер-секрета – старый и расширенный.

Запретить использование старого мастер-секрета можно опцией:

```
(config—tlsproxy)# master secret extended—only
```

60.2 Управление службой и диагностика

Активация службы выполняется командой в режиме конфигурации tlsproxy:

```
(config—tlsproxy)# enable
```

Проверить состояние службы можно командой режима “enable”:

```
# show service tlsproxy status
```

Возможные состояния:

1) “Service is stopped”
2) “Service is running”
3) “Service can’t start”

Состояние (1) – служба не активирована.

Состояние (2) – служба активирована и успешно запущена.

Состояние (3) возникает тогда, когда служба пытается запуститься, но не может открыть слушающие сокеты на адресах/портах, указанных в опциях “listen” и “listen6”. Это может быть вызвано следующими причинами:

1) Слушающий интерфейс ещё не активирован (не получен IP-адрес, не готова аппаратура, и т. д.). В этом случае служба запустится, как только необходимые IP-адреса появятся в системе.
2) Слушающий порт занят другой службой. (Скорее всего это ошибка в конфигурации). Служба запустится, как только порт освободится.
3) Неправильная конфигурация. (Неправильно указаны IP-адреса, интерфейсы не активированы, и т. д.)

Отключение службы осуществляется командой в режиме конфигурации tlsproxy:

```
(config—tlsproxy)# disable
```

При отключении службы все текущие соединения клиентов будут немедленно закрыты.

Если во время работы службы были изменены настройки, то они не вступят в силу немедленно (в приглашении интерфейса появится “~”). В этом случае необходимо перезапустить службу командой “restart” или перезагрузить настройки в действующую службу командой “reload”:

```
(config—tlsproxy~)# restart
(config—tlsproxy)#
```

```
(config—tlsproxy~)# reload
(config—tlsproxy)#
```

Команда “restart” выполняет “жёсткий” перезапуск службы, который равносилен командам “disable”, “enable”. Все соединения клиентов немедленно закрываются.

Команда “reload” выполняет “мягкий” перезапуск службы. Текущие открытые соединения клиентов не прерываются и закрываются штатным образом (по инициативе сервера или клиента). Если при выполнении “reload” произошла ошибка (из-за некорректности конфигурации или недоступности IP-адресов/портов), новые настройки не применяются, продолжают действовать старые настройки.

Удалить все настройки службы можно с помощью команды режима “configure”:

```
# no service crypto tlsproxy
```

Служба tlsproxy ведёт журнал. Просмотр журнала осуществляется командой “show service tlsproxy log”, имеющий стандартный формат команд просмотра журналов:

```
# show service tlsproxy log [all|number <число_последних_строк>] [follow|archive <номер_архива>]
```

Команда без параметров показывает последние 25 строк журнала.

Параметры:

- all – показать весь журнал.
- number – показать указанное количество последних строк.
- follow – режим “слежения”. Выводить новые записи в журнале в реальном времени. Выход – Ctrl-C.
- archive – показать архивный том журнала по указанному номеру.

Количество информации в журнале можно контролировать с помощью опции “log level” в режиме конфигурации службы:

```
(config—tlsproxy)# log level l1—emerg|l2—alert|l3—crit|l4—err|l5—warning|l6—notice|l7—info|l8—debug
        [http—log]
```

Параметры:

- l1-l8 – уровень подробности журнала. Чем выше ‘lN’, тем больше сообщений будет попадать в журнал.
- http-log – подробный лог HTTP-запросов/ответов.

По умолчанию действует настройка:

```
(config—tlsproxy)# log level l5—warning
```

60.3 Другие настройки службы

60.3.1 Распараллеливание работы службы

По умолчанию служба tlsproxy выполняется на одном процессорном ядре.

С помощью опции “processors” можно контролировать количество параллельных процессов службы для повышения производительности обработки TLS-соединений.

```
(config—tlsproxy)# processors <n>|max
```

Параметры:

- n – количество параллельных процессов службы. Не может быть больше, чем количество процессорных ядер в системе.
- max – использовать все процессорные ядра в системе.

Если в опции указывается число, превосходящее количество процессорных ядер в системе, то в конфигурацию попадает число, равное количеству процессорных ядер.

```
(config—tlsproxy)# processors 100
(config—tlsproxy)# do show
processors 8
(config—tlsproxy)# _
```

60.3.2 Максимальное количество соединений

Служба tlsproxy ограничивает количество одновременных HTTPS-соединений от клиентов для того, чтобы не была возможна DoS-атака, вызванная недостатком оперативной памяти из-за большого числа соединений.

По умолчанию максимальное количество одновременных соединений – 2000.

Изменить лимит соединений можно с помощью опции:

```
(config—tlsproxy)# max connections <n>
```

60.3.3 Тайм-ауты соединений

Если web-сервер по каким-либо причинам не устанавливает соединение, запрошенное клиентом, служба tlsproxy закрывает соединение с клиентом по тайм-ауту.

По умолчанию тайм-аут установления соединения с сервером равен 5000 мс.

Также если от клиента или от сервера ожидаются данные, но в течение длительного времени клиент/сервер их не передаёт, служба tlsproxy закрывает соединение по тайм-ауту неактивности.

По умолчанию тайм-аут неактивности равен 10000 мс.

Изменить тайм-аут установления соединения и тайм-аут неактивности можно с помощью опции:

```
(config—tlsproxy)# timeout connect <тайм—аут_соединения> inactive <тайм—аут_неактивности>
```

Времена тайм-аутов указываются в мс.

60.3.4 Строгость проверки CRL

При проверке сертификата TLS-сервера и сертификатов клиентов по умолчанию используется строгая политика проверки CRL. То есть для каждого издателя, чей сертификат участвует в проверке, дол-

жен существовать действительный, непросроченный список отозванных сертификатов. В случае отсутствия или просроченности хотя бы одного используемого CRL, проверяемый сертификат будет признан недействительным.

Ослабить строгость проверки можно опцией:

```
(config—tlsproxy)# crl policy strict off
```

В этом случае становится допустимым отсутствие CRL. Также для проверки на отзыв могут использоваться просроченные CRL.

Чтобы снова включить строгую политику проверки CRL, необходимо указать опцию:

```
(config—tlsproxy)# crl policy strict on
```

60.3.5 Проверка использования сертификатов по назначению

По умолчанию служба tlsproxy требует в сертификатах наличие определённых флагов в расширении KeyUsage и определённых OID-ов в расширении ExtendedKeyUsage.

Требования к сертификатам клиентов:

- В расширении KeyUsage должен присутствовать флаг digitalSignature.
- В расширении ExtendedKeyUsage должен присутствовать OID 1.3.6.1.5.5.7.3.2 (TLS-клиент).

Требования к сертификату TLS-сервера:

- В расширении KeyUsage должны присутствовать флаги digitalSignature и keyEncipherment.
- В расширении ExtendedKeyUsage должен присутствовать OID 1.3.6.1.5.5.7.3.1 (TLS-сервер).

Отключить проверку расширений KeyUsage и ExtendedKeyUsage можно с помощью опции:

```
(config—tlsproxy)# cert usage check off
```

Чтобы снова включить проверку (Extended)KeyUsage, необходимо указать опцию:

```
(config—tlsproxy)# cert usage check on
```

60.3.6 Максимальный объём пересылаемых сертификатов

Протокол TLS предусматривает возможность пересылки цепочки сертификатов, необходимых для проверки сертификата клиента.

Служба tlsproxy ограничивает суммарный объём сертификатов, пересылаемых клиентом. По умолчанию действует ограничение в 10000 байт.

Данный лимит можно изменить с помощью опции:

```
(config—tlsproxy)# cert chain size <размер_в_байтах>
```

60.4 Информирование сервера о клиенте

Служба tlsproxy добавляет в HTTP-запрос к web-серверу заголовок “X-SSL-Client-DN”, содержащий DN-имя субъекта сертификата клиента. Таким образом web-сервер может анализировать данный заголовок и ограничивать/разрешать ресурсы для данного клиента.

60.5 Причины неустановления соединения

Соединение клиента с сервером может не состояться по следующим причинам:

Причина	Поведение службы tlsproxy
Несовместимый протокол TLS	Закрытие TLS-соединения по alert
Ошибка протокола TLS	Закрытие TLS-соединения по alert
Недействительный сертификат клиента	Закрытие TLS-соединения по alert
Неправильный (Extended)KeyUsage в сертификате клиента	Закрытие TLS-соединения по alert
Объём пересылаемых сертификатов превысил лимит	Закрытие TLS-соединения по alert
Используется TLS-клиент Крипто-Про, и не установлена опция ‘master secret allow-old’	Закрытие TLS-соединения по alert
Клиент отсутствует в “белом” списке	Страница “403 Forbidden”
Тайм-аут соединения с сервером	Страница “503 Service Unavailable” или “504 Gateway Time-out”

61. Служба обновления CRL

Служба предназначена для регулярного и своевременного обновления списков отозванных сертификатов CRL (Certificate Revocation List). Каждый удостоверяющий центр выпускает свой список отозванных сертификатов и размещает на своих серверах, откуда этот список может быть получен по протоколам ftp, http, ldap. Сервер может поддерживать только один из перечисленных протоколов или сразу несколько. Каждый список отозванных сертификатов имеет времена начала действия и окончания действия. Удостоверяющий центр должен разместить на сервере новый актуальный CRL до окончания действия старого. Соответственно клиент (в данном случае система Dionis-NX) для корректной работы должна получить с сервера новый CRL также до окончания действия предыдущего. Иначе будет невозможно проверить, действуют ли сертификаты этого удостоверяющего центра.

Источником информации о точках распространения CRL могут быть:

- Сертификат. Любой сертификат (кроме корневого) может содержать расширение CDP (CRL Distribution Point), в котором перечислены адреса (URI), откуда может быть получен список отозванных сертификатов издателя данного сертификата. При включении, служба просматривает локальные хранилища сертификатов удостоверяющих центров, пользовательских сертификатов и OCSP сертификатов и извлекает из найденных сертификатов все точки распространения. Все найденные адреса URI будут использоваться для поддержания CRL издателя в актуальном состоянии.
- Другие криптографические сервисы системы Dionis-NX. Сервисы могут обращаться к службе обновления списка отозванных сертификатов с запросом о поддержании в актуальном состоянии CRL, находящихся по указанному адресу URI.
- Вручную указанные адреса URI, по которым служба может получить CRL. Адреса указываются в секции конфигурирования службы обновления списков отозванных сертификатов.

Таким образом единицей обслуживания для сервиса является адрес URI, по которому может быть получен CRL.

При запуске служба пытается обновить все CRL по указанным адресам. При получении все CRL проверяются. Просроченные CRL не проходят проверку. Недопустимыми считаются также CRL, не прошедшие верификацию подписи. В локальное хранилище списков отозванных сертификатов попадают только более свежие CRL, нежели уже имеющиеся CRL того же издателя.

В настройках службы указывается максимально допустимый размер одного CRL и максимальный размер всего локального хранилища CRL. Также в настройках службы указывается политика обновления CRL. А именно такие параметры, как:

- Интервал времени между попытками обновить CRL в случае, если предыдущая попытка была неудачной.
- Тайм-аут при попытке обновления CRL.
- Время до окончания срока действия CRL, когда надо начать попытки его обновления.

В системе управления Dionis-NX служба обновления списка отозванных сертификатов называется REVOCATION.

61.1 Настройки службы REVOCATION

Для настройки сервиса REVOCATION в Dionis-NX предусмотрена отдельная секция в режиме конфигурирования.

```
adm@DionisNX(config)# service revocation
```

Для включения сервиса используется команда:

```
adm@DionisNX(cfg—service—revocation)# enable
```

Для отключения сервиса используется команда:

```
adm@DionisNX(cfg—service—revocation)# disable
```

61.1.1 Настройка максимально допустимого размера CRL и репозитория CRL

С помощью команд crl—max—size и repository—max—size администратор может указать максимально допустимый размер одного CRL и всего репозитория CRL соответственно.

```
adm@DionisNX(cfg—service—revocation)# crl—max—size 30M
adm@DionisNX(cfg—service—revocation)# repository—max—size 1G
```

По умолчанию максимально допустимый размер CRL устанавливается в значение 50M. Это максимальное значение для параметра. Нельзя установить значение больше этого. По умолчанию максимально допустимый размер репозитория CRL устанавливается в значение 5G.

После установки максимально допустимых размеров для CRL или репозитория CRL, необходимо перезапустить службу с помощью любой из команд reload (изменение параметров на лету) или restart (холодный перезапуск службы).

61.1.2 Настройка политики обновления

Прежде всего настраивается время, когда нужно начать попытки обновления CRL. Это время настраивается, как процент от всего времени действия CRL. Временной интервал (соответствующий указанному проценту) отсчитывается в прошлое от момента окончания действия CRL.

Например следующая настройка указывает, что обновить CRL надо при истечении половины срока его действия:

```
adm@DionisNX(cfg—service—revocation)# fetch margin 50
```

Предположим начало действия CRL 10.05.2020, а окончание действия 13.05.2020. Если значение процента установлено в значение 33%, то обновление CRL начнется ~ 12.05.2020.

Если значение процента установлено в 100, то служба будет постоянно пытаться обновить CRL.

По умолчанию значение процента установлено в 50.

Если CRL еще не был получен, или время обновления уже наступило, а на сервере находится все еще старый CRL, то служба будет повторять попытки получения CRL непрерывно с заданным интервалом. Задать такой интервал можно командой:

```
adm@DionisNX(cfg—service—revocation)# fetch interval 60
```

Временной интервал задается в секундах и по умолчанию составляет 120 секунд.

При запросе к серверу на получение CRL могут случится непредвиденные трудности. Чтобы запрос к серверу не висел вечно, устанавливается тайм-аут на максимальную продолжительность запроса.

```
adm@DionisNX(cfg—service—revocation)# fetch timeout 50
```

Продолжительность измеряется в секундах и по умолчанию составляет 60 секунд. Тайм-аут не может быть больше, чем fetch interval.

После настройки политики обновления CRL, необходимо перезапустить службу с помощью любой из команд reload (изменение параметров на лету) или restart (холодный перезапуск службы).

61.1.3 Настройка уровня журналирования для службы

По умолчанию в журнал службы попадают только наиболее важные сообщения, такие как сообщения о запуске, остановке службы, сообщения об ошибках. Чтобы увидеть более подробный отчет, администратор может установить уровень журналирования debug:

```
adm@DionisNX(cfg—service—revocation)# log debug
```

Отменить подробное журналирование можно командой:

```
adm@DionisNX(cfg—service—revocation)# no log debug
```

Других уровней журналирования кроме нормального уровня (по умолчанию) и уровня debug не предусмотрено.

Чтобы применить изменения уровня журналирования, необходимо выполнить команду холодного перезапуска restart. Внимание, команда reload для изменения уровня журналирования не сработает.

61.1.4 Точки распространения CRL

Администратор может вручную указать адреса URI, откуда следует выкачивать CRL. Делается это командой uri:

```
adm@DionisNX(cfg—service—revocation)# uri ftp://192.168.56.10/incoming/c1.crl
```

Удалить адрес можно либо явно его указав, либо сославшись на стоку по номеру:

```
adm@DionisNX(cfg—service—revocation)# no uri ftp://192.168.56.10/incoming/c1.crl
adm@DionisNX(cfg—service—revocation)# no 3
```

Удалить сразу все адреса URI можно командой:

```
adm@DionisNX(cfg—service—revocation)# no *
```

При добавлении или удалении адресов перезапуск службы не требуется. Изменения применяются на лету.

61.2 Просмотр состояния службы

В enable режиме для просмотра журнала службы необходимо выполнить следующую команду:

```
adm@DionisNX# show service revocation log
```

Для просмотра всего списка отслеживаемых службой адресов URI и статуса обновления по каждому из них используется команда:

```
adm@DionisNX# show crypto revocation
ftp://192.168.1.10/incoming/c1.crl
        status: ok
        file: f6463ffe7c9b54a2588893e050d23545bce64b10502512c6f573a0f21a981722.crl
        issuer: CN=Корневой УЦ,O=Хорошая организация,C=RU
        valid begin: 14.10.2020 09:08:07
        valid end: 14.10.2020 11:08:07
        next update: 14.10.2020 10:08:07
        last succ: 14.10.2020 09:08:07
        last fail: 14.10.2020 05:33:03
ftp://192.168.1.10/incoming/c5.crl
        status: not found
        next update: 14.10.2020 09:36:33
        last fail: 14.10.2020 09:36:23
        fails num: 1461
```

62. VRRP-кластер

VRRP (Virtual Router Redundancy Protocol) — сетевой протокол, предназначенный для увеличения доступности маршрутизаторов, выполняющих роль шлюза по умолчанию. Это достигается путём объединения группы маршрутизаторов в один виртуальный маршрутизатор и назначения им общего IP-адреса, который и будет использоваться как шлюз по умолчанию для компьютеров в сети.

62.1 Основные понятия

- VRRP-маршрутизатор (VRRP Router) — маршрутизатор, на котором работает протокол VRRP. Он может участвовать в одном или более виртуальных маршрутизаторах;
- Виртуальный маршрутизатор (Virtual Router, VR) — абстрактный объект, которым управляет VRRP. Выполняет роль маршрутизатора по умолчанию для компьютеров в сети. Фактически, виртуальный маршрутизатор — это группа интерфейсов маршрутизаторов, которые находятся в одной сети и разделяют Virtual Router Identifier (VRID) и виртуальный IP-адрес;
- Владелец IP-адреса (IP Address Owner) — VRRP-маршрутизатор, который использует IP-адрес, назначенный виртуальному маршрутизатору, как реальный IP-адрес, присвоенный интерфейсу;
- VRRP-объявление (ADVERTISEMENT) — сообщения, которые отправляет Master-маршрутизатор;
- Виртуальный IP-адрес (Virtual IP address) — это IP-адрес, присвоенный интерфейсу одного из маршрутизаторов, которые составляют Virtual Router. Используется также название — основной IP-адрес (Primary IP Address). В VRRP-объявлениях в качестве адреса отправителя всегда используется виртуальный IP-адрес;
- Virtual Router Master или VRRP Master router — VRRP-маршрутизатор, который отвечает за отправку пакетов, отправленных на IP-адрес, который ассоциирован с виртуальным маршрутизатором, и за ответы на ARP-запросы, отправленные на этот адрес. Если владелец IP-адреса доступен, то он всегда становится Master;
- Virtual Router Backup или VRRP Backup router — это группа маршрутизаторов, которые находятся в режиме ожидания и готовы взять на себя роль VRRP Master router, как только текущий VRRP Master router станет недоступным;
- Виртуальный MAC-адрес (Virtual MAC) — 0000:5E00:01xx, где xx — номер группы VRRP.

62.2 Настройка кластера

Для настройки VRRP в режиме configure для всех участников кластера необходимо перейти в режим конфигурации VRRP:

```
Router(config)# service vrrp
```

В этом режиме можно активировать и деактивировать службу VRRP с помощью команд enable/disable, а также создавать и удалять участников (экземпляры) кластера и группы.

62.2.1 Создание участников кластера

Для функционирования кластера необходимо создать хотя бы одного участника кластера.

Для этого необходимо создать instance (экземпляр) кластера:

```
Router(service—vrrp)# instance outside
```

outside здесь любое удобное администратору имя. При этом осуществляется переход в режим настройки данного экземпляра участника кластера.

В этом режиме осуществляется связь виртуального IP-адреса с участником кластера и задаются другие параметры:

Команда	Назначение
description текст	Описание instance для администратора
id <идентификатор>	Задание номера группы для кластера
iface <интерфейс>	Привязка к интерфейсу
ip address <адрес/маска>	Задание виртуального IP-адреса
ip6 address <адрес/маска>	Задание виртуального IPv6-адреса
src-ip <адрес>	Задание IP-адреса, используемого в пакетах
src-ip6 <адрес>	Задание IPv6-адреса, используемого в пакетах
adv-interval <секунды>	Интервал advertisement-оповещений
garp-delay <секунды>	Время в секундах для перехода в состояние Master
priority <приоритет>	Приоритет данного instance в кластере
state <master|backup>	Первоначальное состояние данного instance
password <пароль>	Защита сообщений паролем
vmac	Включить режим подмены виртуального MAC-адреса
preempt	Перехватывать роль у instance с низким приоритетом
preempt-delay <секунды>	Задержка для перехвата роли
track iface <интерфейс> [weight <N> [reverse]]	Включить отслеживание состояния интерфейса. При необходимости возможно задать вес активного состояния интерфейса. Reverse - вес неактивного состояния.

Для всех описанных команд существуют аналоги с префиксом “no”, которые используются для удаления соответствующей настройки. Например:

```
Router(service—vrrp—outside)# no vmac
```

Для удаления экземпляра следует пользоваться командой:

```
Router(service—vrrp)# no instance outside
```

При наличии хотя бы двух маршрутизаторов с созданными instance на них, принадлежащих одной группе и разделяющий одинаковый IP-адрес, кластер (после включения его командой enable) может

выполнять роль виртуального маршрутизатора с заданным виртуальным IP-адресом. При наличии одного маршрутизатора с созданным instance, маршрутизатор также будет выполнять роль виртуального маршрутизатора, но без функций резервирования.

62.2.2 Создание групп синхронизации

По умолчанию, если участник кластера с ролью backup перестает получать сообщения от master, то он переходит в режим master и становится владельцем виртуального IP-адрес. Иногда бывает необходимым рассматривать группу виртуальных IP-адресов как одно целое. Это означает, что при сбое любого члена группы, должно происходить резервирование. Для этого существует понятие группы синхронизации участников кластера:

```
Router(service—vrrp)# group myrouter
```

При этом происходит переход в режим конфигурации группы, в котором можно добавлять и удалять instance с помощью команд member и no member, например:

```
Router(service—vrrp—myrouter)# member outside
Router(service—vrrp—myrouter)# no member outside
```

Группа рассматривается кластером как единое целое, и при сбое любого из instance группы, происходит смена роли (один из backup становится master).

Для удаления группы, воспользуйтесь командой:

```
Router(service—vrrp)# no group myrouter
```

62.2.3 Диагностика

Для просмотра сообщений от службы VRRP следует пользоваться командой show service vrrp log из режима enable:

```
Router# show service vrrp log
```

Кроме стандартных параметров для этой команды, существует параметр states, которые позволяет просмотреть только смену состояний (ролей).

Для просмотра текущей информации о состоянии кластера, следует пользоваться командой:

```
Router# show service vrrp state
```

Для получения более подробной и, наоборот, выборочной информации, можно воспользоваться командами: show service vrrp state all, show service vrrp state sgroups (информация по группам), show service vrrp state topology.

63. Отказоустойчивый аппаратный кластер

Типичная схема использования отказоустойчивого аппаратного кластера на основе маршрутизаторов Dionis-NX представлена на рисунке:

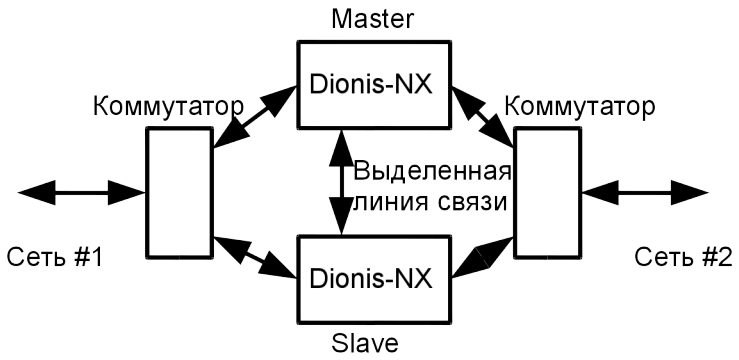

Рис. 63.1: Отказоустойчивый аппаратный кластер

Предположим, маршрутизатор используется для связи сетей #1 и #2. При выходе из строя маршрутизатора связь между сетями будет утеряна. Чтобы этого не произошло, используется резервирование. Вместо одного маршрутизатора, устанавливается два одинаковых маршрутизатора. В каждый момент времени активен только один из них. Второй находится в резерве. Один из маршрутизаторов считается основным (master), второй резервным (slave). Когда работает основной маршрутизатор, резервный блокирует все свои интерфейсы, кроме одного служебного. Резервный маршрутизатор связан с основным специальной выделенной линией связи (dedicated link). Резервный маршрутизатор прослушивает выделенную линию связи и получает от основного маршрутизатора всю информацию, характеризующую состояние компонента TCP/IP, и специально сформированные «пакеты жизни» (heartbeat или advertizing message), которые служат признаком того, что основной маршрутизатор работоспособен. Если пакетов нет слишком долго, считается, что основной маршрутизатор вышел из строя. При этом резервный маршрутизатор временно становится основным (temp master), разблокирует свои интерфейсы и берет на себя все функции по обработке трафика. Если, после проведения ремонта или замены, основной маршрутизатор становится снова доступен и начинает генерировать пакеты жизни, резервный маршрутизатор возвращается в состояние ожидания.

Кроме обмена пакетами жизни, выделенная линия связи между основным и резервным маршрутизатором используется для синхронизации настроек маршрутизаторов и обмена информацией о текущих соединениях.

63.1 Требования к оборудованию

Для организации отказоустойчивого аппаратного кластера на основе системы Dionis-NX рекомендуется использовать два одинаковых маршрутизатора, с одинаковым количеством интерфейсов и производительностью. Наличие выделенной линии связи основного маршрутизатора с резервным является обязательным условием. Также обязательным условием является установка на оба маршрутизатора одинаковой версии ПО Dionis-NX.

63.2 Подготовка к организации кластера

Перед объединением двух маршрутизаторов в кластер, на каждый из них необходимо установить одинаковую версию системы Dionis-NX и выполнить первичную настройку. Первичная настройка необходима для организации выделенной линии связи.

Перед настройкой параметров кластера, необходимо убедиться в том, что интерфейсы на каждом маршрутизаторе пронумерованы единообразно (п. 5.3. Настоятельно рекомендуется нумеровать все интерфейсы маршрутизаторов кластера единым образом во избежание путаницы, так как впоследствии настройки резервного маршрутизатора будут синхронизированы с настройками основного. Важно заметить, что нумерация интерфейсов входит в состав локальных настроек и не входит в состав конфигурации системы, и поэтому, не будет синхронизирована. На каждом маршрутизаторе нумерацию нужно провести отдельно.

После этого необходимо выбрать интерфейс для организации выделенной линии связи. На основном и резервном маршрутизаторе это должны быть одноименные интерфейсы.

Необходимо выбрать подсеть для организации выделенной линии связи. Подсеть служит для уникальной идентификации кластера. Для кластеров, находящихся в одной физической сети, выбранные подсети выделенной линии должны отличаться.

Для минимальной предварительной настройки основного маршрутизатора (master), необходимо выполнить следующие команды в режиме конфигурирования кластера:

```
Router(config)# interface ethernet 2
Router(config—if—ethernet2)# ip address 192.168.10.1/24
Router(config—if—ethernet2)# enable
Router(config)# cluster
Router(config—cluster)# aux interface ethernet 2
Router(config—cluster)# aux peer—addr 192.168.10.2/24
Router(config—cluster)# enable
```

В данном примере «ethernet 2» и «192.168.10.0/24» - выбранные для организации выделенной линии связи интерфейс и подсеть, соответственно. Локальный адрес для выделенного интерфейса задается в настройках этого интерфейса. В данном примере это “192.168.10.1/24”. Выделенный интерфейс должен иметь строго один IP-адрес и быть активным (enable). В настройках кластера указывается выделенный интерфейс и IP-адрес резервного маршрутизатора (slave). Этот адрес должен быть установлен на выделенном интерфейсе резервного маршрутизатора. Команда “enable” запускает кластер в работу.

Для минимальной предварительной настройки резервного маршрутизатора, необходимо в режиме конфигурирования выполнить следующие команды :

```
Router(config)# interface ethernet 2
Router(config—if—ethernet2)# ip address 192.168.10.2/24
Router(config—if—ethernet2)# enable
Router(config)# cluster
Router(config—cluster)# slave
Router(config—cluster)# aux interface ethernet 2
Router(config—cluster)# aux peer—addr 192.168.10.1/24
Router(config—cluster)# enable
```

Команда «slave» в данном примере говорит о том, что данный маршрутизатор является резервным. По-умолчанию, маршрутизатор считается основным. Обратите внимание, что на выделенный интерфейс устанавливается IP-адрес, упомянутый на основном маршрутизаторе при настройке кластера как "aux peer-addr". При настройке кластера на резервном маршрутизаторе в поле "aux peer-addr" указывается IP-адрес выделенного интерфейса на основном маршрутизаторе.

Сформированную таким образом конфигурацию следует сохранить в startup-config. После этого маршрутизаторы можно объединять в кластер. Интерфейсы выделенной линии связи должны быть соединены друг с другом напрямую. Кластер практически готов к работе. Всю остальную настройку можно выполнить позже.

63.3 Настройки кластера

Все настройки кластера производятся в режиме конфигурирования кластера (config-cluster).

Чтобы указать маршрутизатору, является ли он основным или резервным, используется команда «slave». Для резервного маршрутизатора «slave» должен быть установлен, для основного - сброшен. Перевести резервный маршрутизатор в режим основного можно с помощью команды:

```
Router(config—cluster)# no slave
```

Настройка интерфейса и подсети для организации выделенной линии связи между маршрутизаторами внутри кластера описана в п. 63.2.

Для кластера может быть настроен интервал между посылками пакета жизни 63 основным маршрутизатором и тайм-аут, после которого резервный маршрутизатор должен считать, что основной маршрутизатор вышел из строя.

```
Router(config—cluster)# advert 1000
Router(config—cluster)# timeout 3000
```

Времена задаются в миллисекундах. В данном примере периодичность посылки пакетов жизни равна одной секунде, а тайм-аут, после истечения которого резервный маршрутизатор станет временно основным, равен трем секундам. В этом примере резервный маршрутизатор станет временно основным, если он не получил три подряд пакета жизни от основного маршрутизатора кластера. По умолчанию эти времена (advert и timeout) равны 500 мс и 1500 мс, соответственно.

При изменении состояния машин в кластере, т.е. когда резервный маршрутизатор становится активным вместо основного, либо основной возвращается в активное состояние, вытесняя резервный, машина, которая стала активной, посылает в сеть специальные пакеты, предназначенные для сетевых коммутаторов. Эти пакеты выходят через все интерфейсы системы. Дело в том, что сетевой коммутатор поддерживает внутреннюю таблицу соответствия своего физического порта и MAC-адресов, доступных через этот порт. Так как одноименные интерфейсы на двух машинах кластера имеют одинаковые MAC-адреса, но подключены к разным портам коммутатора, то при переходе одного из маршрутизаторов в активное состояние, коммутатору надо как можно быстрее дать знать, что порт, соответствующий MAC-адресу теперь другой. Иначе, пакеты, предназначенные для активного маршрутизатора будут попадать в порт, соединенный с пассивным маршрутизатором. Маршрутизатор, став активным, посылает в сеть служебные пакеты со своим MAC-адресом. Коммутатор, получив такой пакет, узнает, что данный MAC-адрес доступен теперь через другой порт и меняет записи в своих внутренних таблицах. Траффик

начинает уходить к активному маршрутизатору. Так как переход из активного состояния в пассивного и обратно не происходит мгновенно, может понадобиться послать несколько служебных пакетов. Для управления генерацией служебных пакетов существуют настройки кластера.

```
Router(config—cluster)# mac—advert 100
Router(config—cluster)# mac—advert retry 20
```

Директива “mac-advert” задает период выдачи служебных пакетов в сеть (в миллисекундах). Рекомендуется задавать небольшой период. В этом случае восстановления нормальной работы кластера после перехода управления от одной машины к другой произойдет быстрее.

Директива “mac-advert retry” определяет сколько служебных пакетов будет сгенерировано. В данном примере будет выдано 20 пакетов через каждый активный сетевой интерфейс маршрутизатора. Существует значение “indefinite”, при котором пакеты будут генерироваться постоянно.

При загрузке маршрутизатора есть определенный промежуток времени, когда активные сетевые интерфейсы согласуют параметры соединения с удаленной стороной (negotiation). Это процесс асинхронный и может занимать достаточно длительное время. Кроме этого и сама активация интерфейсов в системе - процесс асинхронный. Предположим в кластере резервный маршрутизатор является активным и в это время загружается основной маршрутизатор и перехватывает управление. Если ко времени перехвата управления интерфейсы не готовы к полноценной работе, то проходящий через маршрутизатор траффик будет потерян. Чтобы этого не происходило существует настройка кластера “delay”. Она определяет сколько времени после загрузки основной маршрутизатор не будет перехватывать управление, т.е. время на приведение интерфейсов в рабочее состояние.

```
Router(config—cluster)# delay 20
```

В примере задержка устанавливается в 20 секунд. Конкретное время необходимой задержки сильно зависит от конфигурации маршрутизатора, количества интерфейсов. Определяется в каждом конкретном случае экспериментально.

Основной и резервный маршрутизатор обмениваются информацией об активных соединениях (conntrack), чтобы резервный маршрутизатор (а в случае перезагрузки основного, то и основной) «подхватил» уже установленные соединения автоматически. Если в каких то случаях пересылка информации о соединениях в реальном времени является нежелательной, вы можете отключить ее командой:

```
Router(config—cluster)# no conntrack service
```

Команда conntrack service снова включит передачу информацию о соединениях.

Для синхронизации часов реального времени между основным и резервным маршрутизаторами, предусмотрен специальный механизм. Основной маршрутизатор может посылать резервному через выделенную линию связи текущее время. Период синхронизации можно задать.

```
Router(config—cluster)# time—sync 10
```

В примере период синхронизации времени установлен в 10 минут.

Чтобы запустить кластер, необходимо выполнить следующую команду:

```
Router(config—cluster)# enable
```

Чтобы остановить кластер, необходимо выполнить следующую команду:

```
Router(config—cluster)# disable
```

После запуска кластера администратор может менять его настройки, но эти изменения не будут сразу применены. Чтобы применить введенные настройки, необходимо остановить кластер и вновь его запустить. Если администратор изменил настройки при работающем кластере, он будет предупрежден об этом с помощью значка »~» в приглашении командной строки в режиме конфигурирования кластера:

```
Router(config—cluster)~#
```

Знак »~» означает рассинхронизацию между настройками работающего кластера и текущими настройками в running-config.

63.4 Получение информации о кластере

Получить информацию о текущем состоянии кластера можно с помощью следующей команды, выполненной из привилегированного режима:

```
Router# show cluster

Mode              : master
State             : active
Interface         : ethernet2
Master IP—address : 192.168.30.49/30
Slave IP—address  : 192.168.30.50/30
Advert period     : 500 msec
Advert timeout    : 1500 msec
MAC—advert period : 100 msec
MAC—advert retry  : 20
Starting delay    : 3 sec
Time sync         : 1 min
ConnTrack         : on
```

Свойство “Mode” показывает текущий режим маршрутизатора:

- master - основной маршрутизатор;
- slave - резервный маршрутизатор;
- slave (temp master) - резервный маршрутизатор, получивший управление из-за неработоспособности основного.

Свойство «State» состояние маршрутизатора:

- active - в активном состоянии, т.е. выполняет свои функции согласно своему текущему режиму;
- pause - функционирование временно приостановлено командой “cluster pause”.

Свойства «Interface», «Master IP-address», “Slave IP-address” отображают выделенный интерфейс и его сетевые адреса на основном и резервном маршрутизаторах соответственно.

Свойства «Advert period» и «Advert timeout» отображают период посылки пакета жизни основным маршрутизатором и тайм-аут, по истечении которого основной маршрутизатор считается неработоспособным.

Свойства “MAC-advert period”, “MAC-advert retry” отображают настройки отправки специальных пакетов для коммутаторов. Соответственно период отправки и количество пакетов.

Свойство “Starting delay” отображает задержку перехвата управления основным маршрутизатором при загрузке.

Свойство «ConnTrack» отображает включен ли режим обмена информацией об установленных соединениях.

63.5 Пауза в работе кластера

Иногда требуется принудительно перевести один из маршрутизаторов кластера в неактивное состояние. Для этого используется команда enable-режима “cluster pause”.

При выполнении команды “cluster pause” на основном маршрутизаторе, он переходит в пассивный режим. Не посылает пакетов жизни, блокирует работу всех сетевых интерфейсов, кроме выделенной линии связи с резервным маршрутизатором.

При выполнении команды “cluster pause” на резервном маршрутизаторе, он также переходит в пассивный режим, т.е. не отслеживает тайм-аут при приеме пакетов жизни от основного маршрутизатора и, соответственно, не может перехватить управление.

Чтобы восстановить нормальную работу кластера, используется команда enable-режима “cluster resume”.

63.6 Синхронизация настроек между маршрутизаторами

Так как маршрутизаторы в кластере взаимозаменяемы, то их системные настройки должны быть идентичны. Для этого предусмотрена возможность синхронизации настроек. После настройки системы на основном маршрутизаторе, конфигурация может быть полностью перенесена на резервный маршрутизатор.

Для удобства последующей синхронизации необходимо однократно выполнить ряд действий на основном маршрутизаторе (предполагается, что кластер запущен) в привилегированном режиме:

```
Router# cluster key generate
Router# cluster key export
```

Эти команды предназначены для шифрования соединения между основным и резервным маршрутизаторами. Первая команда генерирует ключ для шифрования, а вторая - передает этот ключ на резервный маршрутизатор. При выполнении второй команды будет запрошен пароль администратора на резервном маршрутизаторе. После выполнения этих действий последующее взаимодействие между основным и резервным маршрутизаторами не потребует ручного ввода пароля администратора.

Для синхронизации системной конфигурации необходимо выполнить следующие действия из привилегированного режима:

```
Router# cluster sync
```

При этом конфигурация (с добавлением служебной информации) основного маршрутизатора будет скопирована на резервный маршрутизатор. Так как заранее неизвестно, какие настройки были изменены, для применения новых настроек требуется перезагрузка резервного маршрутизатора. Поэтому, если настройка на основном маршрутизаторе завершена, то можно использовать команду:

```
Router# cluster sync reboot
```

При этом, после синхронизации настроек, резервный маршрутизатор будет автоматически перезагружен, чтобы применилась новая конфигурация системы.

Для того, чтобы администратор с основного маршрутизатора мог зайти на резервный маршрутизатор или с резервного на основной, предусмотрена команда:

```
Router# cluster connect
```

К имени хоста в приглашении командной строки на резервном маршрутизаторе будет добавлена строка “[slave]”, на основном “[master]” - для того, чтобы администратор легко мог отличить основной маршрутизатор от резервного.

```
Router[slave]#
```

63.7 Дополнительные команды

В разделе «Синхронизация настроек между маршрутизаторами» были указаны команды для исключения ручного ввода команд при взаимодействии маршрутизаторов внутри кластера. Однако при замене одного из маршрутизаторов в кластере или в случае, если маршрутизаторы исключаются из кластера и становятся самостоятельными, может потребоваться очистка некоторых настроек.

Так на основном маршрутизаторе сохраняются параметры для взаимодействия с резервным. Поэтому при замене резервного маршрутизатора необходимо очистить параметры, которые соответствовали старому маршрутизатору, иначе взаимодействие с новым резервным маршрутизатором будет заблокировано. Для этого используется команда привилегированного режима:

```
Router# clear cluster known—hosts all
```

На резервном маршрутизаторе также хранятся параметры основного маршрутизатора. Поэтому, при замене основного маршрутизатора, на резервном следует выполнить команду:

```
Router—slave# clear cluster authorized—keys all
```

64. Балансировка прерываний

При высокой сетевой нагрузке на маршрутизатор становится актуальной проблема балансировки прерываний.

При приеме и отправке траффика сетевые интерфейсы вырабатывают аппаратные прерывания. Каждое аппаратное прерывание привязано к определенному процессорному ядру. Соответствующий обработчик прерывания выполняется на том ядре, к которому привязано аппаратное прерывание. При высокой сетевой нагрузке, время обработки принимаемых и отправляемых пакетов становится существенным. Процессорное ядро может быть полностью, на 100% загружено обработкой прерываний, т.е. перегружено. Когда процессорное ядро перегружено, траффик, который уже невозможно обработать, уничтожается (drop).

Строго говоря современные сетевые карты вырабатывают довольно мало прерываний. Может вырабатываться одно прерывание на несколько пакетов. Однако вместо слишком частых прерываний, когда на каждый принятый пакет вырабатывалось свое прерывание, используется механизм NAPI. Этот механизм временно отключает аппаратные прерывания для конкретной сетевой карты, но при этом после обработки пакета, обработчик самостоятельно проверяет не приняты ли сетевой картой еще новые пакеты (polling). Если пакеты приняты, то новая порция пакетов обрабатывается сразу же. Прерывания вновь будут включены только при отсутствии вновь принятых пакетов.

Одна сетевая карта может иметь несколько аппаратных очередей. Каждая очередь использует свое прерывание. Сетевые пакеты попадают в одну из аппаратных очередей в зависимости от хеш-функции, расчитанной по некоторым полям сетевого пакета. Обычно это адреса и порты отправителя и адресата. Т.е. один сетевой поток будет попадать в одну аппаратную очередь сетевой карты. Аппаратные очереди созданы для того, чтобы весь траффик от сетевой карты обрабатывался не на одном процессорном ядре, а распределялся на разные ядра.

Однако когда сетевых карт много, очередей много и высокая сетевая загрузка, распределение потоков по очередям не решает проблемы. Отдельные процессорные ядра становятся перегружены, а другие ничего не делают. Решить проблему можно динамической балансировкой прерываний (балансировкой обработки прерываний). Балансировщик должен снимать прерывания с перегруженных процессорных ядер и переносить на менее загруженные. В ПО Dionis-NX есть такой балансировщик. Его параметры настраиваются в секции конфигурирования “service balance-irq”.

Балансировщик через определенные отрезки времени анализирует загрузку процессорных ядер. Если в системе есть перегруженные прерываниями ядра - состояние считается ненормальным, если нет - нормальным. В ненормальном состоянии балансировщик ищет наиболее загруженное ядро, выбирает одно из прерываний, привязанных к этому ядру и переносит его на другое, наименее загруженное ядро.

Промежуток времени (период) между анализом загруженности ядер зависит от того, есть ли в системе перегруженные процессоры или нет. Если перегруженные процессоры есть, то балансировщик считает, что период нужно сократить, чтобы быстрее выйти из ненормального состояния. Если перегруженных процессоров нет, то период можно увеличить. Короткий период можно задать опцией “interval active” (по-умолчанию 2 сек), длинный - опцией “interval passive” (по-умолчанию 5 сек).

```
DionisNX(config—service—balance—irq)# interval active 2
DionisNX(config—service—balance—irq)# interval passive 5
```

Уровень загруженности ядра прерываниями (irq, softirq), при котором ядро считается перегруженным, можно задать опцией “threshold” (по-умолчанию 99%).

```
DionisNX(config—service—balance—irq)# threshold 99
```

Выбрав самое перегруженное ядро, балансировщик выбирает одно из прерываний, чтобы перенести его на другое ядро. Есть несколько стратегий выбора прерывания для переноса:

- прерывание, обработчик которого был вызван наибольшее количество раз за последний период (max);
- прерывание, обработчик которого был вызван наименьшее количество раз за последний период (min);
- случайный выбор прерывания (rnd).

С началом использования механизма NAPI, прямая зависимость количества вызовов обработчика и общим вкладом конкретного прерывания в общую загруженность ядра - исчезла. Поэтому рекомендуемая стратегия - это случайный выбор прерывания (rnd). Эта стратегия используется по-умолчанию. Надо заметить, что прерывания, обработчик которых не был вызван ни разу за последний период - игнорируются и не могут быть выбраны для переноса на другое ядро. Стратегию выбора прерывания для переноса можно выбрать опцией “strategy”.

```
DionisNX(config—service—balance—irq)# strategy rnd
```

При выборе балансировщиком ядра, на которое следует перенести прерывание, выбирается наименее загруженное ядро в системе. Однако если в системе не осталось свободных ядер и все ядра довольно сильно загружены, то перенос может не иметь смысла. Опция “load-limit” задает уровень загруженности ядра при котором перенос на него прерываний считается бессмысленным. По умолчанию этот уровень загруженности - 95%.

```
DionisNX(config—service—balance—irq)# load—limit 95
```

С помощью следующей команды задается список процессорных ядер, которые будут использоваться для обработки прерываний:

```
DionisNX(config—service—balance—irq)# affinity 0—15
```

В данном примере для обработки прерываний будут использоваться все процессорные ядра из диапазона с 0-го ядра и по 15-ое, включительно. Соответственно, все активные прерывания будут перенесены с других процессорных ядер на указанные. По-умолчанию для обработки прерываний используются все доступные процессорные ядра.

При необходимости некоторые процессорные ядра можно принудительно исключить из списка кандидатов при выборе целевого ядра для переноса на него прерывания.

```
DionisNX(config—service—balance—irq)# exclude—cpus 0,1,3—7
```

В данном примере из списка исключаются ядра с номерам 0, 1 и весь диапазон от 3-го ядра до 7-го. По-умолчанию список исключений считается пустым.

Таким образом, учитывая, что могут быть заданы включающее правило и исключающее правило одновременно, окончательный список процессорных ядер, на которых разрешена обработка прерываний, будет формироваться по следующей формуле:

используемые_процессоры = affinity & ~exclude_cpus

В формуле символ ~ означает логическое НЕ, символ & - логическое И.

На некоторых мощных маршрутизаторах может использоваться технология NUMA (Non Uniform Memory Access). Эта технология применяется на многопроцессорных (не путать с многоядерными) системах. Каждый процессор имеет "собственную" оперативную память, доступ к которой осуществляется быстро. Доступ к оперативной памяти других процессоров тоже возможен, но он производится с помощью обращения одного процессора к другому по специальным линиям связи. Процессор - "владелец" памяти сам получает значения из своей памяти и пересылает запросившему процессору. Это тяжелая и длительная операция. Кроме того, в многопроцессорных системах контроллеры шин, таких как PCIe, часто привязаны к одному из процессоров. Соответственно память именно этого процессора будет использована DMA-устройствами на этой шине. Все ядра такого процессора будут считаться локальными для устройств, подключенных по шине. Прерывания от устройств могут обрабатывать как локальные, так и не локальные ядра, однако запрос на прерывания так-же придется передавать по специальным линиям связи от локального процессора к удаленному. Если прерывание от устройства обрабатывает нелокальное ядро, то ему придется обращаться к памяти другого процессора, что очень долго.

Таким образом, обработка прерываний на нелокальных ядрах очень неэффективна. По-умолчанию балансировщик не использует нелокальные ядра для перемещения на них прерываний. Если администратор все-таки хочет использовать нелокальные ядра, то можно использовать опцию "non-local-cpus".

DionisNX(config—service—balance—irq)# non—local—cpus

Но даже в этом случае, балансировщик отдает предпочтение локальным ядрам и использует нелокальные только в том случае, когда загрузка всех локальных ядер выше, чем "load-limit".

65. Обновление системы

Администратор имеет возможность производить обновление системы Dionis-NX. Обновление может быть локальным, если администратор имеет физический доступ к оборудованию, или удалённым, если оборудование физически недоступно. Работы по установке и управлению обновлениями администратор должен производить из командной строки в привилегированном режиме.

65.1 DIP-пакеты

Обновления системы Dionis-NX предоставляются в виде DIP-пакетов. Аббревиатура DIP расшифровывается как «Dionis Package». DIP-пакет - это файл с именем вида «dionisnx-1.0-0.x86_64.dip», где «1.0» - версия системы, «0» - номер редакции (релиз), «x86_64» - архитектура целевой платформы. Пакет содержит информацию о предоставляемой системе - версия, дата создания, характеристики и т.д., ядро системы Dionis-NX, образ корневой файловой системы Dionis-NX.

DIP-пакет привязан к конкретному экземпляру оборудования, для которого он был создан. Он не может быть установлен на другой маршрутизатор. Для маршрутизатора вычисляется идентификатор оборудования (Platform ID). Для каждого экземпляра маршрутизатора этот идентификатор имеет уникальное значение. Администратор может узнать идентификатор текущей платформы с помощью команды привилегированного режима:

```
Router# show version
...
Platform ID: 4F7F—7879—F676—E85B—5369
...
```

65.2 Инфраструктура DIP

На маршрутизаторе может быть одновременно установлено несколько экземпляров операционной системы Dionis-NX. Это могут быть и разные версии ПО, и несколько экземпляров ПО одной версии. Возможность использования нескольких версий ПО нужна для безопасного обновления системы, а также для организации отката (fallback) к работоспособному экземпляру системы при возникновении сбоев в работе текущей системы.

Все установленные экземпляры ПО (будем называть их пакетами ПО или OS package) доступны только для чтения. Дополнительные данные для пакетов ПО (конфигурация, настройки и т.д.) хранятся отдельно и доступны для чтения и записи. Эти данные хранятся в области внутреннего диска маршрутизатора, называемом «слот данных» (data slot). Одновременно на диске может существовать несколько слотов данных.

Обычно каждый пакет ПО связан (bind) со своим слотом данных, где он и хранит данные. Однако пакеты и слоты данных не связаны друг с другом жестко. Могут существовать пакеты ПО, не имеющие своего слота данных, а также слоты данных, не привязанные ни к одному пакету.

К примеру, вновь установленное обновление ПО не имеет своего слота данных. Пакет получит слот данных либо автоматически при загрузке, либо администратор вручную свяжет этот пакет с уже существующим слотом данных. В случае загрузки операционной системы, не имеющей на текущий момент своего слота данных, новый слот данных будет создан автоматически и привязан к загружаемой системе. Таким образом, пакет может существовать без слота данных в пассивном режиме, но не может без него работать.

Каждый установленный пакет ПО идентифицируется уникальным именем. Каждый слот данных также идентифицируется уникальным именем. Используя эти уникальные имена, администратор системы может производить различные действия над пакетами ПО и слотами данных. Операции над текущим (активным) пакетом ПО и активным слотом данных ограничены, так как невозможно, к примеру, удалить текущий слот данных, не нарушив работу маршрутизатора.

Команды для пакетов ПО:

Команда	Назначение
os install <dip-pkg>	Установка нового пакета ПО. Источником является DIP-пакет
os remove <os>	Удаление существующего пакета ПО. Невозможно для активного пакета
os rename <old> <new>	Переименование существующего пакета ПО. Меняется уникальное имя пакета в системе
os export <os-name>	Экспорт существующего пакета ПО. Будет создан DIP-пакет
os bind <os> <dataslot>	Привязка пакета ПО к существующему слоту данных. Невозможно для активного пакета ПО. Невозможно для уже привязанного слота данных
os bind <os>	Отвязывание пакета ПО от слота данных. Невозможно для активного пакета ПО
show os	Получение списка установленных пакетов ПО
show os info	Получение подробной информации об установленных пакетах ПО

Команды для слотов данных:

Команда	Назначение
os data create <name>	Создание нового пустого слота данных
os data clone <old> <new>	Клонирование слота данных. Создается новый слот, дублирующий содержимое исходного слота
os data remove <name>	Удаление существующего слота данных. Невозможно для слотов, привязанных к какому-либо пакету ПО
os data rename <old> <new>	Переименование существующего слота данных
os data backup <name> <path>	Создание резервной копии данных на основании существующего слота данных. Создается файл - резервная копия

Команда	Назначение
os data restore <name> <path>	Восстановление слота данных на основании резервной копии данных. Невозможно для активного слота данных
schedule backup <path>	Безопасное создание резервной копии активного слота с перезагрузкой
schedule restore <path>	Восстановление текущего слота данных на основании резервной копии
schedule migrate <os>	Миграция на другой пакет ПО с сохранением текущего слота данных. Требуется перезагрузка
schedule rebind <dataslot>	Миграция на другой слот данных с сохранением текущей пакета ПО. Требуется перезагрузка
show os data	Получить список существующих слотов данных

Команды загрузки системы:

Команда	Назначение
boot default <os>	Задать пакет ПО, который будет загружаться по умолчанию
boot fallback <os>	Задать пакет ПО, который будет загружен в случае необходимости отката к предыдущей версии (fallback).
boot experimental <os>	Установить для пакета ПО признак того, что этот пакет является экспериментальным
show boot	Получить текущую конфигурацию загрузчика

Общие операции:

Команда	Назначение
show os summary	Получить сводку состояния DIP-инфраструктуры

65.3 Установка обновления

Для начала установки DIP-пакет обновления должен быть скопирован в локальную файловую систему маршрутизатора. Это может сделать администратор с помощью команд привилегированного режима «copy» или «ssh get» (п. 37.4). В случае локального обновления источником пакета обновления будет служить флеш-диск. В случае удаленного копирования (с помощью команды «ssh get»), между рабочим местом администратора и маршрутизатором должен быть установлен доверенный канал передачи информации. Копирование обновлений без установления доверенного канала передачи информации не допускается.

Локальными хранилищами файлов на диске маршрутизатора являются пространства имен «file:» и «share:». Хранилище «file:» доступно только из текущей загруженной версии системы Dionis-NX. Каждая установленная версия системы Dionis-NX имеет собственное хранилище «file:», недоступное для

других версий. Хранилище «share:» доступно для всех установленных систем. Это хранилище может быть использовано для передачи данных между разными версиями установленных пакетов ПО.

Копирование может быть выполнено при помощи команды:

```
Router# copy flash0.1:/dionisnx—1.0—0.x86_64.dip file:
```

После этого можно начать установку обновления:

```
Router# os install file:/dionisnx—1.0.1.x86_64.dip
```

Если операция прошла успешно, на машине будет установлено две системы Dionis-NX. Список установленных систем можно получить с помощью команды:

```
Router# show os
```

Можно получить подробную информацию о конкретной установленной системе при помощи команды:

```
Router# show os info dionisnx—1.0—0
```

Любой установленный пакет ПО может быть переименован. Новое название должно быть уникально:

```
Router# os rename dionisnx—1.0—0 mysystem
```

После этого пакет ПО в системе идентифицируется новым именем «mysystem».

Если какой-либо пакет ПО устарел и не используется, его можно удалить:

```
Router# os remove dionisnx—0.9—0
```

65.4 Параметры загрузки

После установки обновления нужно указать первичному загрузчику, какую из установленных систем следует загружать по умолчанию. Следующая команда покажет текущие установки первичного загрузчика:

```
Router# show boot
0 dionisnx—1.0—0 (D) (F) (C)
1 dionisnx—1.0—1
```

Первое поле - порядковый номер установленной системы (начиная с 0). Второе - идентификатор установленной системы. Отображаемые в строке признаки имеют следующее значение:

- (D) - (default). После перезагрузки данная система будет загружена по-умолчанию;
- (F) - (fallback). При возникновении проблем с загрузкой системы по-умолчанию (помеченной флагом »(D)»), произойдет откат к системе, помеченной флагом »(F)» (резервная система);
- (C) - (current). Текущая система, т.е. система, загруженная сейчас;

- (E15) - (experimental). Система загружена в «экспериментальном» режиме. Описание экспериментального режима работы ПО приведено в данном разделе ниже. Число после символа «E» означает количество минут до перезагрузки;
- (d) - (user default). Система была помечена администратором, как система по умолчанию, но по какой-либо причине произошел откат к резервной системе.

Указать загрузчику, какая система является загружаемой по умолчанию, а какая является резервной, можно следующими командами:

```
Router# boot default dionisnx—1.0—1
Router# boot fallback dionisnx—1.0—0
```

Типичные параметры первичного загрузчика при локальном обновлении системы:

```
Router# show boot
0 dionisnx—1.0—0 (F) (C)
1 dionisnx—1.0—1 (D)
```

Старая система становится резервной. По умолчанию загружается новая система.

Для удаленного обновления предусмотрен дополнительный механизм, обеспечивающий доступ администратора к системе при возникновении проблем со вновь установленной системой - работа в экспериментальном режиме. Администратор может пометить систему, как «экспериментальную». При загрузке экспериментальной системы будет взведен специальный таймер и, по истечении указанного таймаута, маршрутизатор будет автоматически перезагружен. После перезагрузки экспериментальной системы произойдет автоматический откат к резервной системе. Механизм работы в экспериментальном режиме позволяет защититься от неверных сетевых настроек в новой системе, при которых удаленный администратор потеряет возможность входа в систему. Если новая система загрузилась успешно и доступна, администратор может дать команду для снятия экспериментального режима. После этого таймер будет остановлен и автоматической перезагрузки не произойдет.

Установка экспериментального режима:

```
Router# boot experimental dionisnx—1.0—1 15
```

Последним параметром задается время в минутах до автоматической перезагрузки. После этой команды, параметры первичного загрузчика будут выглядеть так:

```
Router# show boot
0 dionisnx—1.0—0 (F) (C)
1 dionisnx—1.0—1 (D) (E15)
```

Предположим, что экспериментальная система загрузилась успешно и доступна. Администратор может войти в систему и узнать текущий статус системы и время, оставшееся до автоматической перезагрузки:

```
Router# show boot experimental
```

Далее администратор может снять экспериментальный режим и сделать новую систему "системой по умолчанию":

```
Router# no boot experimental dionisnx—1.0—1
Router# boot default dionisnx—1.0—1
```

Кроме экспериментального режима, который касается всего ПО, существует экспериментальный режим только для файлов конфигурации. Экспериментальные файлы конфигурации описаны в отдельном разделе **??**.

65.5 Конфигурация системы и данные

Каждая установленная система имеет (или получит при первой загрузке) выделенную область для хранения своей конфигурации и данных - слот данных. Так как для каждой системы конфигурация хранится отдельно, то откат на резервную систему восстановит также и резервную конфигурацию.

Типичная задача при установке обновления - миграция существующей конфигурации и данных в новую систему. Чтобы узнать для каких установленных систем уже существует область хранения данных, администратор может выполнить команду:

```
Router# show os data
```

Для получения более подробной информации используется следующая команда:

```
Router# show os summary
```

Для копирования данных из области хранения старой системы в область хранения новой системы, необходимо выполнить команду:

```
Router# os data clone dionisnx—1.0—0 dionisnx—1.0—1
```

Предполагается что dionisnx-1.0-0 - это существующая и настроенная система, а dionisnx-1.0-1 - вновь установленная система. При таком копировании следует иметь в виде что, в принципе, возможна ситуация, когда формат команд новой и старой версий ПО отличается. В этом случае необходимо внести соответствующие изменения в скопированные данные.

Если какой-то слот данных больше не нужен (например, соответствующая система устарела и удалена), данные и конфигурация могут быть стерты с диска:

```
Router# os data remove dionisnx—1.0—0
```

Может быть создан новый пустой слот данных. Это может понадобиться, если администратор желает восстановить данные, используя ранее созданную резервную копию (backup):

```
Router# os data create my_new_slot
```

Для более удобной идентификации слот данных может быть переименован. Новое имя должно быть уникально в системе (относительно других слотов данных).

```
Router# os data rename my_new_slot new_name
```

Где «my_new_slot» - существующий слот данных, а «new_name» - его новое имя.

65.6 Привязка данных

Слот данных может быть привязан к установленному пакету ПО. Это означает, что при загрузке этого пакета ПО для хранения конфигурации и данных будет использован именно привязанный слот данных.

Привязка слота данных к пакету ПО производится следующей командой:

```
Router# os bind dionisnx—1.0—0 data~1
```

Где «dionisnx-1.0-0» - установленный пакет ПО, а «data~1» - имя существующего слота данных.

Если загружается система, не имеющая привязанного слота данных, то новый слот будет создан автоматически и автоматически же привязан к текущему пакету ПО.

Операция по привязке слота данных «os bind» не может быть выполнена для слота данных, который уже привязан к какому либо пакету ПО. Такой слот необходимо сначала отвязать и только потом использовать:

```
Router# os bind dionisnx—1.0—0
```

Данная команда (без указания слота данных) отвяжет пакет ПО от слота данных.

Текущая активная система не может быть привязана или отвязана от слота данных "на лету". Также недопустимы операции над текущим слотом данных. Для операций над текущим пакетом ПО и текущим слотом данных смотрите раздел «Миграция ПО».

Привязки слотов данных можно узнать с помощью команды «show os summary»:

```
Router# show os summary
Installed OSes:
  mysystem {dionisnx—1.0—0} [data~1] (D) (C)
  anothersys {dionisnx—0.9—0} [anotherdata] (F)
  newsys {dionisnx—1.0—1}
Data slots:
  data~1 [mysystem] (C)
  anotherdata [anothersys]
  not—binded—data
```

В выводе этой команды сначала перечислены установленные пакеты ПО. Первое поле - идентификатор (имя) системы. Поле в фигурных скобках показывает версию системы. Поле в квадратных скобках указывает на привязанный слот данных. После установленных пакетов ПО перечислены существующие слоты данных. Первое поле - имя слота данных. Поле в квадратных скобках - пакет ПО, к которому привязан слот.

65.7 Миграция ПО

Часто возникает ситуация, когда администратор хочет установить новую версию ПО Dionis-NX, но использовать текущую конфигурацию. Проблема в том, что текущая конфигурация и данные хранятся в активном слоте данных. Используемый в данный момент слот данных нельзя привязать к вновь

установленной системе с помощью команды «os bind», так как слот данных уже имеет привязку, а отвязать слот от работающей системы невозможно. Таким образом, данная задача решается только с использованием перезагрузки работающей системы.

```
Router# schedule migrate dionisnx—1.0—1
```

Команда планирует миграцию на указанный пакет ПО в ходе следующей перезагрузки. Команду «reboot» для начала перезагрузки администратор должен ввести вручную. На этапе ранней загрузки текущий слот данных будет отвязан от текущей системы и привязан к новой. После этого будет загружена новая система со старым слотом данных.

Существует и обратная задача. Когда требуется мигрировать на другой слот данных, но используя текущий пакет ПО. Эта задача также решается через перезагрузку.

```
Router# schedule rebind data_new
```

Команда планирует привязку слота данных «data_new» к текущему пакету ПО. После перезагрузки будет загружена старая система с новым слотом данных.

65.8 Экспериментальный файл конфигурации

Операция копирования вновь созданного файла конфигурации в startup-config иногда может быть опасной. Так, при удаленном управлении маршрутизатором возможна ситуация, когда при перезагрузке системы с новыми непродуманными настройками, будет утеряна связь. В итоге администратор потеряет возможность удаленного управления. Чтобы этого не допустить существует специальный режим экспериментального конфигурационного файла.

В режиме экспериментального конфигурационного файла администратор указывает отдельно сохраненный файл конфигурации с новыми настройками и планирует его использование при следующей перезагрузке системы.

```
adm@DionisNX# schedule experimental—config file://new_config 15
```

После перезагрузки будет использован указанный файл конфигурации. Однако, через определенный тайм-аут (в данном примере указан тайм-аут в 15 минут), система будет перезагружена с использованием старого корректного файла конфигурации. Таким образом при неправильных новых настройках, система восстановит прежнее состояние и доступность по сети.

В случае, если новый файл конфигурации корректен, то администратор может удаленно войти в систему и выключить режим экспериментального файла конфигурации.

```
adm@DionisNX# clear experimental—config
```

При выключении экспериментального режима отключается таймер, приводящий к перезагрузке.

Также администратор может посмотреть текущее состояние экспериментального режима.

```
adm@DionisNX# show experimental—config
```

Команда покажет включен ли экспериментальный режим и, если включен, время оставшиеся до перезагрузки системы.

Надо помнить, что отключение экспериментального режима не заменяет startup-config новым файлом конфигурации автоматически. Поэтому для завершения операции по переходу на новый файл конфигурации, администратор должен выполнить следующую команду:

```
adm@DionisNX# copy running—config startup—config
```

или

```
adm@DionisNX# copy file://new_config startup—config
```

65.9 Резервная копия ПО

Администратор может создать DIP-пакет из уже установленного пакета ПО. Полученный пакет может использоваться в целях резервного копирования.

```
Router# os export dionisnx—1.0—0 file:
```

Корневая файловая система, ядро и дополнительная информация будут завёрнуты в DIP-пакет, который, в свою очередь, будет помещен в локальное хранилище файлов «file:» с именем dionisnx-1.0-0.x86_64.dip. Созданный DIP-пакет будет привязан к данному экземпляру оборудования и не может быть установлен на другую машину.

66. Обслуживание

66.1 Резервное копирование

В системе Dionis-NX предусмотрено резервное копирование данных. Слот данных содержит текущие настройки системы, данные и файлы системы протоколирования.

Из-за непредсказуемости изменений и состояния файлов, создание полной резервной копии во время штатной работы системы не гарантирует целостность данных. По тем же причинам восстановление системы из полной резервной копии "на лету" невозможно. Для восстановления системы требуется перезагрузка.

66.1.1 Создание резервной копии

Для удобства администрирования все же предусмотрено создание резервной копии работающей системы "на лету". Однако надо помнить, что создание резервной копии в этом случае не гарантирует целостность данных, так как сохранение копий файлов процесс не мгновенный и в момент сохранения одного файла, другие (еще не сохраненные) могут изменяться. Чаще всего это не опасно, так как основные параметры конфигурации системы представлены всего несколькими небольшими файлами, хотя для больших файлов журналирования проблема актуальна.

Создание резервной копии слота данных производится командой:

```
Router# os data backup data~1 share:
```

Где «data~1» - идентификатор слота данных (это может быть как текущий слот данных, так и любой другой), «share:» - место, куда сохранить резервную копию. Резервная копия может быть сохранена в пространство «share:», "file:", а также на флеш-носитель (например flash0.1:). По умолчанию в резервную копию попадут только основные настройки системы. При создании резервной копии предусмотрены следующие опции, меняющие такое поведение:

- files - Сохранять файлы из пространства file:;
- log - Сохранять файлы журналирования;
- old-backups - В пространстве file: могут уже находиться резервные копии с расширением "dbu". Опция позволяет включить их в новую резервную копию. По умолчанию такие файлы будут проигнорированы для экономии дискового пространства;
- name <имя> - Даёт возможность задать имя файла - резервной копии;
- desc <текст> - Даёт возможность задать описание резервной копии.

Для надежного создания резервной копии с гарантируемой целостностью, предусмотрена команда «schedule backup». Чтобы запланировать создание резервной копии раздела данных во время следующей перезагрузки, необходимо в привилегированном режиме (enable - режим) ввести следующую команду:

```
# schedule backup flash0.1
```

Третий аргумент «flash0.1» определяет носитель, на котором будет сохранена резервная копия. Данная запись означает, что резервная копия будет сохранена на первом найденном флеш-диске (отсчет ведется от нуля) и на первом разделе этого диска. При вводе третьего аргумента можно нажать кнопку «Tab», чтобы увидеть доступные в данный момент носители. Для этой команды предусмотрены те же опции что и для команды создания резервной копии “на лету”.

Резервная копия представляется в виде файла backup-dionisnx-<версия>-<дата>-<время>.dbu. Файл содержит сжатый образ раздела данных и текстовый файл описания резервной копии. В общем случае резервная копия может представляться в виде нескольких файлов. Это произойдет в случае превышения файлом размера в 2Гб. Разбиение по 2Гб позволяет хранить большие резервные копии на носителях с файловой системой FAT32.

66.1.2 Просмотр доступных резервных копий

Для просмотра уже существующих на носителе резервных копий используется команда привилегированного режима:

```
Router# show backup share:
```

В данном случае на экран будет выведен список резервных копий, содержащихся в пространстве «share:». Пример вывода:

```
Profile     : share:/backup—dionisnx—1.0—0—121105—104228.dbu
System ID   : dionisnx—1.0—0
Description : Testing
Date/Time   : 2012.11.05 10:42:28
```

66.1.3 Восстановление из резервной копии

Для восстановления раздела данных из резервной копии используется команда привилегированного режима:

```
Router# schedule restore flash0.1:/backup—dionisnx—1.0—0—121105—104228.dbu
```

Третий аргумент указывает файл резервной копии. После этой команды необходимо произвести перезагрузку системы. В процессе перезагрузки старый слот данных будет очищен и на его место будет установлены файлы из резервной копии. После копирования файлов из образа система Dionis-NX продолжит загружаться. После окончания загрузки это будет уже восстановленная из резервной копии система. Вторичная перезагрузка не требуется.

Также можно производить восстановление данных “на лету” в неактивные слоты данных. Восстановление из резервной копии в текущий слот данных (активный) невозможно.

```
Router# os data restore data~1 flash0.1:/backup—dionisnx—1.0—0—121105—104228.dbu
```

Где «data~1» - имя слота данных, в который будут записаны восстановленные данные.

При любом способе восстановления надо помнить, что данные, содержащиеся в целевом слоте данных, будут уничтожены и на их место будут записаны данные из резервной копии.

66.2 Самоконтроль целостности

Выполняемые коды ПО ПАК Dionis-NX расположены на файловой системе, которая не имеет функций записи. Это означает, что модификация кода невозможна принципиально. Для защиты системы от сбоев, при загрузке системы производится контроль целостности выполняемых файлов с использованием алгоритма ГОСТ Р 34.11-94. При выявлении нарушении целостности система автоматически перегружается. В этом случае запись о событии попадает в журнал auth: “Filesystem integrity corrupted!”

Факт выполнения проверки отображается в журнале messages в виде:

```
Apr 11 08:30:00 RAUL crond[17950]: USER root pid 32321 cmd nice —n 19 /usr/bin/check_gostsum
        reboot
```

Наличие данных записей в журнале без сообщения о нарушении целостности означает то, что проверка целостности пройдена успешно.

Администратор в любой момент времени может инициировать проверку целостности в ручном режиме, с помощью команды:

```
adm@DionisNX# integrity
```

Кроме того периодически производится контроль целостности процессов в оперативной памяти. По умолчанию интервал проверки составляет 30 минут. Настроить интервал проверки пямяти можно командой в режиме конфигурации:

```
adm@DionisNX(config)# integrity—mem interval <N>
```

где <N> - интервал в минутах.

Отключить периодическую проверку целостности процессов в памяти можно следующей командой:

```
adm@DionisNX(config)# no integrity—mem interval
```

Администратор в любой момент времени может инициировать проверку целостности процессов в памяти в ручном режиме, с помощью команды:

```
adm@DionisNX# integrity—mem [verbose][dump]
```

Если указан параметр “verbose”, то будет отображен процесс проверки. Если указан параметр “dump”, то в случае обнаружения нарушения целостности процесса его дамп памяти будет записан в журнал.

66.3 Проверка файловых систем

Несмотря на то, что файловая система EXT3 (системный раздел и раздел данных системы Dionis-NX) является журналируемой, в некоторых нештатных ситуациях требуется принудительная проверка файловой системы. Такая проверка по умолчанию будет проводиться каждый месяц, либо после каждых 30 случаев монтирования файловой системы. Дополнительно администратор может запланировать принудительную проверку файловых систем с помощью команды:

```
# schedule fsck
```

Во время следующей загрузки системы, файловые системы будут принудительно проверены на ошибки. Смонтированные файловые системы на работающей системе не могут быть проверены в силу технологических особенностей процесса. Поэтому для проверки требуется перезагрузка.

66.4 Безопасная очистка внешнего носителя

Если внешний носитель (флеш, дискета) содержит конфиденциальную ключевую информацию, и возникает необходимость безопасно её удалить без возможности восстановления, то это можно сделать с помощью команды ‘clear removable’. Данная команда заполняет всё пространство носителя случайными данными. Для дальнейшего использования данного носителя его необходимо будет отформатировать с помощью команды ‘format’ (см. ниже).

Формат команды безопасной очистки внешнего носителя:

```
clear removable <flashN>|<floppyN> [repeat <n>]
```

Если указан параметр “repeat”, то процедура очистки будет выполнена указанное число раз.

66.5 Форматирование внешнего носителя

Форматирование внешнего носителя выполняется с помощью команды:

```
format <flashN>|<floppyN>
```

При форматировании флеш-носителя создаётся один раздел, занимающий всё пространство носителя. На носителе создаётся файловая система FAT.

66.6 Сброс паролей в начальное значение

Если системные пароли были по какой-то причине утеряны, они могут быть сброшены в свои начальные значения. Начальное значение пароля для учетной записи консольного доступа cli - cli. Для администратора adm начальное значение пароля - adm. Для других учетных записей пароли не могут быть сброшены.

Сброс паролей может быть произведен с помощью сервисного (установочного) флеш-диска. Для этого необходимо загрузиться с сервисного флеш-диска и выбрать пункт меню “Обслуживание системы -> Сброс паролей в начальное значение”.

Плата “Сторож” в рабочем режиме (режим JL) предотвращает загрузку с внешних носителей, соответственно получение физического доступа к системе (без вскрытия корпуса) не означает возможность сброса паролей.

66.7 Сообщения об ошибках

66.7.1 Контроль целостности при загрузке системы

```
************  FILESYSTEM CHECKSUM CORRUPTED  **************
*                                                          *
* Please, repair software.                                 *
* The system will be rebooted automatically in 15 seconds. *
*                                                          *
************************************************************
```

Контроль целостности образа нарушен. Система будет перезагружена.

```
**************  GOST CRYPTO LIBRARY DAMAGED  ***************
*                                                          *
* GOST crypto library tests failed.                        *
* Please, repair software.                                 *
* The system will be rebooted automatically in 15 seconds. *
*                                                          *
************************************************************
```

Крипто-библиотека не прошла процедуру само-тестирования. Система будет перезагружена.

66.7.2 Контроль конфигурации при загрузке системы

```
*****************  DATA SLOT UPDATE  ***********************
*                                                          *
* The software version is greater than data slot version.  *
*           The data slot will be updated now.             *
*                                                          *
************************************************************
```

Система загружается с конфигурацией от предыдущей версии ОС.

```
*****************  DATA SLOT DOWNGRADE  ********************
*                                                          *
*  The software version is less than data slot version.    *
*  It can be dangerous. Some settings can be dropped due   *
*       to old software. Do it on your own risk.           *
*                                                          *
************************************************************
```

Система загружается с конфигурацией от более новой версии ОС.

```
****************  STARTUP CONFIG FAILED  *******************
*                                                          *
*    startup—config is corrupted!!! Using reserved copy.   *
```

```
*                                                          *
************************************************************
```

Целостность конфигурации нарушена, загружается резервная копия.

```
***************  STARTUP CONFIG FAILED  ******************
*                                                          *
*  startup—config is damaged!!! Using empty configuration. *
*                                                          *
************************************************************
```

Целостность конфигурации нарушена, загружается пустая конфигурация. В журнале auth присутствует запись: “startup-config damaged”.

66.7.3 Фоновая проверка целостности процессов в оперативной памяти

В журнале auth сообещние: “Process(es) in RAM is corrupted” означает нарушение целостности процессов в оперативной памяти. Система перезагружается.

66.7.4 Системные сообщения и ошибки

Системные сообщения выводятся на консоль оператора и в журнал system. В журнале присутствует информация об источнике сообщения и уровне сообщения. Классификация уровней:

- Info – информационное сообщение;
- Warning – предупреждение;
- Error – ошибка конфигурации. Обычно выводится при задании конфигурации администратором;
- Fatal – требуется немедленное присутствие администратора (не должны происходить никогда).

66.7.4.1 Ошибки типа Fatal

Can not switch on/off system filter	Невозможно активировать/деактивировать системные фильтры
Error occured; entering error mode	Ошибка при включении фильтров
Can not restart (service)	Невозможно перезапустить сервис

66.7.4.2 Ошибки типа Error

Ниже перечислены некоторые важные сообщения.

No access key loaded	Не загружен ключ доступа
Can’t install DIP-package	DIP-пакет с обновлением системы не удаётся установить
Can not create interface:	Не получается создать интерфейс
Can not start service	Не удаётся запустить сервис
Can not start service: wrong config	Неверная конфигурация сервиса

67. Управление через COM-порты

В DionisNX предусмотрена возможность управления устройством через последовательные коммуникационные порты (COM-порты).
Для этого необходима предварительная настройка.

67.1 Настройка контроллера

В первоначальной конфигурации DionisNX всегда есть настройка COM-порта по умолчанию:

```
controller serial 0
 speed 115200
 listen
```

Таким образом порт по умолчанию уже настроен и к нему можно подключать на скорости 115200. При этом опция flow control должна быть отклчена.

Все последовательные порты идентифицируются порядковым номером, который начинается с 0.

Возможные значение **speed**: 0/2400/4800/9600/19200/38400/57600/115200.

Опция **listen** означает, что порт активен и готов к коммуникации.

68. Скрипты конфигурирования

В Dionis-NX существует возможность автоматизации действий при работе с конфигурацией системы. Для этого необходимо написать скрипт конфигурирования на специальном языке программирования (основанном на Lua) и выполнить его с помощью команды script. Скрипт может принимать на вход произвольное число текстовых параметров, а результат его работы направляется в running-config.

Для написания скрипта в среде Dionis-NX может использоваться команда edit. Вы также можете создать скрипт на рабочей станции и скопировать его в Dionis-NX с помощью команды copy (или другим способом).

В простейшем случае, скрипт представляет собой просто фрагмент конфигурации Dionis-NX. Например:

```
interface ethernet 0
enable
```

Пусть файл называется iface. Тогда выполнить его можно с помощью команды:

```
adm@DionisNX# script iface
```

Или более кратко:

```
adm@DionisNX# @ iface
```

Скрипт активирует интерфейс ethernet 0. Однако, такой примитивный скрипт мы могли бы просто скопировать в running-config и получить точно такой же результат:

```
adm@DionisNX# copy iface running—config
```

Действительно, возможности скриптов конфигурирования раскрываются с введением параметров и вставок-шаблонов.

Например, давайте введём параметр – номер интерфейса ethernet.

```
interface ethernet {{arg1}}
enable
```

{{ }} это подстановка. В данном примере arg1 – 1й аргумент скрипта.

Теперь, если администратор вызовет скрипт, он сможет указать номер интерфейса ethernet:

```
adm@DionisNX# @ iface 0
```

Мы можем сделать так, чтобы и само действие над интерфейсом стало аргументом:

```
interface ethernet {{arg1}}
{{arg2}}
```

И вызвать скрипт так:

```
adm@DionisNX# @ iface 0 ”multicast on”
```

Конечно, данный пример пока не очень убедителен, так как он выполняет очень простую задачу. Но попробуем привести более практичный пример. Допустим, администратор часто вынужден выполнять однотипную команду для группы интерфейсов. ethernet 0, ethernet 1 и ethernet 2.

```
%for i=1, 2 do
interface ethernet {{i}}
{{arg1}}
%end
```

Теперь администратор может, например, выключить все эти интерфейсы с с помощью одной команды:

```
adm@DionisNX# @ iface disable
```

Символ % в начале строки означает вставку на языке Lua. В Dionis-NX используется Lua версии 5.1. Функции этого языка в рамках скриптов ограничены работой со строками и конфигурацией системы.

Вы можете вставлять не только отдельные строки, но и блоки кода на Lua, воспользовавшись %{ и %}. Например, напишем для скрипта проверку на то, что аргумент задан.

```
%{
—— проверка на корректность параметров
if  #args ~= 1 then
   inf ”Usage: iface <command>”
   exit(1)
end
%}

%for i=1, 2 do
interface ethernet {{i}}
{{arg1}}
%end
```

В данном примере, мы убедились, что задан 1 аргумент (размер массива args равен 1). В противном случае, будет выдано информационное сообщение с описанием параметра скрипта.

ВНИМАНИЕ! Обратите внимание на вызов exit(1). Он немедленно завершает выполнение скрипта. В этом случае весь вывод в running-config отменяется. Например, в ситуации:

```
interface ethernet 0
%exit(1)
```

Команда шаблона “interface ethernet 0” не будет скопирована в running-config. Дело в том, что шаблон формируется целиком только в самом конце работы скрипта, а exit() прекращает эту процедуру.

Для отладки скриптов часто удобно не сразу выполнять его в running-config, а посмотреть на вывод. Для этого воспользуйтесь командой script-print.

```
adm@DionisNX# script—print iface enable
```

Скрипт выдаст результат работы на экран, без направления его в running-config.

Для записи результата работы скрипта в файл, воспользуйтесь командой script-write:

```
adm@DionisNX# script—write output.log iface enable
```

68.1 Описание встроенных функций и переменных

68.1.1 Аргументы

Аргументы скрипту передаются в виде переменных argN (arg1 …. argN). Кроме того, аргументы передаются в массиве args. Все аргументы являются текстовыми. При необходимости перевода аргумента в число, воспользуйтесь функцией tonumber.

68.1.2 Диагностические сообщения

Для вывода диагностических сообщений используются функции: inf(), warn() и err(). Для вывода: информационных сообщений, замечаний и ошибок соответственно. Например:

```
%{
num = tonumber(arg2)
 if  not num then
err ”Второй параметр должен быть числом!”
exit(1)
%}
```

68.1.3 Аварийное завершение скрипта

Для завершения скрипта используйте функцию exit(). exit(1) означает ошибку, с точки зрения dish. exit(0) – успех. Однако, в обоих случаях скрипт прекращает своё выполнение и вывод конфигурации будет потерян. (Кроме вывода с помощью inf(), err(), warn()).

68.1.4 Анализ конфигурации

С помощью вызова config() вы можете получить текущую конфигурацию в виде массива строк, например:

```
%{
—— получить всю конфигурация в t
t = config()

—— выборка строк 1—го уровня
—— конфигурации с помощью регулярного выражения
t = config(”interface ethernet [0—9]+”)

—— выборка секций 1—го уровня (с содержимым секций)
```

```
—— конфигурации с помощью регулярного выражения
t = config(”interface ethernet ”, true)

—— выборка из заданной секции с содержимым
t = config(””, ”interface ethernet 0”)

—— выборка из заданной секции с содержимым (рекурсивно)
t = config(””, true, ”service dns”)
%}
```

Далее, конфигурация может быть проанализирована стандартными средствами работы со строками Lua.

68.1.5 Выполнение enable команд

С помощью вызова enable(), вы можете выполнить команду enable режима и проанализировать её вывод. Функция возвращает таблицу со строками вывода.

```
%{
t = enable ”show clock”
inf(t[1]) —— вывести информацию о времени
%}
```

Существует также вариант использования enable со вторым параметром-функцией, когда весь вывод передаётся этой функции по строкам:

```
%{
—— записать всю arp таблицу в конфигурацию
enable (”show ip arp”, function(l)
    local r = {}
    l:gsub(”[^ ]+”, function(a) table.insert(r, a) end)
    if  #r >= 5 then
        print(string.format(”ip arp %s %s”, r[1], r[5]))
    end
end)
%}
```

68.1.6 Получение информации о системе

Существуют функции, с помощью которых можно идентифицировать систему, на которой выполняется скрипт конфигурирования.

- sys.build() - строка, идентифицирующая сборку ОС;
- sys.version() - строка с версией ОС;

- sys.hostname() - строка с именем хоста;
- sys.name() - строка с именем ОС.

```
%{
inf(sys.build())
inf(sys.version())
inf(sys.hostname())
inf(sys.name())
%}
```

68.1.7 Контекст вызова

В таблице context содержится контекст вызова dish. Контекст содержит следующие поля:

- cmd - имя команды;
- pwd - текущий путь в dish в виде массива;

Кроме того, переменная stop_on_errors может быть использована для того, чтобы задать режим останавки выполнения скрипта при ошибках.

```
% context.stop_on_errors = true
```

Обратите внимание, что настройка действует для всего скрипта целиком!

68.1.8 Вывод с помощью print()

Вы можете использовать функцию print() для вывода конфигураци в running-config. Однако, при этом следует учесть, что exit() отменяет вывод print(), даже если print() был выполнен до exit();

68.1.9 Ввод с помощью input()

Вы можете использовать функцию input() для ввода информации с консоли. Например:

```
%{
num = input(”Введите число интерфейсов: ”)
num = tonumber(num)
 if  not num then
    err ”Неверное число”
    exit(1)
end
inf(”Creating interfaces...”)
```

```
%}

%for i = 1, num do
interface ethernet {{i}}
%end
```

68.2 Пример скрипта для группового управления интерфейсами

Ниже приводится скрипт for-iface, который позволяет выполнить команду для всех интерфейсов заданного типа, которые определены в running-config или для диапазона интерфейсов из running-config.

Пример использования:

```
! Выключить все интерфейсы ethernet в running—config
adm@DionisNX# @ for—iface ethernet disable

! Выключить среди них только те, которые попадают в заданный диапазон
adm@DionisNX# @ for—iface ethernet disable 0 3
```

Скрипт for-iface:

```
%{
if #args < 2 then
    inf "Make group command to iface(s)"
    inf "Usage: <iface type> <command> [start end]"
    exit(1)
end

s = tonumber(arg3) or 0
e = tonumber(arg4) or 1000000

c = config("interface "..arg1.." ")
%}

%{
for k, v in ipairs(c) do —— all ifaces
    cur = v:gsub("^interface "..arg1.." ([0—9]+).*$", "%1")
    cur = tonumber(cur) or —1
    if cur >= s and cur <= e then
        inf (v) —— show founded iface
%}
{{v}}
{{arg2}}
%{
    end
```

```
end
%}
```

69. Приложение

69.1 Примеры конфигураций

69.1.1 Конфигурация по-умолчанию

Конфигурация по-умолчанию содержит следующие настройки:

- Временная зона соответствует Москве;
- Имя хоста задано как DionisNX;
- Включены настройки TCP/IP-стека по умолчанию;
- Запрещена маршрутизация некорректных пакетов;
- Настроен один интерфейс со статическим адресом 192.168.1.1/24;
- Сервис протоколирования настроен по умолчанию;
- Включен сервис SSH для оператора cli;
- Включена маршрутизация пакетов.

```
!
timezone MSK—3
!
hostname DionisNX
!
ip path—mtu—discovery
ip tcp ecn server—mode
ip tcp selective—ack
ip tcp syncookies
ip tcp timestamps
ip tcp window—scaling
!
ip access—group no—invalid forward
ip access—list no—invalid
 deny state invalid
!
interface ethernet 0
 ip address 192.168.1.1/24
 enable
!
service log
 log
 alert beep
 size 262144 131072
 trace all acl
!
```

```
controller serial 0
 listen
!
service ssh
 enable
!
ip forwarding
!
! $CRC: d360aed1
```

69.1.2 Пример файерволла

```
!
timezone MSK—4
session timeout adm none
!
hostname raul
ip resolver domain cuba.int
ip resolver nameserver 192.168.33.254
!
ip path—mtu—discovery
ip tcp ecn server—mode
ip tcp selective—ack
ip tcp syncookies
ip tcp timestamps
!
ip access—list in16
 deny tcp dst 192.168.33.254 dport 22
 permit dst 192.168.33.0/24
 permit dst 192.168.32.0/24
 permit dst 192.168.16.0/24
 deny
!
ip access—list int33
 deny tcp dport 3127 dst 192.168.33.254
!
ip access—list outside
 permit dst 195.220.32.68 tcp dport 22 syn
 deny tcp syn
 permit state established
 permit state invalid
 permit state related
 deny
!
ip nat—list masq
```

```
 nat tcp dport 22 dnat ip 192.168.33.160 port 22
 nat src 192.168.33.0/24 snat ip 195.220.32.68
!
ip nat—list masq16
 nat src 192.168.33.0/24 masquerade
!
ip nat—list squid
 exclude in tcp dport 80 dst 192.168.33.254
 nat tcp dport 80 src 192.168.33.0/24 redirect port 3127
!
ip nat—list squid—test
 nat tcp dport 80 src 192.168.33.22/32 redirect port 3127
!
interface ethernet 0
 ip address 195.220.32.68/27
 ip access—group outside in
 ip nat—group masq
 enable
!
interface ethernet 1
 ip address 192.168.16.58/24
 ip access—group in16 in
 enable
!
interface ethernet 2
 ip address 192.168.33.254/24
 ip access—group int33 in
 ip nat—group squid
 enable
!
ip route 0.0.0.0/0 195.220.32.65
ip route 192.168.0.0/24 192.168.16.1
ip route 192.168.32.0/24 192.168.16.1
!
service log
 log
 alert beep
 size 262144 131072
 trace
!
service dns
 acl cuba 192.168.33.0/24
 acl net16 192.168.16.0/24
 acl net32 192.168.32.0/24
 acl nets net16 net32 cuba localips
 log all info
 allow query nets
```

```
 allow query—cache nets
 allow recursion nets
 allow transfer none
 limit cache—size 10000000
 limit journal—size 10000000
 listen localips
 notify no
 view default
  zone .
   auto weekly
  zone forward 16.168.192.in—addr.arpa.
   forwarders 192.168.16.3 192.168.16.4
  zone forward bubblegum.int.
   forwarders 192.168.16.3 192.168.16.4
  zone forward bubblegum.net.
   forwarders 192.168.16.3 192.168.16.4 192.168.16.1
  zone forward bubblegum.ru.
   forwarders 192.168.16.3 192.168.16.4 192.168.16.1
  zone master 33.168.192.in—addr.arpa.
   ttl  604800
   soa master raul.cuba.int. admin root@raul.cuba.int. refresh 604800 retry 86400 expire 2419200 negttl
         604800
   ns raul.cuba.int.
   ptr 1 fidel.cuba.int.
   ptr 160 havana.cuba.int.
   ptr 254 raul.cuba.int.
   ptr 3 pkunistan.cuba.int.
   ptr 6 vova—ipsec.cuba.int.
   update
  zone master cuba.int.
   ttl  604800
   soa master raul admin root@raul.cuba.int negttl 604000
   a 192.168.33.1 fidel
   a 192.168.33.160 havana
   a 192.168.33.254 raul
   a 192.168.33.3 pkunistan
   a 192.168.33.6 vova—ipsec
   ns raul
   update
 enable
!
service dhcp
 listen ethernet 2
 broadcast—address 192.168.33.255
 default—lease—time 345600
 domain—name cuba.int
 domain—search cuba.int
```

```
 domain—search bubblegum.int
 gateway 192.168.33.254
 max—lease—time 400000
 min—lease—time 86400
 subnet—mask 255.255.255.0
 name—server 192.168.33.254
 host fidel
  ip 192.168.33.1
  mac 00:22:b0:51:16:37
 host libcode
  ip 192.168.33.2
  mac 00:10:f3:04:18:63
 host peter—fix
  ip 192.168.33.4
  mac f4:6d:04:72:0d:01
 host pkunistan
  ip 192.168.33.3
  mac c8:60:00:61:41:77
 host vova—ipsec
  ip 192.168.33.6
  mac 08:00:27:3f:fd:70
 host white
  ip 192.168.33.7
  mac 00:1b:21:0d:c1:a6
 host xos
  ip 192.168.33.5
  mac e0:cb:4e:62:29:16
 subnet 192.168.33.0/24
  range 192.168.33.10 192.168.33.220
 enable
!
service proxy
 admin—email admin@bubblegum.ru
 listen 192.168.33.254 3128 intercept
 acl bad1 dstdomain sex.ru
 acl lan dst 192.168.33.0/24
 acl net33 src 192.168.33.0/24
 acl nolog srcdom—regex host.*
 acl nolog srcdom—regex pkunistan.*
 acl nolog srcdom—regex rdtsc.*
 acl u1 uri .u1
 log access acls !nolog
 log cache high
 cache limit disk max 500
 cache limit disk min 0
 cache limit memory max 16
 cache replacement—policy disk gdsf
```

```
 cache replacement—policy memory gdsf
 cache type aufs 16 4096
 refresh .\\.swf$ 10000 90% 20000
 refresh .\\.bmp$ 10000 90% 20000
 refresh .\\.png$ 10000 90% 20000
 refresh .\\.gif$ 10000 90% 20000
 refresh .\\.mpg$ 10000 90% 20000
 refresh .\\.avi$ 10000 90% 20000
 caching deny lan
 caching permit all
 http—access deny bad1
 http—access permit net33
 http—access deny all
 enable
!
service ntp
 listen 192.168.33.254
 server 0.ru.pool.ntp.org
 server 1.ru.pool.ntp.org
 server 2.ru.pool.ntp.org
 server 3.ru.pool.ntp.org
!
service ssh
 listen 192.168.33.254 22
 permit—adm—login
 enable
!
ip forwarding
```

69.1.3 Пример использования VRF-Lite совместно с OSPF

Ниже представлены основные настройки конфигурации узлов для Схемы 1 главы VRF данного руководства.

Пример настройки NET2_1:

```
interface ethernet 2
 enable
!
interface ethernet 2.1
 description ”NET2.1”
 ip address 100.0.0.99/24
 enable
!
router ospf
 network 100.0.0.0/16 area 0
!
```

Пример настройки NET2_2:

```
interface ethernet 2
 enable
!
interface ethernet 2.2
 description ”NET2.2”
 ip address 100.0.1.99/24
 enable
!
router ospf
 network 100.0.0.0/16 area 0
!
```

Пример конфигурации NX1:

```
interface ethernet 0
 description ”Link to NX2”
 ip address 10.0.12.1/24
 enable
!
interface ethernet 1
 enable
!
interface ethernet 1.1
 description ”To NET_1.1”
 ip address 192.168.0.1/24
 enable
!
interface ethernet 1.2
 description ”To NX2_vrf1”
 ip address 192.168.12.1/24
 enable
!
interface ethernet 2
 enable
!
interface ethernet 2.1
 description ”To NET_2.1”
 ip address 100.0.0.1/24
 enable
!
interface ethernet 2.2
 description ”To NX2_vrf2”
 ip address 100.0.12.1/24
 enable
!
interface vrf 1
 slave ethernet 1.1
```

```
 slave ethernet 1.2
 enable
!
interface vrf 2
 slave ethernet 2.1
 slave ethernet 2.2
 enable
!
router ospf vrf 1
 network 192.168.0.0/16 area 0
!
router ospf vrf 2
 network 100.0.0.0/16 area 0
```

Пример конфигурации NX2:

```
interface ethernet 0
 description ”Link to NX1”
 ip address 10.0.12.2/24
 enable
!
interface ethernet 1
 enable
!
interface ethernet 1.1
 description ”To NX3_vrf1”
 ip address 192.168.23.2/24
 enable
!
interface ethernet 1.2
 description ”To NX1_vrf1”
 ip address 192.168.12.2/24
 enable
!
interface ethernet 2
 enable
!
interface ethernet 2.1
 description ”To NX3_vrf2”
 ip address 100.0.23.2/24
 enable
!
interface ethernet 2.2
 description ”To NX_vrf2”
 ip address 100.0.12.2/24
 enable
!
interface ethernet 3
```

```
 description ”Link to NX3”
 ip address 10.0.23.2/24
 enable
!
interface vrf 1
 slave ethernet 1.1
 slave ethernet 1.2
 enable
!
interface vrf 2
 slave ethernet 2.1
 slave ethernet 2.2
 enable
!
router ospf vrf 1
 network 192.168.0.0/16 area 0
!
router ospf vrf 2
 network 100.0.0.0/16 area 0
!
```

Пример конфигурации NX3:

```
interface ethernet 0
 description ”Link to NX2”
 ip address 10.0.23.3/24
 enable
!
interface ethernet 1
 enable
!
interface ethernet 1.1
 description ”To NX2_vrf1”
 ip address 192.168.23.3/24
 enable
!
interface ethernet 1.2
 description ”To NET_1.2”
 ip address 192.168.1.1/24
 enable
!
interface ethernet 2
 enable
!
interface ethernet 2.1
 description ”To NX2_vrf2”
 ip address 100.0.23.3/24
 enable
```

```
!
interface ethernet 2.2
 description ”To NET_2.2”
 ip address 100.0.1.1/24
 enable
!
interface ethernet 3
 ip address dhcp
 enable
!
interface vrf 1
 slave ethernet 1.1
 slave ethernet 1.2
 enable
!
interface vrf 2
 slave ethernet 2.1
 slave ethernet 2.2
 enable
!
router ospf vrf 1
 network 192.168.0.0/16 area 0
!
router ospf vrf 2
 network 100.0.0.0/16 area 0
!
```

69.1.4 Пример реализации MPLS-L3VPN

Ниже представлены основные настройки конфигурации узлов для Схемы 2 главы VRF данного руководства.

Основная конфигурация СЗРО-1:

```
interface ethernet 0
 description ”to r1”
 ip address 192.168.1.1/24
 enable
!
interface ethernet 3
!
interface loopback 0
 ip address 99.0.0.1/32
 enable
!
router bgp 5227
```

```
 bgp router—id 99.0.0.1
 neighbor 192.168.1.2 remote—as 5227
 neighbor 192.168.1.2 update—source 192.168.1.1
 !
 address—family ipv4 unicast
  network 5.1.0.0/24 route—map rm—nh
  network 99.0.0.1/32
 exit
!
router access—list al—any permit any
!
router route—map rm—nh permit 10
 match ip address al—any
 set ip next—hop 99.0.0.1
!
```

Основная конфигурация C3PO-2:

```
interface ethernet 0
 description ”to r4”
 ip address 192.168.2.2/24
 enable
!
interface loopback 0
 ip address 99.0.0.4/32
 enable
!
router bgp 5228
 bgp router—id 99.0.0.4
 neighbor 192.168.2.1 remote—as 5228
 neighbor 192.168.2.1 update—source 192.168.2.2
 !
 address—family ipv4 unicast
  network 5.4.2.0/24 route—map rm—nh
  network 99.0.0.4/32
 exit
!
router access—list al—any permit any
!
router route—map rm—nh permit 10
 match ip address al—any
 set ip next—hop 99.0.0.4
!
```

Основная конфигурация R2D2-1:

```
interface ethernet 0
 description ”to r1”
 ip address 192.168.3.2/24
```

```
 enable
!
interface ethernet 3
!
interface loopback 0
 ip address 99.0.0.2/32
 enable
!
router bgp 5225
 bgp router—id 99.0.0.2
 neighbor 192.168.3.1 remote—as 5225
 neighbor 192.168.3.1 update—source 192.168.3.2
 !
 address—family ipv4 unicast
  network 5.1.0.0/24 route—map rm—nh
  network 99.0.0.2/32
 exit
!
router access—list al—any permit any
!
router route—map rm—nh permit 10
 match ip address al—any
 set ip next—hop 99.0.0.2
 set metric 98
!
```

Основная конфигурация R2D2-2:

```
interface ethernet 0
 description ”to r4”
 ip address 192.168.4.2/24
 enable
!
interface loopback 0
 ip address 99.0.0.3/32
 enable
!
router bgp 5224
 bgp router—id 99.0.0.3
 neighbor 192.168.4.1 remote—as 5224
 neighbor 192.168.4.1 update—source 192.168.4.2
 !
 address—family ipv4 unicast
  network 5.1.3.0/24 route—map rm—nh
  network 99.0.0.3/32
 exit
!
router access—list al—any permit any
```

```
!
router route—map rm—nh permit 10
 match ip address al—any
 set ip next—hop 99.0.0.3
!
```

Основная конфигурация r1:

```
mpls ip
!
interface ethernet 0
 description ”to C3PO—1”
 ip address 192.168.1.2/24
 mpls ip
 enable
!
interface ethernet 1
 description ”to r2”
 ip address 10.0.1.1/24
 mpls ip
 enable
!
interface ethernet 2
 description ”to R2D2—1”
 ip address 192.168.3.1/24
 mpls ip
 enable
!
interface loopback 0
 ip address 1.1.1.1/32
 mpls ip
 enable
!
interface vrf 1
 slave ethernet 0
 mpls ip
 enable
!
interface vrf 2
 slave ethernet 2
 mpls ip
 enable
!
router bgp 5226
 bgp router—id 1.1.1.1
 bgp cluster—id 1.1.1.1
 neighbor 2.2.2.2 remote—as 5226
 neighbor 2.2.2.2 update—source 1.1.1.1
```

```
 !
 address—family ipv4 unicast
  no neighbor 2.2.2.2 activate
 exit
 !
 address—family ipv4 vpn
  neighbor 2.2.2.2 activate
 exit
!
router bgp 5227 vrf 1
 bgp router—id 192.168.1.1
 neighbor 192.168.1.1 remote—as 5227
 neighbor 192.168.1.1 update—source 192.168.1.2
 !
 address—family ipv4 unicast
  neighbor 192.168.1.1 next—hop—self
  label vpn export 527
  rd vpn export 5227:1
  rt vpn both 52:27
  export vpn
  import vpn
 exit
!
router bgp 5225 vrf 2
 bgp router—id 192.168.3.1
 neighbor 192.168.3.2 remote—as 5225
 neighbor 192.168.3.2 update—source 192.168.3.1
 !
 address—family ipv4 unicast
  neighbor 192.168.3.2 next—hop—self
  label vpn export 525
  rd vpn export 5225:1
  rt vpn both 52:25
  export vpn
  import vpn
 exit
!
router ospf
 ospf router—id 1.1.1.1
 redistribute static
 network 0.0.0.0/4 area 0
!
router ldp
 router—id 1.1.1.1
 !
 address—family ipv4
  discovery transport—address 1.1.1.1
```

```
  !
  iface ethernet 1
  exit
  !
 exit
 !
!
```

Основная конфигурация r2:

```
mpls ip
!
interface ethernet 0
 description ”to r1”
 ip address 10.0.1.2/24
 mpls ip
 enable
!
interface ethernet 1
 description ”to r4”
 ip address 10.0.2.2/24
 mpls ip
 enable
!
interface ethernet 2
!
interface ethernet 3
!
interface loopback 0
 ip address 2.2.2.2/32
 mpls ip
 enable
!
router bgp 5226
 bgp router—id 2.2.2.2
 bgp cluster—id 2.2.2.2
 neighbor 1.1.1.1 remote—as 5226
 neighbor 1.1.1.1 update—source 2.2.2.2
 neighbor 4.4.4.4 remote—as 5226
 neighbor 4.4.4.4 update—source 2.2.2.2
 !
 address—family ipv4 unicast
  no neighbor 1.1.1.1 activate
  no neighbor 4.4.4.4 activate
 exit
 !
 address—family ipv4 vpn
  neighbor 1.1.1.1 activate
```

```
  neighbor 1.1.1.1 route—reflector—client
  neighbor 4.4.4.4 activate
  neighbor 4.4.4.4 route—reflector—client
 exit
!
router ospf
 ospf router—id 2.2.2.2
 network 0.0.0.0/0 area 0
!
router ldp
 !
 address—family ipv4
  discovery transport—address 2.2.2.2
  !
  iface ethernet 0
  exit
  !
  iface ethernet 1
  exit
  !
 exit
 !
!
```

Основная конфигурация r4:

```
mpls ip
!
interface ethernet 0
 description ”to r2”
 ip address 10.0.2.4/24
 mpls ip
 enable
!
interface ethernet 2
 description ”to C3PO—2”
 ip address 192.168.2.1/24
 mpls ip
 enable
!
interface ethernet 3
 description ”to R2D2—2”
 ip address 192.168.4.1/24
 mpls ip
 enable
!
interface ethernet 4
!
```

```
interface loopback 0
 ip address 4.4.4.4/32
 mpls ip
 enable
!
interface vrf 1
 slave ethernet 3
 mpls ip
 enable
!
interface vrf 2
 slave ethernet 2
 mpls ip
 enable
!
router bgp 5226
 bgp route—map delay—timer 1
 bgp router—id 4.4.4.4
 bgp cluster—id 4.4.4.4
 timers bgp 2 6
 neighbor 2.2.2.2 remote—as 5226
 neighbor 2.2.2.2 update—source 4.4.4.4
 !
 address—family ipv4 unicast
  no neighbor 2.2.2.2 activate
 exit
 !
 address—family ipv4 vpn
  neighbor 2.2.2.2 activate
 exit
!
router bgp 5228 vrf 2
 bgp router—id 192.168.2.1
 timers bgp 2 6
 neighbor 192.168.2.2 remote—as 5228
 neighbor 192.168.2.2 update—source 192.168.2.1
 !
 address—family ipv4 unicast
  neighbor 192.168.2.2 next—hop—self
  label vpn export 528
  rd vpn export 5228:1
  rt vpn both 52:27
  export vpn
  import vpn
 exit
!
router bgp 5224 vrf 1
```

```
 bgp router—id 192.168.4.1
 neighbor 192.168.4.2 remote—as 5224
 neighbor 192.168.4.2 update—source 192.168.4.1
 !
 address—family ipv4 unicast
  neighbor 192.168.4.2 next—hop—self
  label vpn export 524
  rd vpn export 5224:1
  rt vpn both 52:25
  export vpn
  import vpn
 exit
!
router ospf
 ospf router—id 4.4.4.4
 redistribute static
 network 0.0.0.0/4 area 0
!
router ldp
 router—id 4.4.4.4
 !
 address—family ipv4
  discovery transport—address 4.4.4.4
  !
  iface ethernet 0
  exit
  !
 exit
 !
!
```

Лист регистрации изменений									
Изм.	Номера листов (страниц)				Всего листов (страниц) в докум.	№ документа	Входящий № сопрово-дительного докум. и дата	Подп.	Дата
	измененных	замененных	новых	аннулиро-ванных					
1		все			690	НКБГ.632-24			12.01.24