Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата
741	28.8.15			

Руководство по настройке ПО на базе операционной системы (программной оболочки) Dionis NX C 1.2-10 Hand UTM

2015

ОГЛАВЛЕНИЕ

Стр.

1. Общие сведения

1.1 Функциональные возможности

Система представляет собой программно-аппаратный комплекс, включающий в себя маршрутизатор и управляющую его работой Dionis NX C 1.2-10 UTM. Dionis NX C 1.2-10 UTM может управлять ши- роким спектром маршрутизаторов. Характеристики используемых при работе маршрутизаторов в документе не конкретизируются. В настоящем документе приводится описание Dionis NX C 1.2-10 UTM. Dionis NX C 1.2-10 UTM построена на базе ядра ОС Linux 3.10.xx и обладает следующими функциональными возможностями:

- Поддержка статической маршрутизации TCP/IP (v4);
- Поддержка расширенной статической маршрутизации (ip policy route);
- Поддержка динамической маршрутизации (протоколы OSPF, BGP, RIP);
- Поддержка интерфейсов Ethernet;
- Поддержка возможности задания статических ARP-записей;
- Поддержка фильтрации пакетов на основе различных критериев 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;
- Поддержка различных вариантов трансляции IP-адресов (SNAT/DNAT)- обеспечение сокрытия внутренней структуры локальных вычислительных сетей;
- Поддержка криптографической защиты данных, передаваемых по каналам связи сетей общего пользования, использующих протоколы семейства TCP/IP (компоненты СКЗИ):
  - создание и поддержка статических криптотуннелей между узлами с шифрованием и имитозащитой передаваемых IP-пакетов с инкапсуляцией их в протокол “IP в IP”;
  - реализация протоколов IPSEC ГОСТ (IKEv1, ESP), позволяющая создавать статические и динамические туннели IPSEC между узлами ;
- Поддержка протоколов динамической маршрутизации (OSPF, BGP, RIP);
- Поддержка протоколов групповой передачи (Multicasting): IGMP (протокол управления группами Интернет) и DVMRP (дистанционно-векторный протокол многоадресной маршрутизации);
- Поддержка механизмов качества обслуживания (QoS);
- Поддержка виртуальных локальных компьютерных сетей (VLAN);
- Поддержка аггрегации интерфейсов (bonding)- объединения нескольких интерфейсов в один виртуальный для повышения пропускной способности, резервирования интерфейса и т.д.;

- Поддержка инкапсуляции IP-пакетов в туннели GRE;
- 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 протоколирования событий фильтрации IP-пакетов;
- Поддержка протоколирования цикла обработки IP-пакетов при прохождении их через маршрутизатор;
- Поддержка механизма контроля целостности программных компонентов комплекса;
- Поддержка процедур резервного архивирования и восстановления;
- Поддержка функционирования узлов Dionis NX C 1.2-10 UTM в режиме отказоустойчивого кластера.

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 Dionis NX C 1.2-10 UTM, для повышения надежности могут дублироваться . Кроме стандартного способа дублирования, основанного на применении протокола VRRP (п. 44),Dionis NX C 1.2-10 UTM представляет возможность аппаратного дублирования маршрутизаторов (создание отказоустойчивого кластера, п. 45). 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 и составляет не менее 10000. Это количество также может быть ограничено используемой лицензией.

2. Установка с флеш-диска

Установка Dionis NX C 1.2-10 UTM производится с помощью загрузочного инсталяционного флеш-диска. Этот диск связан с конкретным аппаратным комплексом, поставляется вместе с ним и является его неотъемлемой частью. С него производится начальная установка

(как правило, она производится производителем оборудования), сохранение конфигурации, восстановление, запись новых версий и т.д. В случае утери такого диска клиент имеет возможность получения образа диска (исходный вариант без дополнительно сохраненной пользователем на диске информации, бэкапов и сохраненных конфигураций). Далее, снова создав загрузочный диск, пользователь получает возможность опять производить на него необходимые системные действия. Ни для какой другой Системы, кроме той, для которой он предназначен, инсталяционый диск использован быть не может. С другой стороны, восстановление утерянных, но поставленных ранее на данный экземпляр Системы версий Dionis NX C 1.2-10 UTM (одной, нескольких или всех установленных когда-либо на данный экземпляр Систему), может производиться любое количе- ство раз без ограничений.

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 хранятся сжатые образы системы. Файлы со сжатыми образами имеют вид dionisnx-<версия>.x86_64.dip.

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

Несмотря на то, что, как правило, начальная установка Dionis NX C 1.2-10 UTM выполняется произво- дителем оборудования, возможны специальные ситуация, когда такая установка выполняется пользователем, возможно, при отсутствии связи с Интернет. В этом случае пользователь должен иметь полностью сформированный установочный флеш-диск с требуемой версией, полученный от производителя до начала установки. Для установки системы 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 любого внешнего носителя.

После загрузки с установочного флеш-диска на экран будет выведен список возможных дей- ствий (для управления используется псевдо-графический интерфейс):

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

После установки системы необходимо перевести плату “Сторож” обратно в рабочий режим (режим JL).

Пункт “Установка системы” позволяет установить систему на жесткий диск. На одном жестком диске может быть установлено сразу несколько экземпляров системы, однако только один из установленных экземпляров будет активным в данный момент времени. Под обнов- лением системы подразумевается установка новой версии ОС на жесткий диск маршрутизатора. Вновь установленная версия автоматически становится активной. Если старая версия системы больше не требуется, ее можно впоследствии удалить с жесткого диска.

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

После выполнения всех описанных действий, будет произведена установка выбранной системы 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:

```
DionisNX> _
```

или

```
DionisNX# _
```

Здесь DionisNX - имя узла

Команды 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 C 1.2-10 Hand UTM разделены на группы по функциональному назначению. Каждая группа может содержать подгруппы (группы следующих уровней). В соответствии с этим имя команды может быть составным и 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 всегда производятся от имени какой-либо учетной записи (п. 6). 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 (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. Порядок смены пароля учетной записи описан в п. 6.3.3.

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>) выводится информация системы мониторинга (п. 11, Журналирование и отладка).

3.3 Пространства имен

Система может работать с файлами, расположенными:

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

Файловое пространство системыразделено на несколько непересекающихся пространств:

- Локальное дисковое пространство файлов работающей версии ОС (слот данных системы);

- Дисковое пространство файлов маршрутизатора, разделяемое между всеми версиями ОС, 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 хранилища приведен в разделе п. 46.2.

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 ОС (префикс, используемый по умолчанию, если он не указан)
share:	пространство файлов маршрутизатора, разделяемое между всеми версиями ОС, установленными в маршрутизаторе
log:	файловое пространство журналов
ftp:	файлы, доступные по протоколу FTP
http:	файлы, доступные по протоколу HTTP

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 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 (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 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 ”	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 выполнилась удачно, то она влияет на состояние системы и на текущую конфигурацию. В противном случае команда не влияет ни на состояние системы, ни на текущую конфигурацию.

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
enable
ip address 192.168.56.3/24
(config—if—ethernet0)# ip secondary—address 10.10.10.10/24
(config—if—ethernet0)# do show
enable
ip address 192.168.56.3/24
ip secondary—address 10.10.10.10/24
(config—if—ethernet0)# ip secondary—address 20.20.20.20/24
(config—if—ethernet0)# do show
enable
ip address 192.168.56.3/24
ip secondary—address 10.10.10.10/24
ip secondary—address 20.20.20.20/24
!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 10.10.10.10/24
DionisNX(config—if—ethernet0)# do show
```

```
enable
ip address 192.168.56.3/24
ip secondary—address 20.20.20.20/24
(config—if—ethernet0)# no ip secondary—address 20.20.20.20/24
(config—if—ethernet0)# do show
enable
ip address 192.168.56.3/24
```

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
enable
ip address 4.4.4.4/24

! — просматриваем всю конфигурацию
(config—if—ethernet3)# exit
(config)# do show
!
hostname DionisNX
!
interface ethernet 0
 enable
 ip address 1.1.1.1/24
!
interface ethernet 1
 enable
 ip address 2.2.2.2/24
!
interface ethernet 2
 enable
 ip address 3.3.3.3/24
!
interface ethernet 3
 enable
```

```
 ip address 4.4.4.4/24
!
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 двух пространств файловой системы, например, такой:

```
DionisNX# ls /
flash0:
cdrom0:
 file :
log:
```

Файлы на FTP- и HTTP-серверах (доступные по соответствующему протоколу) не показываются в списке по команде “ls /”.

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
```

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 копировании по протоколу SSH (п. 28) 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
mkdir <что>	создание каталога
less <что>	просмотр содержимого файла (с возможностью прокрутки вверх-вниз)
cat <что>	вывод содержимого файла на экран (без возможности прокрутки)

Например:

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

4. Выключение и перезагрузка

Чтобы выключить узел нужно выполнить команду привилегированного режима:

```
# poweroff
```

Нажатие кнопки выключения на корпусе эквивалентно данной команде.

Для перезагрузки узла 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).

5. Предварительная настройка

5.1 Начало работы (Установка и смена паролей)

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

5.2 Общие настройки

Предварительная настройка узла 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:

```
(config)# timezone MSK—3
```

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 задан часовой пояс MSK-3.

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

```
# clock 13:58 31 12 2001
```

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.

5.3 Интерфейсы

В данном разделе рассматривается базовая настройка сетевых интерфейсов Ethernet.

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 IP-адреса/маски подсети):

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

Если интерфейс будет настраиваться с использованием службы DHCP (п. 24), 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 domainlist.

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 интефейсу 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. Пример:

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

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

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
```

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
```

Внимание! Если интерфейс, на который выполняется дублирование, будет удален или отключен без предварительного отключения режима дублирования, весь входящий трафик на интерфейсе с настроенным дублированием будет сброшен! Чтобы предотвратить это, необходимо снова включить интерфейс, или отключить режим дублирования с помощью команды no mirror.

5.4 Статическая маршрутизация

Для того, чтобы узел мог выполнять функции 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).

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

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

В системе 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 ;
- bgp, rip, ospf - маршруты, создаваемые соответствующими службами динамической маршрутизации.

Для добавления статических 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 повышения производительности ядро системы кэширует часто используемые маршруты. Иногда возникает необходимость очистить данный кэш, чтобы новые правила маршрутизации вступили в силу немедленно.

Просмотреть содержимое данного кэша можно с помощью команды привилегированного режима:

```
# show ip route cache
```

Следующая команда очищает кэш маршрутизации:

```
# clear ip route cache
```

5.5 Клиент DNS

Система 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. 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.

5.5.1 Связь с DHCP

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

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

Если необходимо использовать эти команды и применить службу DHCP (п. 24) 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.

Чтобы настроить клиент DNS войдите в режим configure.

5.5.2 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)# 1 ip resolver nameserver 10.0.0.2
(config)#
(config)# ip resolver host  10.0.0.3 zeta.int zeta—alias.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): 10.0.0.2 10.0.0.1;
- список статической привязки IP-адресов к именам: имена zeta.int zeta-alias.int имеют адрес 10.0.0.3.

5.5.3 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.

5.6 Просмотр и настройка ARP-таблицы

Чтобы вывести текущую таблицу соответствия IP- 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 поддерживает таблицы с информацией о хостах находящих- ся 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 суще- ствует 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).

5.7 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.

5.7.1 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.

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.

5.7.2 Настройки протокола TCP

5.7.2.1 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
```

5.7.2.2 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.

5.7.3 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
```

5.7.4 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
```

5.8 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).

5.8.1 Утилита ping

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

5.8.2 Утилита traceroute

Позволяет отследить маршрут, по которому движется пакет пробы.

5.8.3 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. Alt-F1 - выключает показ монитора. 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.

5.8.4 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.

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

Учетная запись консольного доступа позволяет просматривать некоторые параметры системы и часть информации о ее состоянии. Работа под учетной записью консольного доступа 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 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 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
```

В данном примере использованы три 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”.

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”
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”
 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.

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.

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 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
```

6.3.9 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. 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.

7. Ролевая модель

7.1 Права доступа учетной записи администратора

Если администратор не имеет статуса супервизора, то его права доступа к 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 роли, 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 появлении 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 данных;
- 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.show	Получение информации о ключах
@ike.conf	Конфигурация службы и туннелей IKE (п. 43.5)
@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

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. 5.8), whois, arping, nslookup (п. 23.8)

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
@wifi.conf	Конфигурирование WiFi
@wifi.bind	Привязка номера интерфейса WiFi к контроллеру
@wifi.show	Получение информации о WiFi
@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 траффиком

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. 8.3.4)
@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

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
@pautina.server -	Настройка системы pautina. 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

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

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.show	Получение информации OSPF
@bgp.conf	Конфигурирование BGP
@bgp.show	Получение информации BGP
@rip.conf	Конфигурирование RIP
@rip.show	Получение информации RIP

7.2.8.3 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
@igmp.conf	Конфигурирование мультикаст-маршрутизации IGMP
@pim.conf	Конфигурирование мультикаст-маршрутизации 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	
@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

Имена полномочий	Описание
@dip.show	Получение информации о DIP-пакетах
@data.oper	Действия со слотами данных, командами над пакетами ОС и слотами данных (restore, os bind, schedule rebind, schedule restore и т.д.) (п. 46.2)
@data.show	Получение информации о слотах данных
@backup.oper	Команды создания резервной копии данных (os data backup), безопасной резервной копии (schedule backup) (п. 46.2)
@backup.show	Получение информации о резервных копиях
@boot.oper	Команды загрузки системы boot default, boot fallback, boot experemental, и команда миграции на другой пакет ОС schedule migrate (п. 46.2)
@boot.show	Получение информации о параметрах загрузки
@cluster.conf	Конфигурирование кластера (п. 45)
@cluster.oper	Действия с кластером (кроме конфигурирования)
@cluster.show	Получение информации о кластере
@host.conf	Общие настройки (времени, часового пояса, имени узла) (п. 5.2), принудительной проверки файловой системы на ошибки командой schedule fsck (п. 47.2)
@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

Имена полномочий	Описание
@file.oper	Право выполнения файловых операций
@file.net	Право выполнения сетевых операций с файлами
@removable.oper	Право выполнения операций с внешними устройствами (дисками)
@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. 8.3.6)

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 указанным именем 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 отмены таких зависимостей или создания новых зависимостей, а также команды возврата всех зависмостей 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 capibility <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 capibility @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 capibility" задано значение "*", то будут показаны все полномочия системы и их зависимости. Команда"show capibility"может использоваться с параметром "delegate". В этом случае будут показаны только те полномочия, у которых есть зависимости. Например, команда:

```
DionisNX# show capibility * delegate
```

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

Команда:

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

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

Команда

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

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

8. Фильтрация

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 :

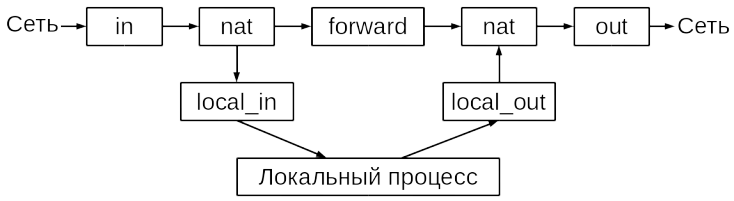

Рис. 8.1: Фильтрация

Название	Назначение
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

8.1 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
```

8.2 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.

8.2.1 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
```

8.2.2 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 .

Например:

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

8.3 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, далее будут описаны основные критерии.

8.3.1 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
```

8.3.2 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
```

8.3.3 Содержимое пакета

Существует возможность задания критерия по содержимому пакета. Для этого используется параметр content <правило>. Где <правило> записывается на специальном языке, и позволяет отслеживать содержимое байт/слов в пакете.

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”
```

8.3.4 Списки недавних пакетов

В системе 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

8.3.5 Состояние соединения

Ядро 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 established
DionisNX(config—acl—wan)# permit related
DionisNX(config—acl—wan)# deny
```

8.3.6 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
```

8.3.7 Другие критерии отбора

Синтаксис других критериев отбора описан в списке команд , 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
dscp	Значение DSCP
datestart,datestop,timestart,timestop,monthdays,weekdays	Время

8.3.8 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
```

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

8.3.9 Комментарии

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

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

8.4 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

9. Многоадресная передача

Система 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 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 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)).

Рассмотрим вкратце 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.

9.1 Общие сведения о настройке многоадресной маршрутизации

В системе имеется возможность настраивать:

- статическую ММ: командой ip mroute;
- динамическую ММ на основе IGMPv2: 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 только одна из четырех типов настроек.

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.

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

```
(config—dvmrp—ethernet0)# rate 100
```

Значение задается в Кбит/сек. Для снятия ограничения установите значение 0 (неограниченно) или удалите команду.

По умолчанию: неограниченно.

Если интерфейс получает МД из разных удаленных подсетей и необходимо,чтобы интерфейс обслуживал МП для этих подсетей, следует описать 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)# bound bo1
(config—dvmrp—ethernet0)# bound 239.255.2.0/24
```

Эти команды определяют для интерфейса ethernet0 две АГ 239.255.1.0/24 и 239.255.2.0/24. Любой трафик МП с адресами назначения, соответствующим указанным границам, 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.

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 , поддерживающий протокол PIM;
- 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.

Фаза 3. Выбор более оптимального маршрута.

1. когда DR получателя получает МД от отправителя, этот DR отправляет Join сообщение по направлению к отправителю;
2. когда DR отправителя получает указанное Join сообщение, этот DR начинает посылать МД напрямую к получателю;
3. таким образом формируется SPT для множества получателей;
4. когда МД приходят из SPT-маршрута на DR получателя или на общий для SPT и RPT маршрутизатор, то данный маршрутизатор начинает отбрасывать пакеты от RPT и посылает сообщения PIM Prune на RP, чтобы отсечь RPT-маршруты, т.к. уже сформирован путь к получателям в обход RP.

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. NHS - ближайший к источнику PIM-маршрутизатор регистрируется на RP (из п.1 он знает,какая RP отвечает за группу G);
4. RCV - получатель заявляет о желании получать трафик группы G: шлет IGMP сообщение Join (*,G); его получает ближайший к RCV PIM-маршрутизатор NHR;
5. NHR - ближайший к получателю PIM-маршрутизатор шлет PIM-сообщение Join(*,G) в сторону RP;
6. RP шлет PIM-сообщение Join(S,G) в строну NHS;
7. построены деревья: SPT - между SRC и RP ; RPT - между RP и RCV

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-preference <VAL>

Задает приоритет маршрутизатора в выборе выделенного маршрутизатора (DR). Чем ниже значение ,тем выше приоритет. По умолчанию: 101.

bsr-cand [LOCAL_IP] [PRIO]

Устанавливает параметры C-BSR. Данная система 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 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-команда.

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 <all|default>

Включает лог:

- default - лог IGMP- и PIM-сообщений
- all - самый подробный лог.

tree-switch-threshold [rate RATE] [interval TIME]

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

Параметры:

- RATE - порог скорости трафика (байт/сек);
- TIME - интервал проверки RATE.

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.

По умолчанию: rate: 6250 байт/сек; time: 20сек.

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)# preference 1
```

По умолчанию: значение указанное в default-preference. Приоритет влияет на выбор данного маршуртизатора как DR для данной сети LAN. Чем меньше данное значение, тем более вероятен выбор данного маршрутизатора как DR. При наличии параллельных проходов к источнику или RP для выбора маршрута применяются сообщения Assert. Используя сообщения Assert, адресованные 224.0.0.13 (группа all-pim-routers) в локальной сети, вышестоящий маршрутизатор может узнать, где осуществляется переадресация сообщений. Нижестоящие маршрутизаторы, получая сообщения Assert, узнают, какой маршрутизатор выбран в качестве DR, и куда следует посылать сообщение Join.

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 интерфейс должен обслуживать МП из разных удаленных подсетей, 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

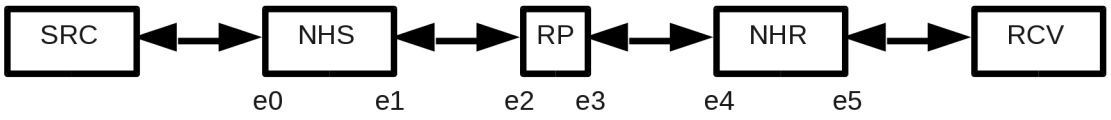

Рис. 9.2: Пример стенда PIM-SM

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 входящий интерфейс должен обслуживать МП из разных удаленных подсетей, следует описать 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).

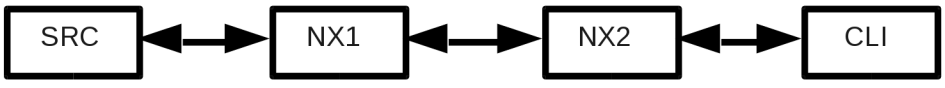

Рис. 9.3: Схема МП на основе IGMP

В данном примере:

- src - это источник МП, с адреса 10.0.0.0/24;
- cli - получатель МП;
- nx1,nx2 - маршрутизаторы ;
- 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 :

- 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 | igmp | pim |dvmrp [groups|cache] | status>

Параметры:

- log - протоколы, в которых регистрируется работа МП;
- staus - информация о статусе работы динамической и статической МП;
- routes - таблица многоадресных маршрутов (для динамической и статической МП);
- vifs - таблица VIF-ов: интерфейсов используемых в МП (для динамической и статической МП);
- igmp - IGMP-информация о МП;
- 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.

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.

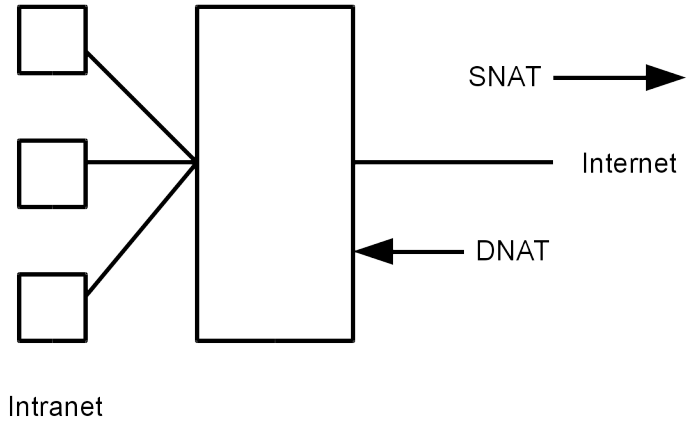

Рис. 10.1: Механизм NAT

В системе 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 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. Журналирование и отладка

В системе существует несколько механизмов, которые могут быть использованы администратором для диагностики проблем, а также для выявления нарушений.

11.1 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
имя или числовое значение протокола	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

Например:

```
DionisNX# tcpdump ethernet 0 numeric dump hex—ascii
```

Для прерывания режима мониторинга необходимо нажать на клавиатуре Ctrl-C.

11.2 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).

11.2.1 Создание списка трассировки

Для создания списка трассировки, 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
```

11.2.2 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.

Например:

```
DionisNX(config)# ip trace—group ping
```

Для отмена действия списка трассировки следует выполнить команду no ip trace-group <имя списка>:

```
DionisNX(config)# no ip trace—group ping
```

11.2.3 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
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

название	описание
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

11.3 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».

11.4 Системные журналы

Кроме журнала IP-пакетов, 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
dish	действия администратора

название	назначение
daemon	сообщения сервисов
kernel	сообщения ядра
router	сообщения от сервисов динамической маршрутизации
alert	сообщения, требующие внимания
auth	сообщения безопасности

Кроме этих системных журналов, существуют журналы для подсистем dhcp, dns, ntp, snmp. Просмотр журнала для них возможен с помощью команды show service <название сервиса> log [параметры] в enable режиме:

```
DionisNX# show service dns log
```

11.5 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 11.4), 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.

11.5.1 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.

11.5.2 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;
- * - 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.

11.6 Служба watcher

Существует возможность отслеживать интересующие администратора события в журналах и собирать их в отдельный журнал. Для этого необходимо настроить службу watcher. Вход в режим конфигурации службы выполняется командой:

```
Router(config)# service watcher
Router(config—service—watcher)#
```

Далее, настраиваются журналы за которыми необходимо осуществлять слежение. Для выбора журнала используйте команду watch <журнал>, например:

```
Router(config—service—watcher)# watch dish
Router(config—service—watcher—dish)#
```

При этом осуществляется переход в режим настройки правил слежения из выбранного журнала. Для удаления слежения за выбранным журналом используйте команду: no watch <имя журнала>. Каждое правило слежения состоит из команды (действия), строки сопоставления и дополнительных параметров.

Команда	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
mailer	Послать сообщение по почте с помощью службы mailer
notice	Послать сообщение в журнал watcher с уровнем notice
warning	Послать сообщение в журнал watcher с уровнем warning

В качестве строки сопоставления используется регулярное выражение, например:

```
Router(config—service—watcher—dish)# alert ”service watcher”
```

В данном случае, будет сформировано событие уровня alert при выполнении команды “service watcher”. Регулярные выражения могут содержать шаблоны, состоящие из символьных классов:

Символьный класс	Соответствие символам
x	x - соответствует сам себе. (Он не может быть равен ни одному из специальных символов ^$()%.*+-?
.	Любой символ
%x	Где x - любой не алфавитно-цифровой символ), соответствует сам себе. Это - стандартный способ экранировки специальных символов
[set]	Соответствует любому символу из набора, заданного в set. Диапазон символов может быть определен, с помощью символа ‘-’ отделяющего начало и конец диапазона.
[^set]	Отрицательный набор символов. Соответствует любому символу, кроме тех, что заданы в наборе set.

Шаблон	Описание
Одиночный символьный класс	Соответствует любому одиночному символу из заданного класса
’*’ за классом	Соответствует 0 или большему количеству повторений символов из заданного класса. Например: c* - символ с 0 или более раз.
‘+’ за классом	Соответствует 1 или большему количеству повторений символов из заданного класса. Эти элементы повторения будут всегда соответствовать самой длинной возможной последовательности.
‘-’ за классом	Соответствует 0 или большему количеству повторений символов из заданного класса. В отличие от *, элементы повторения будут всегда соответствовать самой короткой возможной последовательности
‘?’ за классом	Соответствует 0 или единственному вхождению символа из заданного класса;
^	Соответствует началу строки
$	Соответствует концу строки

Обратите внимание, что служебные символы требуют экранирования. Например:

```
Router(config—service—watcher—dish)# alert ”ip access%—list”
```

Символ ‘-’ здесь экранируется, так как он имеет специальный смысл.

Можно создавать несколько правил слежения. Для удаления правила используйте команду: no <номер правила>.

Обратите внимание, что правило “alert” удобно использовать для звукового и визуального оповещения о выбранных событиях.

Кроме правил, можно задать период слежения за журналом в секундах:

Router(config-service-watcher-dish)# period 1

Если период не задан, то по умолчанию период выбирается случайное значение периода от 3 до 5 секунд.

Для возвращения к настройке по умолчанию используйте: no period.

Для того, чтобы служба watcher стала активной, нужно выполнить команду enable из режима конфигурации службы:

```
Router(config—service—watcher)# enable
```

Для отключения службы, используйте: disable.

Для просмотра журналов службы используйте команду enable-режима:

```
Router# show service watcher log
```

Для того, чтобы полностью удалить конфигурацию службы, воспользуйтесь командой:

```
Router(config)# no service watcher
```

12. VLAN

Поддерживается стандарт IEEE 802.1Q (VLAN). Для создания интерфейса, выходящий трафик с которого будет помечаться идентификатором vlan-сети, а входящий трафик соответству- ющей 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.

13. WIFI-интерфейсы

13.1 Введение

Система имеет поддержку беспроводных интерфейсов и может работать как в режиме беспроводной точки доступа, так и в режиме беспроводного клиента.

Для работы с wifi-интерфейсом используется команда: interface wifi из режима configure.

```
adm@DionisNX(config)# interface wifi 0
adm@DionisNX(config—if—wifi0)#
```

13.2 Работа WIFI-интерфейса в режиме беспроводной точки доступа

Для перевода интерфейса в режим точки доступа необходимо выполнить следующую команду:

```
adm@DionisNX(config—if—wifi0)# mode master
```

Команды доступные для настройки интерфейса в режим точки доступа.

команда	параметр
ssid <Name>	Имя беспроводной сети
passphrase <Password>	Пароль беспроводной сети
channel <Num>	Номер канала беспроводной сети
hw_mode <a|b|g|ad>	Режим работы точки доступа
ignore-broadcast-ssid <1|2|0>	Скрывать SSID. (0 - параметр отключен, 1 - Передавать пустой SSID, 2 - Передавать пустой SSID, но сохронять длинну ssid)
max_num_sta <Num>	Максимальное количество подключаемых станций
wpa <WPA|WPA2|WPA/WPA2>	Настройка режима безопасности
wpa-key-mgmt PSK	Использование wpa-psk алгоритма для управления ключами
wpa-pairwise <TKIP|CCMP|TKIP/CCMP>	Установить алгоритм шифрования парных ключей для точки доступа
rsn-pairwise <TKIP|CCMP|TKIP/CCMP>	Установить алгоритм шифрования для WPA2 точки доступа

13.3 Работа WIFI-интерфейса в режиме беспроводного клиента

Для перевода интерфейса в режим клиента необходимо выполнить следующую команду:

```
adm@DionisNX(config—if—wifi0)# mode managed
```

Настройка интерфейса в режиме клиента сводится к настройке профилей подключения к беспроводной сети.

Для интерфейса доступно несколько профилей подключения. Интерфейс поочередно пытается установить соединение с заданной сетью из каждого профиля.

Для создания нового профиля или входа в текущий необходимо выполнить команду network <Имя>

```
adm@DionisNX(config—if—wifi0)# network net1
adm@DionisNX(config—if—wifi0—net1)#
```

Команды доступные для настройки профилей подключения.

команда	параметр
ssid <Name>	Имя беспроводной сети
passphrase <Password>	Пароль беспроводной сети
priority <Num>	Установить приоритет подключения
proto <WPA|WPA2|WPA/WPA2>	Допустимый протокол работы сети
key-mgmt <NONE|WPA-PSK|WPA-PSK NONE>	Установить алгоритм управления ключами протокола
pairwise <CCMP|TKIP|CCMP TKIP|NONE>	Установить попарный алгоритм шифрования для WPA
group <CCMP|TKIP|CCMP/TKIP>	Установить групповой (broadcast/multicast) алгоритм шифрования для WPA

13.4 Прочие команды, доступные для работы с WIFI-интерфейсом

Для просмотра информации о доступных беспроводных сетях необходимо выполнить команду:

```
adm@DionisNX(config—if—wifi0)# scan
```

Для просмотра только имен доступных беспроводных сетей необходимо выполнить команду:

```
adm@DionisNX(config—if—wifi0)# scan essid
```

14. MODEM-интерфейсы

14.1 Введение

Система имеет поддержку 3G/LTE USB модемов.

Для работы с modem-интерфейсом используется команда: interface modem из режима configure.

```
adm@DionisNX(config)# interface modem 0
adm@DionisNX(config—if—modem0)#
```

14.2 Команды доступные для настройки интерфейса

команда	параметр
apn <Name>	Имя точки доступа
user <UserName>	Имя пользователя для аутентификации
password <Password>	Пароль для аутентификации
phone <Num>	Номер дозвона
speed <Speed>	Скорость модема (необязательный параметр)
modem-backend <Port>	Номер порта модема

14.3 Номер порта модема (modem-backend)

Модем может иметь несколько портов. На данный момент в системе указание конкретного рабочего порта модема производится вручную командой modem- backend.

14.4 Пример настройти интерфейса

Пример настройки модема Huawei E3272 для megafon

```
adm@DionisNX(config)# interface modem 0
adm@DionisNX(config—if—modem0)#
adm@DionisNX(config—if—modem0)# apn internet
adm@DionisNX(config—if—modem0)# user gdata
adm@DionisNX(config—if—modem0)# password gdata
adm@DionisNX(config—if—modem0)# phone *99#
adm@DionisNX(config—if—modem0)# modem—backend USB0
```

15. Bonding-интерфейсы

Система 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. В маршрутизаторе может быть создано два таких интерфейса с номерами 0 и 1.

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;

15.1 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)

15.1.1 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. Если выходит из строя один из интерфейсов, пакеты отправляются на остальные оставшиеся. При подключении портов к разным коммутаторам необходимо выполнить их настройку.

15.1.2 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. Дополнительная настройка коммутатора не требуется.

Внимание: подпараметр fail-over-mac может быть изменен только в том случае, если в текущем bonding-интерфейсе нет подчиненных интерфейсов.

15.1.3 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.

15.1.4 broadcast

Все пакеты передаются на все интерфейсы в группе. Режим обеспечивает отказоустойчивость.

15.1.5 802.3ad

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.

15.1.6 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. Если принимающий интерфейс выходит из строя, то следующий занимает его место, приватизировав его mac-адрес.

15.1.7 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.

15.1.8 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.

15.2 Режимы мониторинга

Существует два режима мониторинга состояния: с помощью arp-проб и состояния линии mii. Если выбран режим arp, то требуется указать IP-адреса удаленных подчиненных интерфейсов.

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
```

16. Сетевые мосты

Система 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 поддерживает протокол STP (Spanning Tree Protocol), который ис- пользуется для того, чтобы избежать петель коммутации. Для настройки STP следует использо- вать следующие команды:

Команда	Описание
[no] stp	Включение/выключение протокола stp
[no] hello	Задание времени hello (stp)
[no] maxage	Задание времени maxage (stp)

Для просмотра информации о сетевом мосте используйте команду show bridge [<iface>], например:

```
DionisNX# show bridge 0
```

17. Интерфейсы E1

Интерфейсы E1 предназначены для передачи голоса или данных. Система 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 будет выглядеть как интерфейс.

Для работы с E1 в системе необходимо сначала задать параметры контроллера E1, а уже затем, на основании параметров контроллера, создавать и настраивать интерфейсы.

17.1 Настройка контроллера

Настройка контроллера E1 производится в режиме конфигурации.

```
DionisNX(config)# controller e1 1
DionisNX(config—e1—1)#
```

17.1.1 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 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
```

17.1.2 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
```

17.1.3 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
```

17.1.4 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
```

17.1.5 Состояние контроллера E1

Текущее состояние контроллера E1 можно узнать с помощью команды «show controller».

```
DionisNX# show controller e1 1
```

17.1.6 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

17.2 Настройка интерфейса

Для работы с E1 в системе 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.

17.2.1 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
```

17.2.2 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.

17.2.3 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»).

17.2.4 Состояние интерфейса

Текущее состояние HDLC интерфейса можно узнать с помощью команды «show interface».

```
DionisNX# show interface hdlc 1
```

17.2.5 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

18. GRE-туннели

Система 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

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
```

19. 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
```

20. VPN-туннели

20.1 Введение

Система имеет поддержку технологии OpenVPN для создания зашифрованных каналов типа точка-точка или сервер-клиенты между компьютерами.

Настройка OpenVPN в системе сводится к настройке динамических интерфейсов типа svpn (серверный интерфейс) и vpn (клиентский интерфейс).

20.2 Импорт, удаление и просмотр доступных ключей и сертификатов

Для использования vpn и svpn интерфейсов в режиме с TLS-аутентификацией или pre-shared ключом защиты необходимо предварительно импортировать соответствующие ключи и сертификаты. Для импорта ключей и сертификатов используется команда vpn import <тип объекта> <путь к файлу> из enable-режима.

```
adm@DionisNX# vpn import ca share:/crt/ca.crt
```

Для удаления импортированного ключа или сертификата vpn используется команда vpn clear <тип объекта> <Импортированный объект> из enable-режима.

```
adm@DionisNX# vpn clear ca ca.crt
```

Для просмотра списка импортированных ключей или сертификатов vpn используется команда vpn show <тип объекта> из enable-режима.

```
adm@DionisNX# vpn show show ca.crt
```

Команды доступные для работы с ключами и сертификатами vpn.

команда	параметр
vpn import ca <CA>	Импорт корневого сертификата
vpn import cert <Cert>	Импорт сертификата сервера или клиетна
vpn import cert-key <Cert-key>	Импорт ключа сертификата сервера или клиетна
vpn import dh-key <Dh>	Импорт ключа Диффи-Хеллмана
vpn import psk <Psk>	Импорт pre-shared ключа защиты
vpn import tls-key <Tls-auth>	Импорт tls-auth ключа
vpn clear ca <CA>	Удаление импортированного корневого сертификата
vpn clear cert <Cert>	Удаление импортированного сертификата сервера или клиетна
vpn clear cert-key <Cert-key>	Удаление импортированного ключа сертификата сервера или клиетна

команда	параметр
vpn clear dh-key <Dh>	Удаление импортированного ключа Диффи-Хеллмана
vpn clear psk <Psk>	Удаление импортированного pre-shared ключа защиты
vpn clear tls-key <Tls-auth>	Удаление импортированного tls-auth ключа
vpn clear all	Удаление всех импортированных объектов
vpn show ca	Отображение списка импортированных корневых сертификатов
vpn show cert	Отображение списка импортированных сертификатов сервера или клиетна
vpn show cert-key	Отображение списка импортированных ключей сертификатов сервера или клиетна
vpn show dh-key	Отображение списка импортированных ключей Диффи-Хеллмана
vpn show psk	Отображение списка импортированных pre-shared ключей защиты
vpn show tls-key	Отображение списка импортированных tls-auth ключей
vpn show all	Отображение списка всех импортированных объектов

20.3 VPN-интерфейс

Для создания vpn-интерфейса используется команда: interface vpn <номер> из режима configure. При этом номер интерфейса может начинаться с 0.

```
adm@DionisNX(config)# interface vpn 1
adm@DionisNX(config—if—vpn1)#
```

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 N>, а для удаления всех remote - команду <no remote all>:

```
adm@DionisNX(config—if—conn)# no remote 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>	Предварительно импортированный корневой сертификат. ca-файл должен быть такой же как на сервере
cert <Сертификат клиента>	Предварительно импортированный сертификат клиента
key <Ключ клиента>	Предварительно импортированный ключ сертификата клиента

команда	параметр
tls-auth <Дополнительный ключ>	Предварительно импортированный 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>	Предварительно импортированный 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. Для поддержки ГОСТ необходимо выполнить команду <cipher gost89>
auth <Алгоритм>	Алгоритм Аутентификации. Для поддержки ГОСТ необходимо выполнить команду <auth gost-mac>
ping <Интервал в секундах>	ping удаленного конца туннеля, если нет передачи 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). Для поддержки ГОСТ необходимо выполнить команду <tls-cipher GOST2001-GOST89-GOST89>

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—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
```

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 key.pem
```

Ключ должен быть указан как на локальном, так и на удаленном конце туннеля, храниться в секрете и передаваться по защищенному каналу. Ключ может быть получен с помощью openvpn.

openvpn –genkey –secret key.pem

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 ca.crt
adm@DionisNX(config—if—vpn1)# cert client.crt
adm@DionisNX(config—if—vpn1)# key 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—if—vpn1)# 1 connection
adm@DionisNX(config—if—conn)# remote 192.168.33.232
adm@DionisNX(config—if—conn)# exit
adm@DionisNX(config—if—vpn1)# tls—client
adm@DionisNX(config—if—vpn1)# pull
adm@DionisNX(config—if—vpn1)# ca ca.crt
adm@DionisNX(config—if—vpn1)# cert client.crt
adm@DionisNX(config—if—vpn1)# key client.key
```

20.4 SVPN-интерфейс

Для создания svpn-интерфейса используется команда: interface svpn <номер> из режима configure. При этом номер интерфейса может начинаться с 0.

```
adm@DionisNX(config)# interface svpn 0
adm@DionisNX(config—if—svpn0)#
```

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>	Предварительно импортированный корневой сертификат
cert <Сертификат сервера>	Предварительно импортированный сертификат сервера
key <Ключ сервера>	Предварительно импортированный ключ сертификата сервера
dh <Ключ Диффи-Хеллмана>	Предварительно импортированный ключ Диффи-Хеллмана
tls-auth <Дополнительный ключ>	Предварительно импортированный 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. Для поддержки ГОСТ необходимо выполнить команду <cipher gost89>
auth <Алгоритм>	Алгоритм Аутентификации. Для поддержки ГОСТ необходимо выполнить команду <auth gost-mac>
ping <Интервал в секундах>	ping удаленного конца, если нет передачи пакетов в течение промежутка времени, большего чем указанный интервал

команда	параметр
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-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 (TLS used only as a control channel). Для поддержки ГОСТ необходимо выполнить команду <tls-cipher GOST2001-GOST89-GOST89>
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

Примечание:

Команда

```
adm@DionisNX(config—if—svpn0)# server 10.8.0.0:255.255.255.0
```

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—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 ca.crt
adm@DionisNX(config—if—svpn0)# cert server.crt
adm@DionisNX(config—if—svpn0)# key server.key
adm@DionisNX(config—if—svpn0)# dh dh1024.key
```

2. Мульти-клиент-сервер

```
adm@DionisNX(config—if—svpn1)# server 10.8.0.0:255.255.255.0
adm@DionisNX(config—if—svpn1)# ca ca.crt
adm@DionisNX(config—if—svpn1)# cert server.crt
adm@DionisNX(config—if—svpn1)# key server.key
adm@DionisNX(config—if—svpn1)# dh 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
```

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 192.168.4.0:255.255.255.0
```

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 192.168.4.0 255.255.255.0
```

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 10.8.0.113:10.8.0.114
```

21. Экспорт статистики по Netflow

Экспорт статистики по протоколу Netflow предоставляет возможность анализа сетевого трафика на уровне сеансов, делая запись о каждой транзакции TCP/IP. Система 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.

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
```

22. Служба NTP

В системе 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>
```

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
```

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
```

23. Служба DNS

В системе реализована возможность настройки службы DNS, обеспечивающая функции по разрешению DNS-запросов.

Для понимания команд настройки необходимо ввести ряд понятий, непосредственно используемых при настройке:

- view (вид): 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;
- master zone (мастер-зона): DNS-зона, для которой настраиваемый сервер является мастер-сервером, т.е. является первичным уполномоченным сервером;
- slave zone (слэйв-зона): DNS-зона, для которой настраиваемый сервер является вторичным уполномоченным сервером; содержимое зоны считывается от одного из серверов, указанных в опции masters данной слэйв-зоны;
- root zone (корневая зона): описание корневых DNS-серверов; самый новый список корневых серверов можно найти на `ftp://ftp.rs.internic.net/domain/named.root`;
- forward zone (форвард-зона): позволяет перенаправить DNS-запросы по зоне другим серверам.

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
```

23.1 Контроль доступа

acl <NAME> <IPLIST>

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,8} | {[!]A.B.C.D[/MASK],8} >

- any,none,localips,localnets - имена встроенных наборов:
  - any - любые адреса;
  - none - никакие адреса;
  - localips - IP-адреса, присвоенные интерфейсам системы;
  - localnets - IP-адреса сетей, обслуживаемых интерфейсами системы;
- 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 »!».

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>

Определяет, каким именно узлам разрешено выполнять ДНС-запросы через настраиваемый сервер.

По умолчанию: все.

allow query-on <IPLIST>

Определяет, на каких внутренних интерфейсах сервера разрешено принимать ДНС-запросы. По умолчанию: все.

allow query-cache <IPLIST>

Определяет, каким именно узлам разрешено получать ответы на запросы из кэша ДНС настраиваемого сервера.

По умолчанию: все.

allow notify <IPLIST>

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>

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>

Указывает, каким узлам разрешено получать зоны от настраиваемого сервера.

По умолчанию: все.

23.2 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.

auto-local-zones

Создает мастер-зоны 0.in-addr.arpa., 127.in-addr.arpa., 255.in-addr.arpa. и localhost. с параметрами по умолчанию.

Эта команда значительно упрощает конфигурацию службы.

В случае ее задания будет невозможно определить зоны с вышеуказанными именами.

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
```

23.3 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.

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.

Рассмотрим далее команды создания зон, а также команды, специфичные именно для зон данного типа.

Создание любой зоны возможно только в режиме конфигурации вида. Допустим, создан вид с именем vname.

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.».

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.

update [PAS]

Включить динамическое обновление A- и PTR- записей для зоны. Параметр PAS - необязательный пароль, который можно использовать для организации удаленного обновления. Более подробно о настройке динамического обновления см.**Динамическое обновление зон** .

zone slave <ZNAME>

Создает слэйв-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.

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
```

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 переназначаться запросы DNS.

Переназначенный 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 северам 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.

Пример:

```
(config—dns—vname—fwd—zeta.)# forwarders 1.2.3.4 2.3.4.5 only
```

zone .

Позволяет задать корневую зону.

auto [[[daily | weekly | monthly] [SRV]] | 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`;
- 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

23.3.1 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 <NAME>] [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.

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; **mx <NAME> <prio> [domain <NAME>] [ttl <N>]**

Команда создает почтовую MX-запись.

Обязательные параметры:

- имя почтового ретранслятора NAME;
- приоритет почтового ретранслятора.

Необязательные параметры:

- имя домена NAME; если не указано, используется имя текущей зоны;
- время жизни записи N; если не указано, используется значение ttl для зоны.

ns <NSNAME> [domain <NAME>] [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.

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.

23.4 Другие настройки

Данные настройки задаются на разных уровнях службы. Для каждой команды область, где может быть вызвана команда, указана в строке Область определения.

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)

listen [PORT] [IPLIST]

Установка номера порта и IP-адресов, на которых будет слушать ДНС-сервер.

Область определения: служба.

По умолчанию: listen 53 localips.

recursion <yes | no >

Нужно ли обслуживать рекурсивные запросы?

Область определения: служба,вид.

По умолчанию: recursion yes.

23.5 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, отдельным DHCP-сервером или другим удаленным ПО, которое может осуществлять динамические DNS-обновления.

В качестве удаленного DHCP-сервера наиболее совместимым с системой 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.

23.5.1 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 обновляемую зону 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
```

23.5.2 Динамическое обновление DNS удаленной DHCP-службой

Для настройки удаленного автоматического обновления DNS удаленной DHCP-службой 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 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 pas
   soa master raul.cuba.int.
   ns raul.cuba.int.
   ptr 254 raul.cuba.int.

  zone master cuba.int.
   update 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
```

23.5.3 Динамическое обновление удаленным DHCP-сервером

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 ;
- после выполнения команды update PAS на экран будет выведена информация об общем ключе и его параметрах, например: key(name:dnskey,secret:ABC123);
- использовать эту информацию для настройки ISC DHCP-сервера.

Пример:

```
#1. отрывок DNS-конфигурации:
 view v1
  match clients localhost 192.168.33.0/24
  zone master 33.168.192.in—addr.arpa.
   update pas
   soa master raul.cuba.int.
   ns raul.cuba.int.
   ptr 254 raul.cuba.int.
```

```
  zone master cuba.int.
   update 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;
}
```

23.6 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.

23.7 Журналы

log <TYPE> <LEVEL>

Определяет тип и уровень журналирования.

Параметр TYPE задает тип информации для журналирования:

- all-first : любой тип информации; логически расположен в начале списка log-команд;
- all-last : любой тип информации; логически расположен в конце списка log-команд;
- 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.

Пояснение: типы all-last/first нужны для удобства.

- тип all-first: допустим, задано журналирование ВСЕЙ информации с уровнем ERROR с помощью all-first, а затем необходимо указать журналирование только UPDATE-информации с уровнем INFO, при этом остальную ифнормацию оставить на уровне ERROR.
- тип all-last: допустим, задано множество команд журналирования - для каждого типа информации свой уровень; а затем стало необходимым для всех типов информации задать один уровень, при этом сохранив старые настройки (для каждого типа свой уровень); в этом случае следует использовать all-last.

23.8 Диагностика

Рассмотрим команду режима enable для диагностики работы службы DNS.

nslookup <DOMAIN> [TYPE] [rr <RRTYPE>] [server <SERVER>] [port <PORT>]

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 : сервер,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.

23.9 Работа со службой

В этом подразделе рассматриваются команды режима enable для работы со службой DNS.

23.9.1 Команды управления сервисом

service dns reload [dynamic | static]

Это команда, которую нужно выполнить после того, как внесены изменения в ресурсные записи зон, для приянтия внесенных изменеий.

Также ее можно выполнить перед командой show service dns dynamic-rrs, чтобы посмотреть самые свежие данные по динамическим ресурсным запиясм.

Необязательные параметры:

- dynamic : обновить динамические зоны; в этом случае происходит запись в зонные файлы из файлов журналов обновлений; при выключении сервиса это происходит автоматически;
- static : обновить статические зоны; требуется ее выполнить после изменения ресурсных записей мастер-зон.

По умолчанию: если не указан ни один из данных параметров, происходит обновление для всех зон.

Данная команда неявно выполняется для всех зон в момент выключения службы.

23.9.2 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.

23.9.3 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.

24. Служба DHCP

Система 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.

24.1 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

.

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.

24.2 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.

24.3 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).

24.4 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.

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.

24.5 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 инетрвала 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.

24.6 Работа со службой

24.6.1 Запуск и остановка службы

Команды выполняются в режиме config-service-dhcp.

enable

Включить DHCP-сервис.

disable

Выключить DHCP-сервис.

24.6.2 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.

24.6.3 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.

24.7 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
```

25. Служба DCHP-RELAY

Система имеет службу DCHP-RELAY, 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.

25.1 Основные настройки

listen-server <IP>

Задает интерфейс, с которого будут перенаправляться DHCP-запросы на сервера DHCP. Кроме того,этот интерфейс будет действовать так же, как и интерфейсы указанные командой listen, т.е. будет принимать DHCP-запросы/ответы как от клиентов, так и от серверов. Поэтому, если реально присутствует всего один интерфейс, достаточно указать его только командой listen-server.

listen <IFACE>

Задать интерфейс, на котором следует ожидать запросы/ответы DHCP.

По умолчанию: все широковещательные интерфейсы.

enable

Включить службу

disable

Выключить службу

25.2 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)

local-port <PORT>

Порт, на котором слушать DHCP-запросы

По умолчанию: 67.

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

25.3 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
```

26. Служба PROXY

Система 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 , 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.

26.1 Общие понятия

26.1.1 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".

26.1.2 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 - IP-адреса источника (медленный тип);
- dst - IP-адреса назначения (медленный тип);
- myip - IP-адреса локальных интерфейсов системы;
- srcdomain - доменное имя источника;
- dstdomain - доменное имя назначения;
- srcdom-regex - регулярное выражение для доменного имени источника;
- dstdom-regex - регулярное выражение для доменного имени назначения;
- random - случайная частота события (задается дробным числом типа 1/N);
- time - дата и или время события;
- port - порт назначения;
- myport - локальный порт системы;
- proto - протокол HTTP или FTP;

- method - метод HTTP;
- user-agent - приложение клиента (UserAgent);
- proxy-auth - имя пользователя;
- proxy-auth-all - все пользователи;
- proxy-auth-regex - регулярное выражение для имени пользователя;
- maxconn - максисмальное число прямых TCP-соединений (X-Forwarded-For не учитывается);
- max-user-ip - максимальное число попыток аутентификации с разных IP-адресов для одного и того же пользователя;
- req-mime - MIME-тип запроса клиента;
- rep-mime - MIME-тип ответа пользователя;
- mac - MAC-адрес для клиентов из той же подсети;
- uri - регулярное выржание для URI;
- urn - регулярное выржание для URN.

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).

26.1.3 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: «ах?» - это «ах» или «а»;
- «*» - количество С перед данным МС больше либо равно 0; например: «ах*» - это «а», «ах», «ахх», «аххх» и т.д.;
- «+» - количество С перед данным МС больше либо равно 1; например: «ах+» - это «ах», «ахх», «аххх» 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$”
```

26.2 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.

26.2.1 Настройка службы в обычном режиме.

Следует настроить порт и адрес(необязательно), на которых служба будет ожидать входящие запросы:

```
(config—service—proxy)# listen 3128 192.168.0.1
```

Данная команда настраивает службу в обычном режиме, т.е. клиентам необходимо будет указать в настройках своих Интернет-браузеров адрес и порт службы.

26.2.2 Настройка службы в прозрачном режиме.

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). 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 3127 192.168.1.254 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 , 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 вышописанные 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 запрещаны.

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 3127 192.168.1.254 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.

26.3 Настройка параметров кэширования.

Далее настроим параметры службы, описывающие кэш.

Настройка типа кэша на диске:

```
(config—service—proxy)# cache type aufs
```

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 размера кэша на диске(первая команда) и в памяти(вторая команда):

```
(config—service—proxy)# cache disk—size 10000
(config—service—proxy)# cache memory—size 64
```

Размер задается в Мб.

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.

26.4 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.

26.5 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.

26.6 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
```

26.7 Настройка аутентификации

Данный раздел относится к п.3 Алгоритма.

Служба имеет возможность проводить аутентифиуацию 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 1 вид аутентифицации:

- локальная

26.7.1 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.

26.8 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 C, 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 C-сети;
- pool3: общее ведро размером 100Мб и скоростью наполнения 10Мбс, внутри которого есть 256 ведер 1Мбс/10Мб для каждой C-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 C-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.

26.9 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
```

26.10 Настройка журналов службы

Служба имеет систему журналов. Всего их существует 5 типов:

- cache: общий журнал службы; имеет пять уровней подробности журналирования, причем каждый следующий уровень включает сообщения предыдущего:

  - low: : сообщения о критических и фатальных ошибках;
  - middle: : сообщения о важных проблемах службы, предупреждения;
  - 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).

При нормальной работе системы не следует устанавливать уровень cache журнала выше middle, чтобы не замедлять работу системы.

Обычно, если нет проблем в работе службе, достаточно бывает следующих журналов:

```
(config—service—proxy)# log access
(config—service—proxy)# log cache middle
```

Также можно отключить журнал доступа, оставив только общий журнал службы(cache).

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 существует правило log-access.

Рассмотрим пример:

```
(config—service—proxy)# acl vip1 src 10.0.0.1 10.0.0.2
(config—service—proxy)# acl vip2 srcdom—regex ^vip
(config—service—proxy)# log—access deny vip1
(config—service—proxy)# log—access deny vip2
(config—service—proxy)# log—access permit all
```

Данные правила запрещают вести журнал доступа для узлов с адресами 10.0.0.1,10.0.0.2 и узлов, имена которых начинаются с vip. Для всех остальных журнал доступа будет вестись, если он включен.

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.

26.11 Работа со службой

Для запуска службы следует выполнить команду:

```
(config—service—proxy)# enable
```

Для остановки службы следует выполнить команду:

```
(config—service—proxy)# disable
```

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 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.

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».

26.12 Мониторинг службы

Служба прокси имеет мощную систему мониторинга своего состояния. Мониторинг службы осуществляется различными подкомандами команды show service proxy info.

Рассмотрим данные подкоманды.

26.12.1 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.

26.12.2 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.

26.12.3 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.

26.12.4 Аутентификация

Следующая команда показывает статистику по аутентификации пользователей:

```
# show service proxy info auth
```

Имеет смысл только при включённой аутентификации пользователей.

26.12.5 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.

26.12.6 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.

26.12.7 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.

26.12.8 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, межпроцессного 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.

26.12.9 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.

26.12.10 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.

26.12.11 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, хранающихся 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.

26.12.12 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.

26.12.13 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).

26.12.14 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.

26.12.15 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.

26.12.16 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).

26.12.17 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.

26.12.18 Рекомендации по настройке

26.12.18.1 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.

27. Служба SNMP

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.

27.1 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, например:

```
(config—service—snmp)# listen 192.168.0.1 udp
```

По умолчанию,используется UDP-порт 161 и любой локальный интерфейс, которому назначен IP-адрес.

27.2 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 name admin@domain
```

Этими командами можно задать, соответственно, физическое местонахождение системы, имя системы и адрес электронной почты администратора системы.

27.3 Настройка правил доступа

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.

27.4 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.

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).

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).

Включение дополнительных нотификаций осуществляется командой:

```
(config—service—snmp)# trap auth
```

27.5 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
```

28. SSH

В рамках системы протокол SSH не считается защищенным и, соответственно, его использования недостаточно для установления доверенного канала передачи данных. Для создания доверенных каналов передачи данных могут использоваться криптотуннели.

28.1 Сервер SSH

В системе реализована возможность удалённого доступа к командному интерфейсу по протоколу SSH.

Чтобы разрешить удалённый доступ к данному узлу, необходима 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. 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
```

Данная опция может быть очищена с помощью команды «no listen».

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».

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
```

Чтобы остановить службу и удалить все настройки, нужно выполнить команду режима конфигурации:

```
(config)# no service ssh
```

28.1.1 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.*
```

28.2 Клиент SSH

В также реализован клиент SSH для удалённого доступа к другим узлам . Команда доступна как из привилегированного, так и из непривилегированного режима. Формат команды:

```
> ssh <user> <host> [<port>]
```

Для узлов 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
```

28.3 Соединение без использования паролей

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 , 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.

В случае, если хост Б работает под управлением операционной системы отличной от , 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 С 1.2-10 Hand UTM, 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
```

28.4 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

.

29. Telnet

Система имеет службу Telnet, реализующую сетевой протокол уровня приложений для создания текстового интерфейса по сети.

29.1 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)# listen 192.168.1.1
```

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 192.168.1.1 и использовать порт 23.

Возможно указание множества сокетов для принятия соединений.

По умолчанию,если не указано ни одной опции listen, служба ожидает соединения на всех интерфейсах 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

30. Сервис DIWEB

В системе 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.

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».

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
```

31. Служба netperf

31.1 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 :

- 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
```

По умолчанию используется порт 12865.

Включить службу следует командой:

```
(config—service—netperf)# enable
```

Для выключения службы следует выполнить команду:

```
(config—service—netperf)# disable
```

31.2 Команда netperf режима enable

netperf <IP[: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>]

Данная команда осуществляет тестирование пропускной способности сети. Имеет два обязательных параметра:

- IP или HOST - 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 локального 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).

32. Служба IPERF

32.1 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 :

- 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 12345 10.0.0.1
```

По умолчанию: listen tcp 5001 0.0.0.0.

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
```

32.2 Команда iperf режима enable

iperf <IP[: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>]

Данная команда осуществляет тестирование пропускной способности сети. Имеет два обязательных параметра:

- IP - 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.

33. Служба SLAGENT

Система имеет службу slagent, реализующую следующие возможности:

- UDP-эхо сетевых пакетов.

33.1 Настройка

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
```

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
```

34. Служба LLDP

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.

34.1 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 {1,8}>
```

Задает от 1 до 8 портов.

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.

34.2 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)# system hostname host1
(config—service—lldp)# system description ”main core”
(config—service—lldp)# system 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.

34.3 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.

34.4 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.

34.5 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 {1,8}]
```

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 [summary] [IFACE {1,8}]
```

Если указан порт - будет показана информацию по нему. Если указан параметр summary - будет показана краткая сводка по физическим сетевым портам.

35. Служба IDS

Система имеет службу предупреждения и предотвращения вторжений ids.

35.1 Режимы

Служба IDS может работать в 2х режимах:

- ids - система обнаружения вторжений;
- ips - система обнаружения и предупреждения вторжений.

35.2 Конфигурация

Конфигурация службы имеет следующие особенности:

- **всегда** существует одна настройка службы по умолчанию, обрабатывающая все сетевые пакеты приходящие на данный интерфейс;
- **могут** существовать настройки службы, привязанные к конкретным сетям или VLAN-ам; это называется настройка вида (view) службы; пакет,принадлежащий указанным в виде сетям/VLAN-ам, обрабатывается на основе конфигурации данного view; это похоже на конфигурацию службы DNS, в которой тоже есть виды (view);
- часть команд службы может быть задана только в настройке службы, а часть команд - как в настройке службы,так и в настройке вида службы.

35.3 Подсистемы

Сетевые пакеты, проходящие через службу, подвергаются следующей обработке в подсистемах службы:

- daq - выборка сетевых пакетов; настройка осуществляется через команду daq;
- decoder - декодирование; ранний этап обработки анализа пакетов; настройка осуществляется через команду decoder;
- preprocesor - препроцессинг; нормализация, статистический анализ, раннее обнаружение не на основе правил; настройка осуществляется через команду preprocessor;
- rules - обнаружение на основе правил; настройка осуществляется через команду ruleset;
- log - регистрация подозрительных,обнаруженных на предыдущих этапах, пакетов; настройка осуществляется через команду log.

Основную роль в обнаружении и предотвращении атак в службе играет четвертый пункт приведенного выше списка подсистем, а именно - правила.

35.4 Правила

Правила определяют:

- пакет: какие именно пакеты (идентифицируемые по двум тройкам протокол-адрес-порт источника и назначения) следует отслеживать;
- условие: что именно в данных пакетах следует проверять на наличие или отсутствие;
- действие: что именно следует делать с пакетом в случае выполнения условия.

Более подробно о создании правил см. подраздел Настройка подсистемы обнаружения.

Правила службы могут быть пяти типов:

- decoder - правила подсистемы декодирования; их можно включать и отключать; изменять нельзя;
- preprocessor - правила подсистемы препроцессинга; их можно включать и отключать; изменять нельзя;
- detection - правила подсистемы обнаружения; их можно включать и отключать; изменять нельзя;
- private-data - правила подсистемы обнаружения личных данных; их можно включать и отключать; изменять нельзя;
- manual - собственные правила пользователя; их можно создавать,удалять,изменять,включать и отключать.

Первые четыре типа правил - это неизменяемый системный набор. Правила системного набора могут зависеть от системных переменных (см.ниже). Пользовательские правила могут зависеть не только от системных переменных, но и от пользовательских переменных. Рассмотрим далее более подробно понятие переменных.

Всего в системе существует 2 незаисисмых набора правил:

- std - стандартные правила.
- cni - правила режима CNII Pautina/Pluton, на которые система переключается в одном и только одном случае - включен режим ПАУТИНА командой log shmem.

35.5 Идентификаторы правил

В настройках и правилах службы используются идентификаторы подсистем (GID) и идентификаторы правил (SID). Остановимся более подробно на этих параметрах.

GID - идентификатор (положительное число) подсистемы службы ids, которая выдала предупреждение. Некоторые GID:

- 1 - подсистема правил;
- >100 - различные препроцессоры и подсистема декодирования;

- >=8000 - пользовательские правила.

Если в системном 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.

SID - идентификатор (положительное число) правила.

В пользовательских правилах можно указать sid и gid, к значениям которых автоматически прибавляется 1000000 и 8000, соответственно, чтобы они не пересекались с системными sig и gid.

35.6 Переменные

В службе есть понятие именованных переменных, которые могут быть двух типов:

- ipvar - именованная переменная, хранящая IP-адреса;
- portvar - именованная переменная, хранящая номера портов.

Существуют предопределенные наборы системных переменных обоих типов, имеющие некоторые стандартные значения, например: «portvar ftp-ports 21 2100 3535», т.е. переменная с именем ftp-ports равная набору чисел 21,2100,3535, определяющих набор FTP-портов. Смысл переменной определен не в ее определении, а в определении правила, использующего ее.

Системные переменные используются в системном наборе правил. Вы можете их изменять, но не можете удалять.

Пользовательские переменные, в свою очередь, могут создаваться,удаляться и изменяться пользователем по своему усмотрению. Они могут быть использованы в пользовательских правилах.

35.7 Условные обозначения

В описании команды могут быть приведены следующие ее характеристики:

- Тип - это тип команды, определяющий, в каком конфигурационном пространстве она может быть задана: в глобальном пространстве или в пространстве вида:
  - глобальный - обозначает, что команда может быть задана только в настройках службы;
  - вид - обозначает, что команда может быть задана только в настройках вида службы;
  - тип не указан - следовательно команда может быть задана в обоих конфигурационных пространствах, только если в начале текущего подраздела, в котором находится команда, не указан тип всех команд данного подраздела;
- Режим - определяет, в каком режиме команда может выполняться:

  - ips - команда применима только в режиме mode ips;
  - ids - команда применима только в режиме mode ids;
  - режим не указан - в обоих режимах;
- По умолчанию - это значение команды по умолчанию, т.е если не задавать команду. Может отсутствовать - тогда значение не определено.

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.

35.8 Начальная настройка

Служба IDS требует для своего функционирования достаточно много оперативной памяти. Для полноценного использования необходимо хотя бы 500Мб памяти. Минимально необходимый размер памяти 150-200Мб.

35.8.1 Обязательные настройки

service ids

Вход в режим конфигурации службы IDS.

iface <IFACE>

Определение интерфейса, трафик которого необходимо анализировать в службе для обнаружения и/или предотвращения вторжений.

iface-pair <IFACE1> <IFACE2>

Определение интерфейса, который необходимо использовать для анализа и фильтрации трафика (для режима mode ips).

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).

Примечание: в настоящее время данная команда не используется.

mode <ids|ips>

Данная команда определяет ,какой из двух режимов работы службы будет использован: режим обнаружения (ids) или предотвращения вторжений (ips).

В режиме ips работают такие действия правил как drop,sdrop,reject,pass. В режиме ids эти действия не работают.

Тип: глобальный. По умолчанию: ids.

35.8.2 Необязательные настройки

ipvar home-net <IP>

Данная команда задает адрес внутренней (защищаемой) сети.

По умолчанию: any

ipvar external-net <IP>

Данная команда задает адрес внешней сети (источника угроз).

По умолчанию: any

Следующие две команды определяют, атаки на какие именно службы защищаемой сети будут отслеживаться и,возможно,предупреждаться. Для этого путем переопределения системных переменных задаются IP-адреса и номера портов данных служб:

ipvar <dns-servers|smtp-servers|http-servers|sql-servers|telnet-servers|ssh-servers|ftp-servers|sip-servers|aim-servers> {IPVAR|[!]IP[/MSK]|any,8}

Данная команда задает новые значения для системных переменных, хранящих IP-адреса. По умолчанию: см. show service ids ipvars default.

portvar <http-ports|shellcode-ports|oracle-ports|ssh-ports|ftp-ports|sip-ports|file-data-ports|gtp-ports> {PORTVAR|[!]PORT1[:PORT2],8}

Данная команда задает новые значения для системных переменных, хранящих номера портов.

По умолчанию: см. show service ids portvars default.

35.9 MPLS-сети

mpls <multicast|max-headers <VAL|unlimited>|overlapping-ip|payload-type <ipv4|ipv6|ethernet> >

Данная команда устанавливает поддержку MPLS-трафика. Параметры:

- multicast - поддержка многоадресного MPLS-трафика;
- max-headers - максимальное число MPLS-заголовков в пакете;
- overlapping-ip - означает, что IP-адресные пространства могут перекрываться в MPLS сети;
- payload-type - тип полезной нагрузки MPLS.

Тип: глобальный. По умолчанию: MPLS-трафик не поддерживается.

35.10 Настройка ограничений

limit pdu <VAL>

Данная команда задает максимальный размер фрагмента данных PDU (Protocol Data Unit). Несколько PDU внутри одного сегмента TCP или один PDU, распределенный на несколько TCP-сегментов, будет пересобран в один PDU на один пакет. Если PDU превышает заданный, он будет разбит на части, каждая из которых будет в отдельном пакете.

Тип: глобальный. По умолчанию: 16384.

limit asn1 <VAL>

Данная команда задает максимальное число узлов,отслеживаемых в процессе ASN1-декодирования.

Тип: глобальный. По умолчанию: 0 (не отслеживать)

35.11 Переменные

В службе существует понятие переменных. Переменная - это именованный набор элементов одного типа, в котором могут быть также и другие переменные данного типа, за исключением текущей.

Перменные используется в правилах обнаружения.

Существует 2 типа элементов, которые может хранить переменная:

- номера портов - их хранит переменная portvar;
- IP-адреса - их хранит переменная ipvar.

Рассмотрим формат переменных.

[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 следует использовать в команде изменения приоритет,равный приоритету изменяемой переменной.

[PRIO] portvar <NAME> {PORTVAR|[!]PORT1[:PORT2],8}

Параметры команды:

- PRIO - номер ,под которым следует вставить переменную в списке переменных;
- NAME - имя переменной;
- PORTVAR - другая переменная типа portvar;
- PORT1[:PORT2] - порт или интервал портов;
- ! - отрицание следующего параметра.

Имя переменной any - зарезервировано и не может быть переопределено. Переменная portvar any задает все порты.

Переменные также делятся на 2 типа:

- пользовательские - их может создавать ,изменять и удалять пользователь; они далее могут использоваться в пользовательских правилах;
- системные - их можно только изменять; они используются в системных правилах.

Для изменения ранее заданной переменной следует использовать в команде изменения приоритет,равный приоритету изменяемой переменной.

35.12 Виды

Виды позволяют привязать конфигурацию службы к одной или нескольким подсетям или VLAN. Особенности привязки:

- чтобы сетевой пакет попал в вид, проверяется его адрес источника и назначения на вхождение в подсеть или VLAN, указанные в виде;
- если адрес источника попадает в один вид,а адрес назначения в другой, то выбирается тот вид, который имеет меньший номер (приоритет);
- если трафик попадает в какой-либо вид, остальные виды далее не просматриваются на предмет попадания трафика в них;
- если пакет не попадает ни в один вид, используется конфигурация службы по умолчанию, которая должна быть обязательно.

[N] view <NAME>

Данная команда создает вид NAME под номером N.

Далее необходимо привязать вид к подсети или VLAN. Иначе он не будет использоваться.

match <vlans|nets> {<IP|VLAN_ID>,8}

Данная команда привязывает данный вид к одному или нескольким подсетям (nets), либо VLAN (vlans).

По умолчанию: match-ip 0.0.0.0

35.13 Правила отбора

Существует возможность указать, какой именно трафик должен обрабатываться службой.

Это возможно только для режима mode ips.

[N] permit

Данная команда задает правила отбора трафика в службу.

[N] deny

Данная команда задает правила отбора трафика, который не нужно обрабатывать в службе.

Эти правила - упрощенный аналог прафил фильтрации трафика из гл.Фильтрация. Синтаксис правил смотрите в гл.Фильтрация.

35.14 Подсистема выборки пакетов

Цель подсистемы выборки пакетов (data acqustion,daq) - взять пакет из интерфейса и передать его на обработку другим подсистемам службы.

daq bufsize <VAL>

Данная команда позволяет настроить размер буфера (в Мб), выделяемого для подсистемы выборки пакетов.

Тип: глобальный. Режим: ids. По умолчанию: 128Мб

daq snaplen <VAL>

Данная команда определяет, сколько байт из каждого пакета анализируется службой.

Тип: глобальный. Режим: ids. По умолчанию: 1514

daq queue-len <VAL>

Данная команда задает размер очереди пакетов.

Тип: глобальный. Режим: ips. По умолчанию: не ограничено

35.15 Настройки подсистемы декодера

После того, как пакет захвачен системой, он направляется в подсистему декодера для декодирования. Этапы декодирования:

- декодирование канального уровеня поддерживает следующие протоколы: ethernet, token ring, fddi, slip, ppp и др.;
- декодирование сетевого и транспортного уровня: IP, ICMP;
- декодирование транспортного уровня: TCP/UDP.

Таким образом, мы имеем вызов декодеров различных протоколов. На всех этапах декодирования происходит проверка данных, содержащихся на соответствующем уровне пакета и, если обнаруживаются какие-либо аномалии/ошибки в данных и включены соответствующие правила декодера (ruleset decoder), то пакет может быть либо отброшен, либо выдано соответствующее предупреждение декодера.

Следующие команды позволяют настроить базовые особенности всех декодеров. Более специфические особенности определяются правилами декодера, которые подключаются командой ruleset decoder.

decoder checksum <ACT> <TYPE>

Данная команда задает, что делать с пакетами с неверной контрольной суммой.

Действие ACT:

- drop - отбросить пакет;
- calc - расчитать контрольную сумму.

Тип пакета TYPE задает сетевой протокол: ip,icmp,tcp,udp или all(любой протокол).

Режим: ips. По умолчанию: ничего не делать.

no decoder alerts <TYPE>

Данная команда отключает предупреждения подсистемы декодирования. Типы предупреждений:

- all - все предупреждения;
- ipopt - предупреждения, связанные с проверкой длин IP-опций;
- tcpopt - предупреждения, связанные с проверкой длин TCP-опций;
- exttcpopt - предупреждения, связанные с проверкой экспериментальных TCP-опций;
- obstcpopt - предупреждения, связанные с проверкой устаревших TCP-опций;
- tcpttcpopt - предупреждения, связанные с проверкой TCP-опций протокола T/TCP;
- ttcpopt - предупреждения, связанные с проверкой опций протокола T/TCP.

По умолчанию: все включено.

Тип: глобальный По умолчанию: ничего не отключено.

decoder drops <TYPE>

Данная команда включает отбрасывание плохих пакетов. Виды пакетов (TYPE) для отбрасывания:

- all - отбрасывать все плохие пакеты;
- ipopt - отбрасывать пакеты,признанные плохими в связи с проверкой длин IP-опций;
- tcpopt - отбрасывать пакеты,признанные плохими в связи с проверкой длин TCP-опций;
- exttcpopt - отбрасывать пакеты,признанные плохими в связи с проверкой экспериментальных TCP-опций;
- obstcpopt - отбрасывать пакеты,признанные плохими в связи с проверкой устаревших TCP-опций;

- tcpttcpopt - отбрасывать пакеты,признанные плохими в связи с проверкой TCP опций протокола T/TCP
- ttcpopt - отбрасывать пакеты,признанные плохими в связи с проверкой опций протокола T/TCP;
- oversized - отбрасывать пакеты,если значение, находящееся в поле “Длина пакета”, превышает реальную длину пакета.

По умолчанию: ничего не отбрасывать. Тип: глобальный Режим: ips

decoder esp

Данная команда задает декодировать ESP-протокол.

По умолчанию: не декодировать ESP-протокол. Тип: глобальный

decoder teredo

Данная команда задает декодировать Teredo(IPv6-over-UDP-over-IPv4)-протокол.

По умолчанию: не декодировать Teredo-протокол. Тип: глобальный

decoder ignore-ports <tcp|udp> {PORT1[:PORT2],8}

Данная команда предписывает игнорировать указанные tcp- и/или udp-порты при декодировании трафика.

По умолчанию: не игнорировать. Тип: глобальный

35.16 Настройки подсистемы препроцессинга

После того, как пакет прошел подсистему декодера, он попадает в подсистему препроцессора.

Основные функции препроцессора:

- протокольная нормализация (т.е. приведение данных пакета к каноническому виду);
- обнаружение на основе статистики;
- препроцессорное обнаружение.

Подсистема препроцессора пакетов включает в себя множество препроцессоров (ППР), которые по умолчанию отключены.

Каждый ППР анализирует свой протокол и выдает предупреждения, в случае обнаружения аномалий/ошибок в данных протокола или попыток вторжения через него.

Существует два базовых ППР, от которых зависит часть других ППР - это ППР фрагментации и ППР потока. После прохода через эти два препроцессора пакет попадает в остальные ППР, которые соответствуют конкретным протоколам (в основном прикладного уровня), например dns-, ssh-препроцессор и др.

Таким образом путь пакета до входа в подсистему обнаружения на основе правил можно рассмотреть следующим образом:

- декодеры;

- ППР фрагментации (preprocessor frag): ip-дефрагментация: пересборка фргаментированных пакетов;
- ППР потока (preprocessor stream):
  - контроль сессий: проверка принадлежности пакета к установленной сессии;
  - пересборка потока: конструирование псевдо-пакета из пакетов, образующих сессию (поскольку, например, пользовательский ввод telnet-сессии,может быть разделен между несколькими пакетами);
- ППР протокола (preprocessor <PROTO>): полученный псевдо-пакет передается препроцессорам протокола вышележащего уровня для анализа.

В некоторых ППР существует такое понятие как **экземпляр** (instance). Экземпляр ППР позволяет связать настройку ППР для конкретных IP-адресов и/или портов назначения. При входе в экземпляр он по-умолчанию привязан к адресу 0.0.0.0. Если настройка производится не в экземпляре, то будем считать что мы настраиваем **конфигурацию по умолчанию**, которая будет действовать,если пакет не попал ни в один из экземпляров.

Множество команд в экземпляре и в конфигурации по умолчанию не обязательно совпадают. В конфигурации по умолчанию могут присутствовать команды, влияющие на весь данный ППР, включая все его экземпляры, например, команда: limit memory,задающая максимальный объем памяти, который может быть выделен данному ППР. Назовем такие команды **глобальными** командами ППР. Глобальные команды ППР существуют только в конфигурации ППР по умолчанию.

ППР при включении делает возможным использовать специальные ППР-специфичные опции правил, которые позволяют анализировать трафик для протоколов, обслуживаемых данным ППР. Без данных ППР-специфичных опций правил все равно возможно написать правила для анализа того или иного протокола, например при помощи использования опций правил content и byte_test.

Таким образом, препроцессоры влияют на обнаружение на разных стадиях работы службы:

- обнаружение известных атак и аномалий протоколов на ранней стадии (до стадии обнаружения правилами);
- облегчение обнаружения новых(неизвестных) атак и аномалий протоколов во время стадии обнаружения правилами: некоторые ППР вводят специальные опции правил обнаружения,облегчающие анализ соответствующего протокола.

Для настройки ППР следует ввести команду:

preprocessor <NAME>

Здесь NAME - имя препроцессора. В службе имеются следующие препроцессоры:

- frag - фрагментация пакетов;
- stream - ip-,udp-,icmp-,tcp-потоки, соединения, сессии;
- arp - для протокола ARP;
- dcerpc - для протоколов DCE/RPC;
- dnp3 - для протокола DNP3;
- dns - для протокола DNS;
- ftptelnet - для протоколов FTP, TELNET;
- gtp - для протокола GTP;

- http - для протокола HTTP;
- imap - для протокола IMAP;
- modbus - для протокола Modbus;
- norm - нормализация пакетов;
- pop - для протокола POP;
- portscan - обнаружение сканирования портов;
- priv - обнаружение пересылки личных данных;
- rep - настройка репутации;
- rpc - для протокола RPC;
- sip - для протокола SIP;
- smtp - для протокола SMTP;
- ssh - для протокола SSH;
- ssltls - для протоколов SSL, TLS.

35.16.1 Общие команды конфигурации по умолчанию

disabled

Эта команда отключает действие препроцессора для текущего экземпляра ППР или для конфигурации ППР по умолчанию. Глобальные команды продолжают действовать, например, limit memory - продолжает задавать предел доступной памяти для препроцессора, даже если он отключен командой disabled.

Для создания экземпляра ППР, введите команду:

instance <NAME>

Имя экземпляра определяется пользователем. Не для всех ППР возможно создания экземпляра. Для некоторых ППР имя команды для создания экземпляра может отличаться, но общее правило такое, что имя команды начинается на instance. Например, instance-tcp - команда создания экземпляра для ППР потока; данная команда названа так для того,чтобы явно указать, что экземпляр описывает именно TCP-конфигурацию в ППР потока.

limit memory <VAL>

Данная команда определяет максимальный объем памяти для ППР. Всегда задается в Кб.

35.16.2 Общие команды конфигурации экзэмпляра ППР

match-ip <IP[/MSK]>

Данная команда привязывает данный экзэмпляр к одному или нескольким(в зависимости от ППР) IP/адресам назначения и/или портам.

По умолчанию: match-ip 0.0.0.0

detect-anomalies

Данная команда включает обнаружение аномалий протокола или сетевой подсистемы, соответствующей данному ППР. Рекомендуется всегда включать данную опцию, т.к. от нее могут зависеть другие опции ППР.

35.16.3 Препроцессор фрагментации

Данный ППР осуществляет сборку пакетов из фрагментов, поскольку IP-трафик может быть фрагментирован.

Цели препроцессора по части обнаружения опасного трафика:

- службе дается информация об узлах защищаемой сети, чтобы избежать атак, основанных на знании злоумышленником данных о функционировании IP-стека целевых узлов по части сборки фрагментированного TCP-потока, т.е. чтобы избежать атак, связанных с дефрагментацией пакетов в TCP-потоке;
- службе дается информация о топологии защищаемой сети для избежания атак основанных на TTL.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor frag

После настройки ППР необходимо включить правила препроцессора командой ruleset preprocessor.

35.16.3.1 Команды конфигурации по умолчанию

max-frags <N>

Задает максимальное число фрагментов, которые будут одновременно отслеживаться.

По-умолчанию: 8192.

35.16.3.2 Команды экземпляра

max-overlapped <N>

Задает максимальное число перекрывающихся фрагментов. Для работы этой опции необходимо включить detect-anomalies.

По умолчанию: не ограничено.

min-fraglen <N>

Задает минимальный размер фрагмента в байтах. Для работы этой опции необходимо включить detect-anomalies.

По умолчанию: не ограничено.

min-ttl <N>

Задает минимальный размер TTL для фрагментированного пакета. Для работы этой опции необходимо включить detect-anomalies.

По умолчанию: не ограничено.

timeout <N>

Задает время хранения фрагментов в ППР. Если фрагмент не использован в течение этого времени, он будет удален.

По умолчанию: 60 сек.

policy < target-os <OS> | type <TYPE> >

Данная команда задает политику обработки фрагментов на ОС, используемой в защищаемой сети (home-net). Параметр OS определяет ОС защищаемой сети:

- aix - AIX;
- ios - Cisco IOS;
- fbsd - Free BSD;
- hpjd - HP JetDirect (printer);
- hpux10 - HP-UX B.10.20;
- hpux11 - HP-UX 11.00;
- irix - IRIX;
- lin24 - Linux 2.2, 2.4;
- rh - RedHat 7.1-7.3;
- mos - MacOS;
- ncd - NCD Thin Clients;
- obsd - Open BSD;
- ovms - Open VMS 7.1;
- os2 - OS/2;
- osf1 - OSF1;
- sos4 - SunOS 4.1.4;
- sos5 - SunOS 5.5.1,5.6,5.7,5.8;
- tru64 - Tru64 UNIX v5.0A,v5.1;
- vax - Vax/VMS;
- win - Windows(9x,NT4,2000,XP);
- sol - Solaris;
- other - другие ОС.

Параметр TYPE определяет тип обработки фрагментов:

- bsd - преимущество имеет фрагмент со смещением меньше, либо равным смещению последующего фрагмента (по умолчанию);
- bsd-right - перимущество имеет последующий фрагмент,если текущий фрагмент имеет смещение не больше, чем у последущего фрагмента;
- last - преимущество у последующего фрагмента;
- first - преимущество у текущего фрагмента;
- linux - преимущество текущего фрагмента,если его смещение меньше смещения последующего фрагмента;

- windows - политика ОС Windows;
- solaris - преимущество фрагмента, если его смещение меньше смещения текущего фрагмента и больше, либо равно смещения текущего фрагмента.

По умолчанию: policy type bsd.

35.16.4 Препроцессор потока

Цели препроцессора:

- отслеживание TCP-потока и пересборка TCP-псеводпакетов(сессий) из отдельных пакетов для контроля за TCP-сессиями;
- отслеживание UDP,ICMP,IP-сессий.

Для входа в режим конфигурации препроцессора введите команду:

preprocessor stream.

После настройки ППР необходимо включить правила препроцессора командой ruleset preprocessor.

35.16.4.1 Команды конфигурации по-умолчанию

max-sessions <tcp|udp|icmp|ip> <N>

Максимальное число отслеживаемых сессий для выбранного протокола.

По умолчанию:

- для tcp - 262144;
- для udp - в 2 раза меньше, чем для tcp;
- для icmp - в 2 раза меньше, чем для udp;
- для ip - в 4 раза меньше, чем для icmp.

track-sessions <tcp|udp|icmp|ip> <yes|no>

Отслеживать ли сессии для указанных протоколов.

По умолчанию:

- для tcp - yes;
- для udp - yes;
- для icmp - no;
- для ip - no.

timeout <udp|icmp|ip> <N>

Таймаут для сессии udp, icmp или ip.

По умолчанию: 30 сек.

active-response <MAX> <INT>

Данная команда устанавливает параметры для ответных сообщений TCP RST или ICMP Unreachable при отключении сессии:

- MAX - задает максимальное число посылаемых ответных сообщений;
- INT - задает интервал между сообщениями.

По умолчанию: active-response 0 1

instance-tcp <default|NAME>

В этой команде параметр NAME задает имя экзэмпляра TCP-конфигурации. Если имя выбрано default, то это будет экзэмпляр TCP-конфигурации по умолчанию,т.е. он не будет привязан к каким-либо адресам назначения. Все прочие экзэмпляры должны быть привязаны к конкретным адресам командой match-ip. По умолчанию они привязываются к адресу any (0.0.0.0).

35.16.4.2 Команды экземпляра

Команды экзэмпляра ППР описывают TCP-конфигурации ,привязанные к определенным IP-адресам.

flush-factor <N>

Данная команда предписывает производить очистку,если замечено уменьшение размера сегмента после прохождения указанного числа сегментов с не уменьшающимся или постоянным размером. Обычно это означает окончание запроса или ответа.

По умолчанию: 1000.

limit <overlapped | queued-segments | window-size> <N>

Данная команда позволяет установить максимальные значения для:

- overlapped : числа перекрывающихся пакетов на сессию;
- queued-segments : числа сегментов в очереди на пересборку для данной TCP-сессии;
- window-size : размера окна TCP.

По умолчанию:

- limit overlapped 0;
- limit queued-segments 2621;
- limit window-size 0.

policy <TYPE>

Данная команда задает политику обработки TCP-сегментов на узлах защищаемой сети:

- first - предпочтение первому перекрывающемуся сегменту;
- last - предпочтение последнему перекрывающимся сегменту;
- bsd - BSD;
- old-linux - Linux 2.2 и ниже;
- linux - Linux 2.4 и выше;
- windows - Windows(9x,ME,2000,XP);
- win2003 - Windows 2003;
- solaris - Solaris 9.x и выше;
- vista - Windows Vista;
- hpux - HPUX 11 и выше;
- hpux10 - HPUX 10;
- irix - IRIX 6 и выше;
- macos - MacOS 10.3 и выше.

ports <client|server|both> {PORT|SERVICE|PORT1:PORT2,4}

Данная команда задает имя или номер порта или интервалы портов клиента, сервера или обоих, для которых следует осуществлять пересборку пакетов.

По умолчанию: ports client 21 23 25 42 53 80 110 111 135 136 137 139 143 445 513 514 1433 1521 2401 3306

session-hijacking

Данная команда задает возможность обнаружения TCP-атаки «Человек посередине»

timeout

Данная команда устанавливает таймаут сессии

По умолчанию: 30сек.

35.16.5 Препроцессор сканирования портов

Позволяет обнаруживать и предупреждать попытки сканирования портов.

Для входа в режим конфигурации данного препроцессора следует ввести команду:

preprocessor portscan

35.16.5.1 Команды конфигурации препроцессора по умолчанию

detect-ack-scans

Задает возможность обнаружения ACK TCP-сканирования.

По умолчанию: отключено.

watch {IP[/MSK][PORT1[:PORT2]],4}

Задает возможность наблюдать только за сканированием узлов, определяемых указанными адресами и/или парами адрес/порт.

По умолчанию: наблюдать за любыми адресами и портами назначения.

ignore-scanned {IP[/MSK][PORT1[:PORT2]],4}

Игнорировать IP-адреса или пары адрес-порт/порты целей сканирования.

ignore-scanners {IP[/MSK][PORT1[:PORT2]],4}

Игнорировать IP-адреса или пары адрес-порт/порты источников сканирования.

log

Данная команда предписывает вести отдельный журнал , куда будут записываться попытки сканирования. Записи в данный журнал будут более подробными, нежели в журнал предупреждений.

Максимальный размер журнала - 5Мб. Максимальное число архивов журнала - два (текущий и предыдущий).

protocol <all|{PROTO,4}>

Данная команда задает возможность обнаруживать попытки сканирования для перечисленных протоколов. Параметр all - все протоколы.

По умолчанию: все протоколы.

sense <low|medium|high>

Данная команда задает чувствительность ППР: низкую,среднюю и высокую.

type <all|portscan|portsweep|decoy-portscan|distrib-portscan>

Включение типа сканирования:

- portscan - один источник сканирует много портов на одном хосте назначения;
- portsweep - один источник сканирует один порт на множестве хостов назначения;
- decoy-portscan - IP-адрес источника смешивается с другими IP-адресами, чтобы не дать себя обнаружить;
- distrib-portscan - много источников сканируют много портов на одном хосте назначения;
- all - все типы.

35.16.6 Препроцессор SSH

Позволяет обнаруживать и предупреждать следующие типы атак и эксплойтов:

- Challenge-Response Overflow;
- CRC32;
- Secure CRT;
- Protocol Mismatch.

Обнаружение атак, как правило, происходит на ранней стадии SSH-сессии (по умолчанию - в пределах 25 полученных SSH-пакетов).

Для входа в режим конфигурации этого препроцессора следует ввести команду:

preprocessor ssh

35.16.6.1 Команды конфигурации препроцессора по умолчанию

ports <{PORT,4}>

Задает порты, на которых следует ожидать SSH-трафик.

По умолчанию: 22.

inspection max-encrypted-packets <N>

Задает максимальное число анализируемых SSH-пакетов в SSH-сессии. При превышении данного числа пакетов - данная SSH-сессия игнорируется (не анализируется дальше).

По умолчанию: 25.

alert max-client-bytes <N>

Максимальный объем данных без ответа, разрешенных к передаче. При превышении - предупреждение об атаке Challenge-Response Overflow или CRC-32.

По умолчанию: 19600.

alert max-version-length <N>

Максимальная длина строки SSH-версии сервера. При превышении - предупреждение об атаке переполнения Secure CRT.

По умолчанию: 80.

alert <response-overflow-exploit|crc32-exploit|seccrt-exploit|protomismatch-exploit|bad-direction|invalid-payload-size|non-ssh-traffic>

Данная команда включает возможность обнаружения следующих SSH-эксплойтов и атак:

- response-overflow-exploit - Challenge-Response Overflow эксплойт;
- crc32-exploit - CRC32 в SSHv1 эксплойт;
- seccrt-exploit - Secure CRT эксплойт;
- protomismatch-exploit - Protocol Mismatch эксплойт;
- bad-direction - клиент генерирует серверный трафик и/или наоборот;
- invalid-payload-size - неверный размер SSH-данных;
- non-ssh-traffic - не SSH-трафик на SSH-портах;
- non-ssh-ports - SSH-трафик на не SSH-портах.

По умолчанию: все отключено.

35.16.7 Препроцессор личных данных PRIV

Позволяет обнаруживать и предупреждать попытки пересылки личных данных (номер кредитных карт и т.д.).

Для входа в режим конфигурации этого препроцессора следует ввести команду:

preprocessor priv

35.16.7.1 Команды конфигурации препроцессора по-умолчанию

mask-output

Включение функции сокрытия личных данных в журналах предупреждений.

alert private-data <NUM>

Инициирование предупреждения о пересылке личных данных, если они были замечены в пакетах хотя бы NUM раз.

По умолчанию: 25

35.16.8 Препроцессор DNS

Позволяет обнаруживать и предупреждать DNS-атаки.

Для входа в режим конфигурации этого препроцессора следует ввести команду:

preprocessor dns

35.16.8.1 Команды конфигурации по умолчанию

ports <{PORT,4}>

Данная команда задает номера портов, на которых ППР должен проверять DNS-трафик.

По умолчанию: 53.

alert <obsoleted-rrs | experimental-rrs | rdata-overflow>

Включение возможности выдачи предупреждения о:

- obsoleted-rrs - использовании устаревших ресурсных записей (RFC 1035);
- experimental-rrs - использовании экспериментальных ресурсных записей (RFC 1035);
- rdata-overflow - переполнении TXT RData клиентом DNS.

По умолчанию: не предупреждать.

35.16.9 Препроцессор HTTP

Позволяет обнаруживать и предупреждать HTTP-атаки.

Для входа в режим конфигурации этого препроцессора следует ввести команду:

preprocessor http

35.16.9.1 Команды конфигурации по умолчанию

codepage <VAL>

Данная команда задает кодировку для декодирования символов UTF.

По умолчанию: 1251.

Строки реальной конфигурации: IIS Unicode Map Codepage: 1251

limit memory <log | decompress>

Задает максимальные объемы памяти ППР для

- log - журналирования данных о URI и Hostname HTTP-сессий; этот параметр определяет максимальное число сессий, которые могут обрабатываться одновременно;
- decompress - декомпрессии; этот параметр определяет максимальное число сессий, которые могут быть декомпрессированны одновременно.

Строки реальной конфигурации:

- limit memory log : Memcap used for logging URI and Hostname : 23040000
- limit memory decompress : Max Gzip Memory: 900000

*По умолчанию : * limit memory log = 2304 * MAX_CONS = 23040000 , где MAX_CONS = 10000 - число сессий, которые могут быть журналированы одновременно. * limit memory decompress = 90 * MAX_COMP_CONS = 900000, где MAX_COMP_CONS = 10000 - число сессий, которые могут быть декомпрессированы одновременно.*

limit compressed-size <VAL>

Задает максимальный размер сжатых данных пакета для декомпрессии.

Строки реальной конфигурации:

- limit compressed-size : Gzip Compress Depth: 3276 и Gzip Decompress Depth: 6552 (в 2 раза больше)

По умолчанию: 3276.

alert <non-http-ports | proxy>

Включение возможности предупреждать, если:

- non-http-ports - замечен HTTP-трафик на не-HTTP портах;
- proxy - замечено использование клиентами непрозрачных прокси-серверов.

Строки реальной конфигурации:

- alert proxy: Detect Proxy Usage: NO

По умолчанию: не предупреждать

instance <default|NAME>

Создает экземпляр ППР и входит в его настройку. В HTTP-прерпроцессоре могут быть 2 типа экзэмпляров:

- обычный экзэмпляр - обрабатывает трафик с указанными в его конфигураци HTTP-серверами;
- экзэмпляр по умолчанию - обрабатывает весь остальной HTTP-трафик, не попавший в обычные экземпляры.

35.16.9.2 Команды экземпляра

profile <all|apache|iis|iis4|iis5>

Задает профиль экзэмпляра:

- all - для всех типов серверов;
- apache - для Apache-серверов;
- iis - для IIS-серверов;
- iis4/5 - для IIS4- и IIS5-серверов.

Эта команда нужна для упрощения настройки экзэмпляра. Следует учитывать, что не все команды экзэмпляра будут доступны,если вы задали эту команду. Таким образом, для наиболее тонкой настройки использовать эту команду не нужно.

Команды, доступные после задания команды profile:

- ports;
- codepage;
- alert all;
- alert proxy;
- alert normalize;
- alert decompress;
- alert limit dir-size;
- alert limit req-header-size;
- alert limit req-headers;
- alert limit req-header-spaces;
- inspect client request;
- inspect client uri-only;
- inspect server;
- inspect methods;
- inspect cookies;
- log xff;
- log uri;
- log hostname.

Остальные команды настройки экземпляра использовать будет невозможно, если перед этим будет выдана команда profile.

Следующие команды задают определенные начальные значения параметров конфигурации экзэмпляра. Эти начальные значение можно затем посмотреть по команде show service proxy log.

ports <{PORT,4}>

Эта команда задает порты HTTP-трафика серверов данного экэмпляра.

*По умолчанию: * если задана команда profile, порт по умолчанию - 80 * если не задана команда profile - все порты.*

codepage <NUM>

Эта команда задает кодировку символов для текущего экзэмпляра.

По умолчанию: 1251

log <uri|xff|hostname>

Включить ведение журнала ППР для:

- hostname - поля Host заголовка HTTP;
- uri - данных URI;
- xff - поля X-Forwarded-For заголовка HTTP.

По умолчанию: не вести журнал.

inspect client post <all|none|SIZE>

Эта команда предписывает проверять сообщения POST-клиента. Параметры:

- all - проверять все данные в сообщении POST (по умолчанию);
- none - не проверять сообщения POST;
- SIZE - проверять указанное число байт сообщения POST (от 1 до 1460).

По умолчанию: inspect client post all

inspect client request <all|none|NUM>

Эта команда предписывает проверять HTTP-запросы клиентов. Параметры:

- all - проверять все данные в запросах клиентов (по умолчанию);
- none - не проверять запросы клиентов;
- SIZE - проверять указанное число байт в запросах клиентов. (от 1 до 1460).

По умолчанию: inspect client request all

inspect client <uri-only | no-pipiline-req | non-strict-uri>

Эта команда предписывает установить параметры проверки запросов клиентов:

- uri-only - проверять только URI-запросы (нужны uricontent правила);
- no-pipeline-req - не проверять конвейерные запросы HTTP;

- non-strict-uri - подразумевается,что URI находится между первым и вторым пробелами в строке GET запроса; обработка плохо сформированных URI.

*По умолчанию: все опции отключены: * проверять не только URI, но и данные запроса * проверять конвейерные запросы * URI должен быть правильно сформирован (без лишних данных после второго пробела с строке GET запроса).*

inspect server response <all|none|NUM>

Эта команда предписывает проверять HTTP-ответы серверов. Параметры:

- all - проверять все данные в ответах серверов (по умолчанию);
- none - не проверять ответы серверов;
- SIZE - проверять указанное число байт в ответах серверов (от 1 до 65535).

По умолчанию: inspect server response all

inspect cookies

Эта команда включает проверку cookies HTTP-запросов и ответов.

По умолчанию: не проверять.

inspect methods <METHODS>

Эта команда включает проверку следующих типов HTTP-методов помимо GET и POST, проверяемых по умолчанию: options,head,put,patch,delete,trace,link,unlink,connect.

По умолчанию: проверять только GET- и POST-методы.

no alert all

Эта команда выключает все предупреждения ППР.

По умолчанию: предупреждения включены.

no alert proxy

Эта команда разрешает использование прокси-серверов для данного экзэмпляра без выдачи предупреждения.

Эта опция имеет смысл, только если включена опция конфигурации по умолчанию proxy alert.

По умолчанию: не разрешать использование прокси.

alert limit <dir-size | small-chunks <MAXSIZE> <NUM> | req-header-size | req-header-spaces | req-headers |chunk-size>

Эта команда позволяет задать различные пределы параметров ППР и назначает выдачу предупреждений об их превышении.

Параметры:

- dir-size - максимальный размер имени директории в URL;
- req-header-size - максимальный размер заголовка запроса;
- req-header-spaces - максимальное число пробелов заголовке запроса;
- req-headers - максимальное число полей в HTTP-заголовке;

- small-chunks - задает максимальный размер маленького куска (MAXSIZE); если число маленьких кусков идущих подряд превысит NUM - будет предупреждение; куски могут быть в запросе или ответе,если в нем есть поле Transfer-Encoding: chunked;
- chunk-size - задает максимальный размер куска,при превышении которого будет предупреждение.

По умолчанию: нет ограничений.

alert chars <multi-slash <YESNO> | iis-backslash <YESNO> | apache-delimiter <YESNO> | iis-delimiter <YESNO> | tab-uri-delimiter | non-rfc-chars <{HEX,4}> >

Эта команда предписывает предупреждать об:

- multi-slash - использовании множества слэшей подряд;
- iis-backslash - использовании обратного слэша вместо обычного;
- apache-delimiter - использовании табуляции в качестве разделителя;
- iis-delimiter - использовании нестандартных разделителей IIS;
- tab-uri-delimiter - использовании табуляции в качестве разделителя для URI;
- non-rfc-chars - использование прочих символов в строке запрашиваемого URI (до 4х).

Параметр YESNO:

- yes - обрабатывать и предупреждать;
- no - только обрабатывать.

По умолчанию: отключено

alert decompression [unlimited]

Эта команда предписывает предупреждать об ошибках в декомпрессии сжатых данных.

Необязательный параметр unlimited - возможность проверки декомпрессии неограниченного объема сжатых данных. Этот параметр можно задать,если предварительно задали limit compressed-size 65535.

По умолчанию: отключено

alert directory-path <traversal | webroot-traversal> <YESNO>

Эта команда предписывает обрабатывать и предупреждать об:

- traversal - использовании самоссылающихся путей файлов и механизма обхода каталогов через путь файла,например /foo/./bar, /foo/../bar;
- webroot-traversal - атаке выхода в файловую систему HTTP-сервера.

Параметр YESNO:

- yes - обрабатывать и предупреждать;
- no - только обрабатывать.

По умолчанию: отключено

alert encoding extended-ascii-uri

Эта команда включает поддержку расширенных ASCII-кодов в URI HTTP-запроса.

По умолчанию: отключено.

alert encoding <ascii | iis-double-decoding | utf8 | iis-unicode | iis-u-encoding|iis-bare-byte-encoding> <YESNO>

Эта команда определяет, нужно ли декодировать текст по той или иной технологии и,возможно, предупреждать об этом.

Обрабатывать и предупреждать об использовании:

- ascii - ascii-кодирования;
- iis-double-decoding - IIS 2-фазного кодирования;
- utf8 - кодирования UTF-8 в URI;
- iis-unicode - кодирования UTF-8 в сообщении;
- iis-u-encoding - u-кодирования;
- iis-bare-byte-encoding - кодирования bare-байт.

Параметр YESNO:

- yes - обрабатывать и предупреждать;
- no - только обрабатывать.

По умолчанию: все отключено

alert normalize <utf|cookies|headers|javascript [NUM]>

Эта команда предписывает предупреждать об ошибках нормализации различных сущностей HTTP.

Параметры:

- utf - ошибка нормализации ответа с типом Content-Type: utf-16(32)le(be) в ответ с типом utf-8;
- cookies - ошибка нормализации cookies;
- headers - ошибка нормализации полей заголовка (кроме поля Cookies);
- javascript - ошибка нормализации Javascript ; NUM - максимальное число последовательных пробелов в JavaScript.

По умолчанию: не предупреждать.

35.16.9.3 Таблица соответствия команд службы и строк конфигурации

Ниже приводится таблица соответствия строк конфигурации ППР из таблицы,выводимой по команде show service ids log, и команд службы.

Строка реальной конфигурации	Команда
SERVER	match-ip
Server profile	profile
Ports (PAF)	ports
Server Flow Depth	inspect server respinse
Client Flow Depth	inspect client request
Max Chunk Length	alert limit chunk-size
Max Header Field Length	alert limit req-header-size
Max Number Header Fields	alert limit req-headers
Max Number of WhiteSpaces allowed..	alert limit req-header-spaces
Inspect Pipeline Requests	inspect client no-pipeline-req
URI Discovery Strict Mode	inspect client non-strict-uri
Allow Proxy Usage	alert proxy
Disable Alerting	alert all
Oversize Dir Length	alert limit dir-size
Only inspect URI	inspect client uri-only
Normalize HTTP Headers	alert normalize headers
Inspect HTTP Cookies	alert normalize cookies
Inspect HTTP Responses	всегда включено
Extract Gzip from responses	alert decompression
Unlimited decompression …	alert decompression unlimited
Normalize Javascripts in HTTP Resp..	alert normalize javascript
Normalize HTTP Cookies	alert normalize cookies
Enable XFF and True Client IP	log xff
Log HTTP URI data	log uri
Log HTTP Hostname data	log hostname
Extended ASCII code support in URI	alert encoding extended-ascii-uri
Ascii	alert encoding ascii
Double Decoding	alert encoding iis-double-decoding
%U Encoding	alert encoding iis-u-encoding
Bare Byte	alert encoding iis-bare-byte-encoding
UTF 8	alert encoding utf-8
IIS Unicode	alert encoding iis-unicode
Multiple Slash	alert chars multi-slash
IIS Backslash	alert chars iis-backslash
Directory Traversal	alert directory-path traversal
Web Root Traversal	alert directory-path webroot
Apache WhiteSpace	alert chars apache-delimiter
IIS Delimiter	alert chars iis-delimiter
IIS Unicode Map Codepage	codepage
Non-RFC Compliant Characters	alert chars non-rfc-chars

35.16.9.4 Начальные значения при отсутствии profile

SERVER: 0.0.0.0
  Server profile: All

Ports (PAF):
Server Flow Depth: 0
Client Flow Depth: 0
Max Chunk Length: 0
Max Header Field Length: 0
Max Number Header Fields: 0
Max Number of WhiteSpaces allowed with header folding: 0
Inspect Pipeline Requests: YES
URI Discovery Strict Mode: YES
Allow Proxy Usage: NO
Disable Alerting: NO
Oversize Dir Length: 0
Only inspect URI: NO
Normalize HTTP Headers: NO
Inspect HTTP Cookies: NO
Inspect HTTP Responses: YES
Extract Gzip from responses: NO
Unlimited decompression of gzip data from responses: NO
Normalize Javascripts in HTTP Responses: NO
Normalize HTTP Cookies: NO
Enable XFF and True Client IP: NO
Log HTTP URI data: NO
Log HTTP Hostname data: NO
Extended ASCII code support in URI: NO
Ascii: OFF
Double Decoding: OFF
%U Encoding: OFF
Bare Byte: OFF
UTF 8: OFF
IIS Unicode: OFF
Multiple Slash: OFF
IIS Backslash: OFF
Directory Traversal: OFF
Web Root Traversal: OFF
Apache WhiteSpace: OFF
IIS Delimiter: OFF
IIS Unicode Map Codepage: 1251
Non—RFC Compliant Characters: NONE
Whitespace Characters: NONE

35.16.9.5 Начальные значения для profile all

Те же самые, что и для отсутствия профиля. Отличия следующие:

Ports (PAF): 80
Server Flow Depth: 300
Client Flow Depth: 300
Max Chunk Length: 500000
Max Number of WhiteSpaces allowed with header folding: 200

Ascii: YES alert: NO
Double Decoding: YES alert: YES
%U Encoding: YES alert: YES
Bare Byte: YES alert: YES
IIS Unicode: YES alert: YES
Multiple Slash: YES alert: NO
IIS Backslash: YES alert: NO
Directory Traversal: YES alert: NO
Web Root Traversal: YES alert: YES
Apache WhiteSpace: YES alert: NO
IIS Delimiter: YES alert: NO

35.16.9.6 Начальные значения для profile apache

Те же самые, что и для отсутствия профиля. Отличия следующие:

Ports (PAF): 80
Server Flow Depth: 300
Client Flow Depth: 300
Max Chunk Length: 500000
Max Number of WhiteSpaces allowed with header folding: 200
URI Discovery Strict Mode: NO
Ascii: YES alert: NO
UTF 8: YES alert: NO
Multiple Slash: YES alert: NO
Directory Traversal: YES alert: NO
Web Root Traversal: YES alert: YES
Apache WhiteSpace: YES alert: NO

35.16.9.7 Начальные значения для profile iis

Те же самые, что и для отсутствия профиля. Отличия следующие:

Ports (PAF): 80
Server Flow Depth: 300
Client Flow Depth: 300
Max Chunk Length: 500000
Max Number of WhiteSpaces allowed with header folding: 200
URI Discovery Strict Mode: NO
Ascii: YES alert: NO
%U Encoding: YES alert: YES
Bare Byte: YES alert: YES
IIS Unicode: YES alert: YES
Multiple Slash: YES alert: NO
IIS Backslash: YES alert: NO
Directory Traversal: YES alert: NO
Web Root Traversal: YES alert: YES
Apache WhiteSpace: YES alert: NO

IIS Delimiter: YES alert: NO

35.16.9.8 Начальные значения для profile iis4

Те же самые, что и для profile iis. Отличия следующие:

Double Decoding: YES alert: YES

35.16.9.9 Начальные значения для profile iis5

Те же самые, что и для profile iis. Отличия следующие:

Double Decoding: YES alert: YES

35.16.10 Препроцессор FTP/TELNET

Позволяет обнаруживать и предупреждать FTP- и Telnet-атаки.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor ftptelnet

35.16.10.1 Команды конфигурации препроцессора по умолчанию

alert encrypted-traffic [continue]

Эта команда предписывает предупреждать о зашифрованном FTP/TELNET-трафике.

Параметр continue: продолжать проверять зашифрованную сессию до тех пор, пока не встретится команда окончания шифрования.

alert telnet-ayt <N>

Эта команда предписывает предупреждать, когда число последовательных AYT (Are You There) команд достигнет указанного значения. Только для sateful-режима.

alert telnet-anomalies

Эта команда предписывает предупреждать об аномалиях Telnet-протокола: если согласование, начатое командой SB, не завершается командой SE.

telnet-normalize

Эта команда включает функцию нормализации Telnet-трафик: удаление Telnet ESC-последовательности.

telnet-ports <{PORT,4}>

Эта команда предписывает рассматривать указанные порты как порты Telnet-трафика.

35.16.10.2 Команды экземпляра FTP-сервера

Для входа в режим конфигурации экзэмпляра следует ввести команду:

instance-ftpserver <NAME>

Параметр NAME задает имя конфигурации экзэмпляра.

Если указано имя default, происходит вход в конфигурацию экзэмпляра FTP-сервера по умолчанию.

Если указано имя, отличное от default, происходит вход в конфигурацию экзэмпляра FTP-сервера, привязанную к IP-адресу.

Если трафик не попадает ни в одну из конфигураций, привязанных к IP-адресу, он обслуживается конфигурацией экзэмпляра FTP-сервера по умолчанию.

ports <{PORT,4}>

Эта команда задает порты, рассматриваемые препроцессором, как порты трафика командного канала FTP.

alert allowed-cmds <{CMD,8}>

Эта команда задает дополнительные разрешенные FTP-команды в дополнение к тем, что описаны в RFC 959.

Все остальные команды будут рассматриваться, как неизвестные, и будут выдаваться предупреждения, если они будут замечены.

alert param-format <{CMD,8}>

Эта команда предписывает проверять атаки на формат строк для указанных команд.

alert telnet-esc <yes|no>

Эта команда определяет, предупреждать или нет о Telnet ESC-последовательностях в командном канале FTP.

alert ignore-telnet-erase <yes|no>

Эта команда определяет, игнорировать или нет Telnet ESC-последовательности для TNC EAC и TNC EAL в командном канале FTP.

alert limit paramlen <VAL>

Эта команда предписывает предупреждать о превышении заданной длины параметра (VAL) команды FTP.

alert limit paramlen-list <VAL> <{CMD,8}>

Эта команда предписывает предупржедать о превышении заданной длины параметра (VAL) указанных команд FTP (CMD).

param-format <CMD> <FORMAT>

Эта команда задает формат параметров для указанной команды CMD.

Параметр FORMAT описывается следующей таблицей:

- int - целое число;
- number - целое число от 1 до 255;

- char <ABC…> : любой символ из перечисленных;
- date <DATEFMT> : задает формат даты, где DATEFMT:
  - n : число;
  - C : символ;
  - : необязательный формат;
  - {F1 | F2 | …} : один из форматов;
  - .+- : другие разрешенные символы;
- string : строка;
- host_port : имя узла и порт по RFC 959;
- long_host_port : имя узла и порт по RFC 1639;
- extended_host_port : имя узла и порт по RFC 2428;
- {F1 | F2 | …} : один из форматов;
- [F1] : необязательный формат.

inspection ignore-data-channel <yes|no>

Эта команда определяет, включить (yes) или выключить (no) игнорирование FTP-канала данных правилами и остальными ППР.

Команда нужна для оптимизации, когда известно, что FTP-передача данных идет из доверенного источника.

Если используются правила для отслеживания вирусов - не следует включать эту опцию.

35.16.10.3 Команды экземпляра FTP-клиента

Для входа в режим конфигурации экземпляра следует ввести команду:

instance-ftpclient <NAME>

Параметр NAME задает имя конфигурации экзэмпляра.

Если указано имя default, происходит вход в конфигурацию экземпляра FTP-клиента по умолчанию.

Если указано имя, отличное от default, происходит вход в конфигурацию экзэмпляра FTP-сервера, привязанную к IP-адресу.

Если трафик не попадает ни в одну из конфигураций, привязанных к IP-адресу, он обслуживается конфигурацией экзэмпляра FTP-клиента по умолчанию.

alert telnet-esc <yes|no>

Команда определяет, предупреждать или нет о Telnet ESC-последовательностях в командном канале FTP.

alert ignore-telnet-erase <yes|no>

Эта команда определяет, игнорировать или нет Telnet ESC-последовательности для TNC EAC и TNC EAL в командном канале FTP.

alert bounce <yes|no> [{IP:PORT|IP:PORT1:PORT2],4}]

Эта команда определяет, предупреждать или нет об атаках FTP-bounce. Необязательные параметры - это адреса и порты, которые разрешены для открытия сервером будучи указаны в команде PORT.

Атака FTP-bounce происходит, когда в FTP-команде PORT указан адрес, который не равен адресу клиента, который прислал данную команду. Атака используется для обхода межсетевых экранов при сканировании портов: сервер может попытаться открыть сокет на интересующей атакующего машине при помощи FTP-команды PORT.

alert limit response-len <LEN>

Эта команда предписывает предупреждать, если размер FTP-ответа сервера превышает указанное максимальное значение. Такая проверка является базовым методом обнаружения переполнения буфера на сервере.

35.16.11 Препроцессор SMTP

Позволяет обнаруживать и предупреждать SMTP-атаки.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor smtp

35.16.11.1 Команды конфигурации препроцессора по умолчанию

ports <{PORT,4}>

Эта команда задает порты SMTP-трафика.

alert <base64-decoding|bitenc-decoding|qp-decoding|uu-decoding> <all|none|N>

Эта команда предписывает предупреждать о неуспешных попытках декодирования или извлечения MIME-вложений.

Параметры:

- base64-decoding - BASE-64-декодирование MIME-вложений;
- bitenc-decoding - извлечение незакодированных MIME-вложений;
- qp-decoding - Quoted-Printable-декодирование MIME-вложений;
- uu-decoding - Unix-to-Unix-декодирование вложений;
- all - неограниченное потребление памяти для декодирования;
- none - не декодировать;
- N - объем памяти в байтах для декодирования.

alert commands <unknown | invalid {CMD,8}>

Эта команда предписывает предупреждать, если получена неизвестная или неразрешенная SMTP-команда. Параметры:

- unknown - получена неизвестная команда;
- CMD - список из неразрешенных SMTP-команд.

alert normalize <all | none | commands {CMD,8}>

Эта команда предписывает нормализовать команды и предупреждать о провалах нормализации: проверка наличия более одного пробела после команды. Параметры:

- all - проверять все команды;
- none - не проверять команды;
- commands - список проверяемых команд.

alert disable

Эта команда выключает все предупреждения данного ППР.

alert limit <header N | response N| cmdline N | cmdline-list N {CMD,8}>

Эта команда предписывает предупреждать о превышении указанной длины N:

- header - SMTP-заголовка;
- response - SMTP-ответа;
- cmdline - командной строки SMTP;
- cmdline-list - командной строки SMTP для указанных команд CMD.

alert xlink2state [drop]

Эта команда предписывает выключить предупреждения о команде xlink2state и ,возможно, отбрасывать данную команду, если указан параметр drop.

Команда xlink2state используется для обмен сведениями о топологии маршрутизации между Microsoft Exchange SMTP-серверами.

inspect mode <stateless|stateful>

Эта команда определяет установку режима ППР:

- stateless - без состояния;
- stateful - с состоянием (многие функции ППР выполняют свою работы не только над пакетами, но и над данными, распределенными по многим пакетам).

inspect ignore-data

Эта команда предписывает игнорировать данные SMTP-сообщений при анализе правилами.

inspect ignore-tls-data

Эта команда предписывает игнорировать зашифрованные по TLS данные SMTP-сообщений при анализе правилами.

limit memory <decoding | log> <N>

Эта команда определяет максимальный размер памяти для:

- decoding - декодирования вложений;
- log - журналирования команд.

Размер памяти зависит от других параметров ППР. Если будет указан слишком маленький размер - будет выведено сообщение о соответствующей ошибке с рекомендацией о размере лимита памяти.

log <mailfrom|rcptto|filename|email-hdrs [N]|all>

Эта команда предписывает вести журнал для:

- mailfrom - e-mail адрес отправителя (MAIL FROM команда);
- rcptto - e-mail адрес получателя (RCPT TO команда);
- filename - имена вложений;
- email-hdrs - заголовки SMTP; N - длина журналируемой части заголовка;
- all - все перечисленное.

35.16.12 Препроцессор ARP

Позволяет обнаруживать и предупреждать ARP-атаки. Работает только в режиме ids. Таким образом, предупреждая,но не запрещая подозрительные пакеты.

Цели этого ППР:

- декодирование ARP-пакетов;
- обнаружение ARP-атак, одноадресных ARP-запросов и несогласованных отображений Ethernet в IP.

Для входа в режим конфигурации данного препроцессора следует ввести команду:

preprocessor arp

Если не задавать ни каких опций в ППР, он будет осуществлять только проверку несогласованных отображений MAC в IP.

35.16.12.1 Команды конфигурации ППР по умолчанию

alert spoofing <IP> <MAC>

Эта команда задает правильное отображение MAC-адреса в IP-адрес. Необходимо для отслеживания атак загрязнения ARP-кэша (атак типа ARP-spoofing).

alert unicast

Эта команда задает отслеживание одноадресных ARP-запросов. Существование таких запросов может быть подозрительным, т.к. ARP-запросы должны быть широковещательными.

35.16.13 Препроцессор RPC

Позволяет обнаруживать и предупреждать RPC-атаки.

Цели ППР:

- нормализация фрагментированного RPC-запроса в один нефрагментрованный RPC-запрос.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor rpc

35.16.13.1 Команды конфигурации ППР по умолчанию

ports {PORT,4}

Эта команда устанавливает порты RPC-трафика.

alert fragments

Эта команда предписывает предупреждать о фрагментированных RPC-запросах.

no alert <incomplete|multiple|large>

Эта команда предписывает не предупреждать:

- если весь запрос не помещается в один пакет;
- если в одном пакете множество запросов;
- если сумма фрагментированных записей превышает один пакет;

35.16.14 Препроцессор DCERPC

Позволяет обнаруживать и предупреждать DCE/RPC-атаки.

DCE/RPC - это технология реализации RPC (Удаленного вызова процедур) для DCE (Распределённая вычислительная среда). Транспорт для DCE/RPC: SMB,TCP,UDP,RPC-over-HTTP.

Цели ППР:

- SMB-дефрагментация для команд: Write, Write Block Raw, Write and Close, Write AndX,Transaction, Transaction Secondary, Read, Read Block Raw и Read AndX;
- DCE/RPC-дефрагментация.

Для входа в режим конфигурации этого препроцессора введите команду:

preprocessor dcerpc

35.16.14.1 Команды конфигурации по умолчанию

limit fragment-size <N>

Эта команда задает максимальный рразмер фрагмента, который может участвовать в дефрагментации. Если фрагмент больше указанного, он усекается до указанного размера и далее далее идет на процедуру дефрагментации.

smb-fingerprint-policy <server |client|both>

Эта команда устанавливает политику работы ППР на основе данных, полученных от SessionSetupAndX-запроса и соответствующего ответа.

В начальной фазе SMB-сессии клиенту нужно аутентифицировать себя с помощью команды SessionSetupAndX. Данная команда и ответ на нее содержат информацию об ОС и версии, которая позволяет ППР установить политику обработки трафика для данной сессии и лучше защититься против Windows- и Samba-атак.

threshold reassemble

эта команда устанавливает минимальный размер данных в буфере дефрагментации и десегментации, при достижении которого осщуествляется пересборка пакета и посылка его в подсистему обнаружения.

По умолчанию: отключено.

no defatagmentation

Эта команда отключает дефрагметнацию в ППР.

По умолчанию: не отключать.

alert <smb|memory|connection-oriented|connectionless|all>

Эта команда предписывает предупреждать в случае:

- memory - достижения или превышения limit memory-значения;
- smb - событий,связанных с обработкой SMB-трафика;
- connection-oriented - событий,связанных с обработкой DCE/RPC с установкой соединения;
- connectionless - событий,связанных с обработкой DCE/RPC без установки соединения;
- all - всех перечисленных событий.

35.16.14.2 Команды конфигурации экзэмпляра

alert <shares {SHARE,4} |max-andx-commands>

Эта команда предписывает предупреждать в случае:

- shares - попытки соединения с указанными ресурсами по SMB с помощью команд TreeConnect, TreeConnectAndX;
- max-andx-commands - максимальное число SMB AndX-команд в очереди, при превышении которого выдается предупреждение.

policy <w2000|w2003|xp|vista|samba|samba320|samba322>

Эта команда задает ОС защищаемого сервера:

- w2000 - Windows 2000;
- w2003 - Windows 2000;
- xp - Windows XP;
- vista - Windows Vista;
- samba - Samba;
- samba320 - Samba-3.0.20;

- samba322 - Samba-3.0.22.

По умолчанию: policy xp.

ports <PROTO> [{PORT,4}] [{PORT1:[PORT2],4}]

Эта команда устанавливает порты, используемые в качестве транспорта DCE/RPC, и серверные DCE/RPC-порты. Параметры:

- PROTO: протокол, для которого определяются порты:
  - smb - SMB-порты;
  - tcp - TCP-порты
  - udp - UDP-порты
  - rpc-over-http-proxy - RPC-over-HTTP proxy;
  - rpc-over-http-server - RPC-over-HTTP server;
- PORT,PORT1,PORT2 - порты и интервалы портов.

35.16.15 Препроцессор DNP3

Позволяет обнаруживать и предупреждать DNP3-атаки. Протокол DNP3 используется в SCADA-сетях. Если сеть не содержит DNP3-устройств, то не нужно использовать данный ППР.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor dnp3

35.16.15.1 Команды конфигурации по умолчанию

alert wrong-crc

Эта команда предписывает предупреждать в случае неверной контрольной суммы в кадрах канального уровня для DNP3-пакетов.

ports {PORT,4}

Эта команда устанавливает порты DNP3-трафика.

35.16.16 Препроцессор GTP

Позволяет обнаруживать и предупреждать GTP-атаки. Протокол GTP (протокол туннелирования GPRS) используется в комуникационных сетях для создания каннала между различными GSN (обслуживающими узлами GPRS). Если сеть не содержит GPRS-устройств, то не нужно использовать данный ППР.

Цели ППР:

- отслеживание попыток вторжения в централизованную часть подсистемы GPRS через GTP-протокол.
- извелечение IP/TCP/UDP пакета из GTP-пакета: обнаружение и правила работают так, как будто нет GTP-заголовка, т.к. он отрасывается.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor gtp

35.16.16.1 Команды конфигурации по умолчанию

ports {PORT,4}

Эта команда устанавливает порты GTP-трафика.

35.16.17 Препроцессор MODBUS

Позволяет обнаруживать и предупреждать MODBUS-атаки. Протокол MODBUS используется в SCADA-сетях. Если сеть не содержит MODBUS-устройств, то не нужно использовать данный ППР.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor modbus

35.16.17.1 Команды конфигурации по умолчанию

ports {PORT,4}

Эта команда устанавливает порты MODBUS-трафика.

35.16.18 Препроцессор IMAP

Позволяет обнаруживать и предупреждать IMAP-атаки.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor imap

35.16.18.1 Команды конфигурации по умолчанию

ports {PORT,4}

Эта команда устанавливает порты IMAP-трафика.

alert <base64-decoding |bitenc-decoding| qp-decoding |uu-decoding> <all|none|N>

Эта команда предписывает предупреждать о неуспешных попытках декодирования или извлечения MIME-вложений.

Параметры:

- base64-decoding - BASE-64-декодирование MIME-вложений;
- bitenc-decoding - извлечение незакодированных MIME-вложений;
- qp-decoding - Quoted-Printable-декодирование MIME-вложений;
- uu-decoding - Unix-to-Unix-декодирование вложений;
- all - неограниченное потребление памяти для декодирования;
- none - не декодировать;
- N - объем памяти в байтах для декодирования.

35.16.19 Препроцессор POP

Позволяет обнаруживать и предупреждать POP-атаки.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor pop

35.16.19.1 Команды конфигурации по умолчанию

ports {PORT,4}

Эта команда устанавливает порты POP-трафика.

alert <base64-decoding |bitenc-decoding| qp-decoding |uu-decoding> <all|none|N>

Эта команда предписывает предупреждать о неуспешных попытках декодирования или извлечения MIME-вложений.

Параметры:

- base64-decoding - BASE-64-декодирование MIME-вложений;
- bitenc-decoding - извлечение незакодированных MIME-вложений;
- qp-decoding - Quoted-Printable-декодирование MIME-вложений;
- uu-decoding - Unix-to-Unix-декодирование вложений;
- all - неограниченное потребление памяти для декодирования;
- none - не декодировать;
- N - объем памяти в байтах для декодирования.

35.16.20 Препроцессор SIP

Позволяет обнаруживать и предупреждать SIP-атаки. Протокол SIP - протокол уровня приложений для создания,модификации и завершения сессий Internet-телефонии,видео-трансляций или видео-конференций с одним или более участниками.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor sip

35.16.20.1 Команды конфигурации по умолчанию

ports {PORT,4}

Эта команда устанавливает порты SIP-трафика.

alert <uri-len|call-id-len|req-name-len|from-len|to-len|via-len|contact-len|content-len> <N>

Эта команда предписывает предупреждать о превышении максимальной длины следующих параметров:

- uri-len - URI;
- call-id-len - поля CallID;
- req-name-len - имени запроса;
- from-len - поля From;
- to-len - поля To;
- via-len - поля Via;
- contact-len - поля Contact;
- content-len - содержимого.

Длина N задается в байтах.

no inspection media-data

Эта команда предписывает не проверять аудио- и видеоданные в SIP-трафике.

limit <sessions|dialogs> <N>

Эта команда задает максимаальное число:

- sessions - сессий (по умолчанию 10000);
- dialogs - диалогов (по умолчанию 4).

inspection methods {METHOD,14}

Эта команда задает методы SIP, для которых следует проверять SIP-сообщения.

Поддерживаются следующие методы: invite,cancel,ack,bye,register,options,refer,subscribe,date,join,info,

35.16.21 Препроцессор REP

Позволяет отбросить, запретить или разрешить трафик с указанных IP-адресов.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor rep

35.16.21.1 Команды конфигурации по умолчанию

permit <IP[/MSK]>

Эта команда определяет разрешенный IP-адрес источника или сеть.

deny <IP[/MSK]>

Эта команда определяет запрещенный IP-адрес источника или сеть.

nested-ip <inner|outer|both>

Эта команда определяет, какой именно IP-адрес проверять в случае IP-инкапсуляции:

- inner - внутренний (IP источника пакета, который является вложенным);
- outer - внешний (IP источника пакета, в который вложен другой IP-пакет);
- both - оба.

35.16.22 Препроцессор SSLTLS

Позволяет обнаруживать и предупреждать SSL/TLS-атаки. Данный ППР:

- декодирует SSL и TLS трафик;
- опционально определяется, нужно ли инспектровать данный трафик.

Для входа в режим конфигурации препроцессора следует ввести команду:

preprocessor ssltls

35.16.22.1 Команды конфигурации по умолчанию

ports {PORT,8}

Эта команда устанавливает порты SSL- и TLS-трафика. По умолчанию ППР знает о следующих портах SSL- и TLS-трафика:

- 443 HTTPS;
- 465 SMTPS;
- 563 NNTPS;
- 636 LDAPS;
- 989 FTPS;
- 992 TelnetS;
- 993 IMAPS;
- 994 IRCS;
- 995 POPS.

no inspection encrypted-traffic [trust-servers]

Эта команда предписывает не проверять зашифрованный трафик. Будет проверяться только SSL-рукопожатие. Далее, как только обнаруживается зашифрованный трафик со стороны клиента и сервера после SSL-рукопожатия, дальнейшие проверки не осуществляются.

Если же одна из сторон во время создания сессии (например,во время SSL-рукопожатия) сообщила об ошибке, сессия считается не зашифрованной.

Сессия считается зашифрованной, если зашифрованные данные замечены с обеих сторон (клиента и сервера). Однако, в случае опции trust-servers, это требование снимается и данные предполагаются зашифрованными, будучи зарегистрированными только на стороне клиента (инициатора SSL/TLS сессии) - это используется, если можно доверять соответствующим серверам и твердо известно, что они не были скомпроментированы. В результате можно получить прирост производительности ППР.

35.16.23 Препроцессор нормализации

Данный ППР осуществляет нормализацию пакетов в режиме mode ips, что помогает снизить шансы вторжения.

Для входа в режим конфигурации препроцессора введите команду:

preprocessor norm

35.16.23.1 Команды конфигурации по умолчанию

icmp

Эта команда включает нормализацию ICMP-протокола:

- очистка поля Код в Echo-запросах и Echo-ответах.

tcp <ips|urp|trim|allow-option <sack|echo|partial-order|conn-count|alt-checksum|md5|OPT> >

Эта команда включает нормализацию TCP-протокола:

- base - базовая нормализация TCP:
  - очистка зарезервированных флагов;
  - очистка Указателя срочности, если Флаг срочности не установлен и наоборот;
  - очистка Указателя срочности и Флага срочности, если в пакете нет данных (полезной нагрузки TCP);
  - очистка дополняющих байт TCP-опций;
- ips - отбрасывать сегменты, которые не могут быть верно пересобраны;
- urp - не уменьшать указатель важности, если он превышает данные;
- trim - удалять данные в SYN- и RST-пакетах, урезать данные до границ TCP-окна и MSS;
- allow-option - не очищать указанные опции:

  - sack - SACK опции (4,5);
  - echo - echo опции (6,7);
  - partial-order - опции частичного порядка (9,10);
  - conn-count - счетчик соединения (11,12,13);
  - alt-checksum - опции альтернативной чексуммы (14,15);
  - md5 - опции MD5 (19);
  - OPT - опция, заданная числом.

tcp ecn <packet|stream>

Эта команда включает нормализацию TCP-протокола в части контроля за ECN-флагами:

- packet - очищать ECN-флаги в пакетах;
- stream - очищать ECN-флаги ,если их использвание не было согласовано.

ip <base [MIN_TTL <NEW_TTL>] |df|rf|tos|trim>

Эта команда включает нормализацию IP-протокола:

- base - базовая нормализация IP:
  - нормализация TTL;
  - очистка поля TOS;
  - все октеты полей опций устанавливаются в NOP (НетОперации);
- MIN_TTL : минимальный TTL пакета;
- NEW_TTL : новый TTL пакета;
- df - очищать бит DontFragment входящих пакетов;
- rf - очищать Зарезервированный бит входящих пакетов;
- tos - очищать бит TOS (Тип обслуживания) входящих пакетов;
- trim - обрезать пакеты с избыточными данными до длины, равной сумме длины заголовка канального уровня и размера датаграммы ,указанного в IP заголовке; однако не обрезать меньше,чем минимальный размер кадра.

Нормализация TTL осуществляется следующим образом: если служба получает пакет с TTL < MIN_TTL, то TTL пакета устанавливается в NEW_TTL.

35.17 Настройки подсистемы обнаружения

35.17.1 Общие настройки

Тип команд данной секции: глобальный.

detection search-method <ac|ac-bnfa|lowmem> [non-queue|split|no-stream-inserts|max-pattern-len]

Эта команда задает алгоритм поиска в пакетах по образцу содержимого, указанного в правилах:

- ac - полный Aho-Corasick: большие затраты памяти, большая производительность;
- ac-bnfa - бинарный Aho-Corasick: средние затраты памяти, средняя производительность;
- lowmem - низкие затраты памяти, низкая/средняя производительность.

Дополнительные параметры:

- non-queue - найденное алгоритмом поиска не кладется в очередь;
- split - правило с портами any/any источника/назначения добавляется к каждому не-any/any правилу,таким образом увеличивая затраты памяти и увеличивая производительность; если правил any/any много, затраты памяти могут быть значительными; данная опция позволяет не добавлять каждое any/any правило к каждому не any/any, тем самым уменьшая затраты памяти и производительность
- no-stream-inserts - не анализировать пакет, который должен быть пересобран: увеличение производительности;
- max-pattern-len - максимальная длина образца в алгоритме поиска по образцу: может увеличить производительность и уменьшить затраты памяти.

По умолчанию: detection search-method ac-bnfa non-queue

detection max-queue-events <VAL>

Эта команда устанавливает максимальное число событий для правила, которые помещаются в очередь до момента выбора события.

По умолчанию: 5

detection port-limit-any <VAL>

Эта команда устанавливает максимальное число портов источника и назначения, которые должны быть в правиле, чтобы признать его правилом с портами any/any.

По умолчанию: 1024

detection debug <TYPE>

Эта команда задает уровень отладочной информации подсистемы обнаружения. Параметр TYPE может быть следующим:

- all - все типы отладочной информации;
- port-group-evaluation - информация о портах во время оценки пакета;
- fast-patterns - информация о быстрых образцах, используемых для данной группы портов;
- port-group-compilation - инфомация о портах во время создания групп портов;
- port-group-compiled - информация о созданных группах портов;
- port-group-uncompiled - информация о несозданных группах портов;
- content-fast-pattern - если в правиле используются быстрые образцы, вывести информацию о содержании, использованном в анализе;
- port-limit-any-exceed - предупреждать о превышении detection port-limit-any-значения для правила.

35.17.2 Настройка группы правил

Отдельные правила сгруппированы по группам правил.

Типы групп правил:

- системная - правила данной группы можно отключать и включать:
  - личные данные (ruleset private-data) - группа правил обнаружения атак на личные данные;
  - препроцессор (ruleset preprocessor) - группа правил подсистемы препроцессинга;
  - декодер (ruleset decoder) - группа правил подсистемы декодирования;
  - обнаружение (ruleset detection) - группа подгрупп правил подсистемы обнаружения всех прочих атак;
- пользовательская (ruleset manual) - группа пользовательских подгрупп правил; правила данных подгрупп можно создавать,удалять, отключать и включать.

Идентификация отдельного правила системной группы осуществляется по номерам GID и SID, которые можно узнать при просмотре списка правил с помощью команды show service ids rules. Для группы правил обнаружения (ruleset detection) в списке правил не указывается их GID - это значит, что он по умолчанию равен 1.

ruleset <decoder|preprocessor|private-data|detection <NAME>|manual <NAME> >

По этой команде осуществляется вход в режим конфигурации группы правил декодера, препроцессора, личных данных или вход в режим конфигурации подгруппы правил прочего обнаружения (ruleset detection NAME) и пользовательского обнаружения атак (ruleset manual NAME).

Параметр NAME задает:

- имя подгруппы для группы правил обнаружения; может быть строго определенным, например: smtp - подгруппа правил обнаружения SMTP-атак , dns - подгруппа правил обнаружения DNS-атак;
- имя подгруппы для группы пользовательских правил; это имя выбирается пользователем.

Далее, для простоты, будем подгруппу (для ruleset detection и ruleset manual) называть группой.

35.17.2.1 Настройка правил системной группы

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 ids 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 ids 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;
- reject : записать пакет в журнал и одновременно запретить его; дополнительно послать "TCP reset" (для TCP) или "ICMP port unreachable" (для UDP);
- sdrop : запретить пакет.

35.17.2.2 Настройка правил пользовательской группы

В этом случае в дополнение к командам включения и выключения правил (см. Настройка правил системной группы), 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 tcp any any —> any any ( gid:8001; sid:1000001; rev:1; classtype:unknown; )
```

35.17.3 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;
- 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).

35.17.3.1 Управление пользовательским правилом

on

Эта команда включает текущее пользовательское правило (по умолчанию правило и так включено).

off

Эта команда выключает текущее пользовательское правило.

35.17.3.2 Заголовок пользовательского правила

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>

Эта команда задает тип протокола. Параметр может быть следующим: tcp,udp,ip,icmp.

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.

35.17.3.3 Опции для описания пользовательского правила

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.

35.17.3.4 Опции для проверки содержимого пакета

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 симолов 0..9 ,a..f, заключенных между 2мя симолами «*»;
- 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.

35.17.3.4.1 nocase Это значение параметра задает нечувствительный к регистру поиск; «payload content abc nocase» будет искать, например, такие строки как abc, ABC, abC и др.

35.17.3.4.2 rawbytes При этом значении параметра правило будет рассматривать только чистые, неизмененные фазой декодирования и нормализации, данные пакета: такие ППР, как telnet, rpc, smtp по умолчанию нормализуют данные перед поиском по образцу. Если необходимо анализировать чистые, неизмененные данные - следует использовать данную опцию.

35.17.3.4.3 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.

35.17.3.4.4 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.

35.17.3.4.5 match-offset <PAT2> < min <VAL1|BEVAL1> | <max <VAL2|BEVAL2> > Параметр VAL1 задает смещение относительно смещения последнего найденного образца, с которого начинать поиск образца PAT2. Т.е. VAL1 это то же самое, что и offset, но смещение задается не относительно начала данных пакета, а относительно смещения последнего найденного образца.

Параметр VAL2 задает максимальное смещение относительно смещения последнего найденного образца, до которого следует осуществлять поиск образца PAT2.

Т.е. данной командой мы задаем интервал поиска образца PAT2, относительно смещения последнего найденного образца.

Рассмотрим пример: payload content ABC match-offset DEF min 10 max 20.

Данная команда ищет по образцу /ABC.{10}DEF/, причем DEF ищется в первых 20 байтах после ABC,но не далее.

Аргументы для max и min:

- VAL - число от -65535 до 65535;
- BEVAL - имя заданного в данном правиле byte-extract параметра.

35.17.3.4.6 http-client-body Этот параметр предписывает вести поиск только в неизменном HTTP-запросе клиента. Размер данных, проверяемых в запросе, определяется опцией HTTP-препроцессора ‘inspect client post’.

35.17.3.4.7 http-cookie Этот параметр предписывает вести поиск только в извлеченных Cookie-заголовках (кроме имен заголовков и переносов строк в конце Cookie-заголовка) HTTP-запроса клиента и HTTP-ответа сервера.

Особенности применения данного параметра:

- будет работать,только если включена опция HTTP-препроцессора ‘inspect cookies’;
- извлеченный Cookie может быть нормализован,если включена alert normalize cookies в HTTP-препроцессоре.

35.17.3.4.8 http-raw-cookie Этот параметр предписывает вести поиск только в извлеченных Cookie-заголовках (кроме имен заголовков и переносов строк в конце Cookie-заголовка) HTTP-запроса клиента и HTTP-ответа сервера. Особенности применения данного параметра:

- будет работать,только если включена опция HTTP-препроцессора ‘inspect cookies’;
- извлекается не нормализованный Cookie.

35.17.3.4.9 http-header Этот параметр предписывает вести поиск только в HTTP-заголовках HTTP-запроса клиента и HTTP-ответа сервера.

HTTP-заголовки могут быть нормализованы, если включена опция HTTP-препроцессора alert normalize headers.

35.17.3.4.10 http-raw-header Этот параметр предписывает вести поиск только в ненормализованных HTTP-заголовках HTTP-запроса клиента и HTTP-ответа сервера.

35.17.3.4.11 http-method Этот параметр предписывает вести поиск только в поле Method HTTP-запроса.

35.17.3.4.12 http-uri Этот параметр предписывает вести поиск только в нормализованных данных поля URI HTTP-запроса.

35.17.3.4.13 http-raw-uri Этот параметр предписывает вести поиск только в ненормализованных данных поля URI HTTP-запроса.

35.17.3.4.14 http-stat-code Этот параметр предписывает вести поиск только в данных поля Status HTTP-ответа сервера.

35.17.3.4.15 http-stat-msg Этот параметр предписывает вести поиск только в данных поля Status Message HTTP-ответа сервера.

35.17.3.4.16 fast-pattern [only | <OFF>,<LEN>] Этот параметр предписывает активировать быстрый поиск по образцу - позволяет ускорить поиск образца. Дополнительные параметры:

- only - содержимое PAT используется только в механизме быстрого поиска в содержимом, и не учитываются остальные опции правила. Полезно, если заранее известно ,что образец должен находится где-либо в данных пакета (не важно, где именно).
- OFF,LEN - задает смещение и длину части образца для поиска. Полезно, если образец слишком длинный, чтобы сохранить память в механизме быстрого поиска.

35.17.3.4.17 [PRIO] payload isdataat [!]<VAL> relat 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.

35.17.3.4.18 [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).

35.17.3.4.19 [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.

35.17.3.4.20 [PRIO] payload detbuf <transport|application|sip-header|sip-body|dce-stub-data|m
Этот параметр устанавливает курсор обнаружения на :

- transport - данные транспортного уровня пакета;
- application - данные уровня приложения (application);
- 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 (ППР dnp3) фрагмента уровня приложения.

Будучи установленным, курсор будет действовать на все опции правила, заданные после него.

[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.

payload cvs

По этой команде осуществляется детектирование атак на CVS. Порты CVS должны быть включены в ППР stream.

payload http-encode <uri|header|cookie> [no] encoding-types <{utf8,double-encode,non-ascii,ascii,iis-encode,uencode,bare-byte}>

Эта команда предписывает предупреждать об использовании кодирования в запросе или ответе.

Место поиска:

- uri - URI-поле запроса клиента;
- header - HTTP-заголовки запроса или ответа;
- cookie - Cookie-поля HTTP-запроса или ответа.

Тип кодирования:

- utf8 - UTF8-кодирование;
- double-encode - double-кодирование;
- non-ascii - не ASCII-кодирование;
- ascii - ASCII-кодирование;
- iis-encode - IIS-Unicode-кодирование;
- uencode - u-кодирование;
- bare-byte - bare-byte-кодирование.

Прочие параметры:

- no - предупреждать,если используемое кодирование не входит в перечисленный список кодирований.

payload urilen < <more|less LEN0 > | <LEN0 LEN1> > [raw|norm]

Эта команда задает ограничения на длину URI и тип буфера, в котором осуществляется проверка длин URI. Параметры:

- LEN0 - максимальная длина URI;
- LEN1 - минимальная длина URI;
- more LEN0 - длина больше, чем LEN0;
- less LEN0 - длина больше, чем LEN0;
- raw - использовать ненормализованный буфер (по умолчанию);
- norm - использовать нормализованный буфер.

payload ftpbounce

Эта команда задает обнаружение FTP-bounce атак.

35.17.3.5 Препроцессоро-зависимые опции правил

Данные опции доступны, если включен соответсвующий ППР, указанный в описании опции.

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.

Замечание: должен быть включен ППР ssltls.

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.

Замечание: Должен быть включен ППР ssltls.

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.

Замечание: Должен быть включен ППР dcerpc.

payload priv-pattern <credit-card|us-ssn|us-ssn-nodash|email> <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;
- email - адреса электронной почты;
- COUNT - число обнаружений личных данных до генерации предупреждения.

Замечание: Должен быть включен ППР priv.

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.

Замечание: Должен быть включен ППР sip.

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).

Замечание: Должен быть включен ППР sip.

payload sip-header {CODE|CODE1-CODE2,4}

Эта команда задает тип отслеживаемых SIP-кодов ответа (чсила от 100 до 999, или интервалы таких чисел).

Замечание: Должен быть включен ППР sip.

payload gtp version <VAL>

Эта команда предписывает проверять на указанную версию протокола GTP.

Замечание: Должен быть включен ППР gtp.

payload gtp info <VAL>

Эта команда предписывает проверять на наличие указанного информационного элемента GTP.

Замечание: Должен быть включен ППР gtp.

payload gtp type {VAL,4}

Эта команда предписывает проверять на наличие указанных типов сообщений GTP.

Замечание: Должен быть включен ППР gtp.

payload modbus func <VAL>

Эта команда предписывает проверять на наличие указанного значения Function code в заголовке Modbus-протокола.

Замечание: Должен быть включен ППР modbus.

payload modbus unit <VAL>

Эта команда предписывает проверять на наличие указанного значения Unit ID в заголовке Modbus-протокола.

Замечание: Должен быть включен ППР modbus.

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.

Замечание: Должен быть включен ППР dnp3.

payload dnp3 func <VAL>

Эта команда предписывает проверять на наличие указанного значения Function code в заголовке DNP3-протокола.

Замечание: Должен быть включен ППР dnp3.

payload dnp3 obj <GROUP> <OVAR>

Эта команда предписывает проверять на наличие указанных объектных заголовоков в DNP3-протоколе.

Параметры:

- GROUP - группа объекта;
- VAR - объект.

Замечание: Должен быть включен ППР dnp3.

35.17.3.6 Опции для проверки заголовка пакета

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 ППР stream.

header frag-offset [LOGIC] <VAL>

Эта команда задает ограничение на значение смещения фрагмента в IP-заголовке. Параметры:

- LOGIC - задает оператор сравниния со значением VAL: not,more,less.

Например, для того чтобы в правило попадали все первые фрагменты IP-сессии, следует выполнить следующие команды:

```
header fragbits more—frags
header fragoffset 0
```

header ttl <[LOGIC] VAL0 | VAL0 VAL1>

Эта команда задает ограничение на значение TTL в IP-заголовке. Параметры:

- LOGIC - задает оператор сравнения со значением VAL0: more,less,equal,more-equal,less-equal;
- VAL0,VAL1 - интервал значений.

header tos [LOGIC] VAL

Эта команда задает ограничение на значение TOS в IP-заголовке. Параметры:

- LOGIC - задает оператор сравнения со значением VAL: not;
- VAL - значение TOS.

header id VAL

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 [LOGIC] VAL0 [VAL1]

Эта команда задает обнаружение пакетов с данными заданной длины.

Параметры:

- LOGIC - задает оператор сравнения со значением VAL: not;
- VAL0 VAL1 - интервал значений (в байтах)

header seq VAL

Эта команда предписывает проверять значение номера последовательности TCP-пакета (TCP Sequence number).

header ack VAL

Эта команда предписывает проверять значение номера уведомления TCP-пакета (TCP Acknowledge number).

header window-size [not] VAL

Эта команда предписывает проверять значение размера окна TCP-пакета. Параметр not - проверять,что пакет не имеет заданный размер окна.

header icmp-code < more|less VAL0 | VAL0 VAL1>

Эта команда предписывает проверять значение кода ICMP-ответа. Параметры:

- LOGIC - задает оператор сравнения со значением VAL0: more,less;
- VAL0,VAL1 - интервал значений.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-type < more|less VAL0 | VAL0 VAL1>

Эта команда предписывает проверять значение типа сообщения ICMP. Параметры:

- LOGIC - задает оператор сравнения со значением VAL0: more,less;
- VAL0,VAL1 - интервал значений.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-id VAL

Эта команда предписывает проверять значение идентификатора сообщения ICMP.

Данная опция может быть задана только при icmp-типе протокола в правиле.

header icmp-seq VAL

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 sameip

Эта команда предписывает проверять на совпадение IP-адресов получателя и отправителя.

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 ППР 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 позовляет 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.

35.18 Подсистема фильтров частоты

Подсистема фильтрации по частоте предотвращает атаки, основанные на частоте обнаружения некоторого действия. Пользователь может сконфигурировать службу так, что при достижении некоторого события(правила) заданной частоты обнаружения осуществляется выполнения нового действия. Для каждого правила можно сконфигурировать множество фильтров частоты: будет использован первый найденный для правила фильтр частоты.

rate-filter <GID> <SID> <src|dst|rule> <COUNT/SEC> <ACT> <TO> [nets {IP/MSK,4}]

Параметры:

- GID,SID - идентификаторы подсистемы и правила, для которых применяется данная команда;
- src - счетчик событий COUNT; ведется по числу уникальных IP-адресов источника;
- dst - счетчик событий COUNT; ведется по числу уникальных IP-адресов назначения;
- rule - счетчик событий COUNT; ведется по числу срабатывания правила;
- COUNT/SEC - счетчик событий за интервал времени SEC(в секундах);
- SEC - интервал времени в сек., за который считается счетчик событий; если он равен 0 , то COUNT - общее число событий;
- 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, которе начало использоваться в результате срабатывания данного фильтра частоты; если равно 0, то не будет производиться возврат к старому действию;
- nets - список сетей источника(если задан параметр src) или назначения(если задан параметр dst), к которым применяется фильтр.

35.19 Подсистема контроля задержек

Тип команд данной секции: глобальный.

Данная подсистема представляет собой механизм пороговых значений, который может использорваться для контроля над задержками в работе службы. Проверка на задержки осуществляется для пакетов и правил. Можно также указать действие,применяемое, если обнаружена чрезмерная задержка.

latency packet <LAT> [skip-slow] [log] [stats]

Параметры:

- LAT - максимальная задержка в микросекундах; примерные значения: 100/250/1000 для 1G/100M/5M сетей;
- skip-slow - пропускать дальнейший анализ пакета, задержка которого превысила максимальную;
- log - регистрировать в журнале пакеты, задержка которых превысила максимальную;
- stats - писать в журнал статистику по каждому пакету.

latency rule <LAT> [N] [skip-slow [TIME]] [log] [log-alert]

Параметры:

- LAT - максимальная задержка в микросекундах; примерные значения: 100/250/1000 для 1G/100M/5M сетей;
- N - число превышений максимальной задержки правила, после которого данное правило перестает использоваться (по умолчанию: 5);
- skip-slow - останавливать дальнейший анализ правила, задержка которого превысила максимальную;
- TIME - время в секундах, на которое останавливается анализ правила, задержка которого превысила максимальную (в случае опции skip-slow); по умолчанию: 60сек;
- log - регистрировать в журнале правила, задержка которых превысила максимальную;
- log-alert - записывать в журнал предупреждения правил, задержка которых превысила максимальную.

35.20 Подсистема событий

Подсистема событий отвечает за:

- частоту журналирования обнаружений: для уменьшения частоты записи в журнал предупреждений для каких-либо или всех правил;
- подавление правил в зависимости от трафика входящего и/или исходящего на указанные IP-адреса;
- сортировку событий и максимальное число событий, которое может зарегистрировано или отслежено для каждого пакета в каждый момент времени.

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 filter block <GID> <SID> <src|dst> {IP[/MSK],4}

Данная команда позволяет указать события, о которых не нужно предупреждать. Т.е. правило может быть включено, но предупреждения выдавать оно не будет.

Команда предназначена для уменьшения числа ложных предупреждений.

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;
- src - счетчик событий COUNT ведется по числу уникальных IP-адресов источника;
- dst - счетчик событий COUNT ведется по числу уникальных IP-адресов назначения;
- IP/[MSK] - список IP-адресов, для которых применяется команда подавления предупреждений.

event queue <max-events|max-to-log|order-events <priority|content-length> >

Эта команда задает параметры очереди событий. Параметры:

- max-events - размер очереди событий; задает максимальное число событий, запоминаемых для каждого пакета/сессии;
- max-to-log - задает максимальное число событий, регистрируемых в журнале для каждого пакета/сессии;
- order-events - задает тип сортировки событий в очереди; priority - по важности; content-length - по длине параметра опции content-правила.

Тип: глобальный

35.21 Статистика работы службы

Тип команд данной секции: глобальный.

Имеется возможность вести статистику работы службы. Измеряются показатели работы реального времени и теоретически достижимые максимальные показатели работы службы.

perfmon [flow] [flow-ip] [events] [TIME]

Параметры:

- flow - статистика о наблюдаемом типе трафика и протоколах;
- flow-ip - для каждой пары хостов ведется следующая статистика:
  - TCP Packets - число TCP-пакетов;
  - TCP Traffic in Bytes - объем TCP-трафика в байтах;
  - TCP Sessions Established - число установленных TCP-сессий;
  - TCP Sessions Closed - число закрытых TCP-сессий;
  - UDP Packets - число UDP-пакетов;
  - UDP Traffic in Bytes - объем UDP-трафика в байтах;
  - UDP Sessions Created - число созданных UDP-сессий;
  - Other IP Packets - число других IP-пакетов;
  - Other IP Traffic in Bytes - другой IP трафик в байтах;
- events - статистика отношения числа событий, которые не выполнились (правило анализировалось и не сработало) к числу событий, которые выполнились (правило анализировалось и сработало);
- TIME - время между интервалами (в сек.).

Для просмотра данной статистики следует использовать команду:

show service ids log <perfmon|perfmon-flowip>

Максимальные размеры perfmon-журналов - 5Мб. Максимальное число архивов журналов perfmon - 2.

stat rules [N] [SORT]

Эта команда включает ведение статистики по производительности подсистемы обнаружения.

Параметры:

- N - число самых «медленных» правил, которые записываются в журнал статистики;
- SORT - порядок сортировки:
  - checks - по числу проверок правила;
  - matches - по числу попаданий в правило;
  - nomatches - по числу непопаданий в правило;
  - avg-ticks - по среднему числу процессорных тактов;
  - avg-ticks-per-match - по числу процессорных тактов на попадание;
  - avg-ticks-per-nomatch - по числу процессорных тактов на непопадание;
  - total-ticks - по общему числу процессорных тактов.

Чтоб посмотреть статистику по производительности, следует остановить службу и выполнить команду: **show service ids stat-rules**.

stat preprocs [N] [SORT]

Эта команда подключает ведение статистики по производительности препроцессоров.

Параметры:

- N - число самых «медленных» препроцессоров, которые записываются в журнал статистики;
- SORT - порядок сортировки:
  - checks - по числу проверок правила;
  - avg-ticks - по среднему числу процессорных тактов;
  - total-ticks - по общему числу процессорных тактов.

Чтоб посмотреть статистику по производительности препроцессоров, нужно остановить службу и выполнить команду: **show service ids stat-preprocs**

35.22 Журналы

Тип команд данной секции: глобальный.

log <local>

Эта команда включает ведение журналов предупреждений в режиме:

- local: журналы ведутся в локальной файловой системе.

log shmem <SUS_IP> [size <SZ>] [port <PORT>] [name <SNAM>][ID] [packet]

Эта команда включает режим журналирования в разделяемую память и одновременно переключает систему на работу с правилами ЦНИИ ЭИСУ. Особенности данной работы состоят в следующем:

- система переключается на cni-правила (это другой,особый набор системных правил, который может отличаться от std-правил);
- все команды изменения действия системного правила, включения и отключения системных правил, а также команды создания собственных правил применяются **только** к std-правилам и никак не применяются к cni-правилам;
- команды включения и отключения ruleset-ов применяются к cni-правилам - можно как и для std-правил включать только определенный набор ruleset-ов;
- команда обновления локальных правил service ids rules upgrade/downgrade работает только для std-правил и никак не действует на cni-правила;
- cni-правила обновляются автоматически.

Параметры команды:

- SUS_IP - IP-адрес сервера-приемника предупреждений;
- SZ - размер разделяемой памяти для хранения предупреждений (15мб по умолчанию);
- PORT - номер порта сервера-применика предупреждений (1101 по умолчанию);
- SNAM - название сенсора (sensor1 по умолчанию);
- ID - идентификатор сенсора (1 по умолчанию);
- packet - следует передавать данные пакета, на который выдано предупреждение.

log <BD> <IFACE> <SRV> [user <USR>] [password <PAS>] [PORT][brief]

Эта команда включает режим пересылки информации о предупреждениях на удаленный сервер в базу данных snortdb.

Параметры:

- BD : типа базы данных (пока поддерживается только postgresql);
- IFACE : интерфейс, с которого отправляются данные на сервер баз данных (тип ethernet или bond);
- SRV : адрес удаленного сервера базы данных;
- PORT : порт сервера SRV, на котором ожидает запросов база данных (по умолчанию - 5432);
- USR : пользователь (по умолчанию - snort);
- PAS : пароль (по умолчанию - pass);
- brief : краткая информация (по умолчанию - полная).

35.23 Просмотр информации

show service ids status

Эта команда выводит следующую информацию:

1. статус сервиса: включен или выключен;
2. версию правил.

show service ids statistics

Эта команда показывает статистику сервиса.

show service ids log [alerts|portscans|perfmon|perfmon-flowip]

Эта команда показывает следующие журналы:

- alerts - разные предупреждения;
- portscans - предупреждения о сканировании портов;
- perfmon - показывает базовую статистику perfmon;
- perfmon-flowip - показывает статистику perfmon flow-ip.

Максимальные размеры журналов portscans и perfmon - 5Мб. Максимальное число архивов журналов portscans и perfmon - 2 (текущий и предыдущий).

Если тип журналов не задан - показывает журнал службы.

show service ids ipvars [VIEW]

Эта команда показывает переменные ipvars для конфигурации по умолчанию или для вида VIEW, если он указан. Все имена переменных будут показаны заглавными буквами, хотя в конфигурации они определены строчными.

show service ids portvars [VIEW]

Эта команда показывает переменные portvars для конфигурации по умолчанию или для вида VIEW,если он указан. Все имена переменных будут показаны заглавными буквами, хотя в конфигурации они определены строчными.

show service ids rules <decoder|private-data|preprocessor|detection <NAME>|manual <NAME> > [disabled|enabled] [VIEW]

Эта команда показывает правила соответствующей группы правил для конфигурации по умолчанию или для вида VIEW,если он указан.

Параметры:

- disabled - отключенные правила;
- enabled - включенные правила.

В начале правила будет показан его номер. Этот номер может использоваться для поиска правила после выдачи команды show service ids rule by-message:

show service ids rule by-message <MSG> [VIEW]

Эта команда ищет правило по сообщению предупреждения и ,если оно найдено, выдает название списка правил и номер правила в этом списке.

35.24 Обновление правил

Обновление правил службы возможно двух типов:

- локальное, с помощью флэшки - происходит обновление std-правил;
- удаленное, когда служба работает с CNII-правилами (в конфигурации присутствует команда log shmem) - происходит обновление cni-правил.

service ids rules upgrade <DEV>

По этой команде выполняется обновление встроенных системных правил из указанного архива. Тип архива должен быть gz или bz2.

В случае успешного выполнения обновления правил будет выведено сообщение (пример):

```
Info: Rules successfully upgraded to version 2013.05.23 (type: —std)
```

Перед обновлением осуществляются следующие проверки архива правил:

- должно быть наличие файла preproc_rules/sensitive-data.rules;
- должно быть наличие файла preproc_rules/decoder.rules;
- должно быть наличие файла preproc_rules/preprocessor.rules;
- должно быть наличие в директории etc следующих файлов: classification.config, reference.config, sid-msg.map, snort.conf, threshold.conf, unicode.map;
- должно быть наличие в директории rules хотя бы одного файла с расширением rules (например: rules/x11.rules).

service ids rules downgrade

Эта команда выполняет отмену обновления встроенных системных правил из указанного архива к правилам, которые были изначально в данном релизе.

В случае успешного выполнения отката обновления правил будет выведено сообщение (пример):

```
Info: Rules successfully downgraded to version 2012—04—13.12:01:15 (type: —std)
```

updatecheck <SRV> [TO]

Данная команда осуществляет проверку наличия более свежих правил (только std-правила), чем те, что в системе уже установлены.

Параметры -

- SRV - это URL директории на сервере,где хранятся архивы правил.
- TO - период проверки (в часах, от 1 до 23).

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.

35.25 Примеры конфигураций

35.25.1 Пример 1.

Обнаружение ICMP-Echo-сообщения (пинг) из сети 192.168.33.0/24 .

```
iface ethernet 3
mode ids
log local
preprocessor frag
preprocessor stream
ruleset manual my
 1 rule r
  action alert
  proto icmp
  src 192.168.33.0/24
enable
```

35.25.2 Пример 2.

Обнаружение атак внутри текущей сети.

```
service ids
 iface ethernet 3
 mode ids
 log local
 1 ipvar home—net 192.168.33.0/24
 preprocessor frag
  instance my
   detect—anomalies
   policy type linux
 preprocessor stream
  track—sessions icmp yes
  track—sessions ip yes
  instance—tcp default
   detect—anomalies
   policy linux
   session—hijacking
 preprocessor portscan
  detect—ack—scans
  log
  protocol all
  sense medium
  type all
 preprocessor priv
 preprocessor http
  alert proxy
  codepage 861
  instance default
   ports 80 8080
   profile apache
   alert limit dir—size 300
   alert limit req—headers 30
```

```
  alert decompression
  alert normalize headers
  alert normalize javascript 200
  alert normalize utf
  inspect client request 1000
  inspect cookies
  inspect methods connect put head
  inspect server response 2000
  log hostname
  log uri
  log xff
 instance iis
  match—ip 192.168.0.1 192.168.1.1
  ports 80 81 82
  profile iis5
  no alert proxy
ruleset decoder
ruleset detection icmp
ruleset preprocessor
ruleset private—data
enable
```

Описание примененной конфигурации:

- Защищаемая сеть 192.168.33.0/24 , внешняя сеть - любая;
- Режим ids - обнаружение вторжений;
- Препроцессоры:
  - stream (отслеживание всех сессий, аномалий TCP, атаки TCP MInM, linux политика);
  - frag (аномалии фрагментации,политика linux);
  - portscan (средняя чувствительность, ведение отдельного журнала препроцессора,все типы сканирования портов, ack-сканирование);
  - priv (частные данные);
  - http:
    * кодовая страница 861;
    * для вэб-серверов 192.168.0.1/1.1 на портах 80-82 задействован профиль IIS5; для всех остальных - профиль Apache на портах 80 и 8080;
    * предупреждения об использовании прокси-сервера для всех вэб-серверов, кроме 192.168.0.1/1.1 (профиль iis5); другие предупреждения и настройки;
- Журнал предупреждений: локальный, в файловую систему;
- Активированы все системные группы правил (часть из них по умолчанию отключена).

Тестирование работы:

- следует набрать с другой машины команду nmap -PN -v 192.168.33.xxx, где 192.168.33.xxx - адрес интерфейса ethernet3, который указан в службе ids;

- следует посмотреть журналы предупреждений при помощи команд show service ids log portscan/alerts;
- в журнале alerts должна появиться примерно такая запись: 02/07-18:26:09.293886 [* *] [122:1:1] <NULL> portscan: TCP Portscan [* *];
- в журнале portscan - такая:

```
Time: 02/07/13—18:26:09.293886
event_id: 51
192.168.33.144 —> 192.168.33.149 (portscan) TCP Portscan
Priority Count: 13
Connection Count: 15
IP Count: 1
Scanner IP Range: 192.168.33.144:192.168.33.144
Port/Proto Count: 15
Port/Proto Range: 21:8888
```

35.25.3 Пример 3.

```
iface ethernet 0
mode ids
1 ipvar home—net 192.168.33.0/24 83.1.2.3
2 ipvar external—net !home—net
daq snaplen 1514
perfmon events flow flow—ip 10
daq bufsize 32
decoder checksum calc all
decoder esp
decoder teredo
detection search—method ac—bnfa no—stream—inserts
log postgresql ethernet 2 192.168.33.160 full
preprocessor frag
 limit memory 1000 1000
 instance default
  detect—anomalies
  policy type linux
preprocessor stream
 limit memory 16000
 track—sessions icmp yes
 track—sessions ip yes
 track—sessions tcp yes
 track—sessions udp yes
 instance—tcp default
  policy linux
  session—hijacking
  timeout 30
preprocessor portscan
```

```
 detect—ack—scans
 log
 protocol all
 sense high
 type all
 watch 83.220.32.68 83.220.32.70 192.168.33.254
preprocessor rpc
 alert fragments
preprocessor http
 alert proxy
 limit compressed—size 5000
 limit memory decompress 1000
 limit memory log 3000
 instance default
  profile apache
  alert decompression
  alert normalize cookies
  alert normalize headers
  alert normalize javascript 100
  alert normalize utf
  log hostname
  log xff
preprocessor smtp
 inspect ignore—data
 inspect mode stateful
 ports 25
preprocessor ftptelnet
 alert encrypted—traffic yes continue
 alert telnet—anomalies
 alert telnet—ayt 50
 inspection mode stateful
 telnet—normalize
 telnet—ports 23
 instance—ftpserver default
  ports 21
  alert limit paramlen 256
 instance—ftpclient default
  alert bounce yes
  alert telnet—esc yes
preprocessor ssh
 ports 22
 alert bad—direction
 alert crc32—exploit
 alert invalid—payload—size
 alert max—client—bytes 19600
 alert max—version—length 80
 alert non—ssh—ports
 alert non—ssh—traffic
```

```
 alert protomismatch—exploit
 alert response—overflow—exploit
 alert seccrt—exploit
 inspection max—encrypted—packets 10
preprocessor dns
 ports 53
 alert experimental—rrs
 alert obsoleted—rrs
 alert rdata—overflow
preprocessor ssltls
preprocessor arp
 alert unicast
preprocessor priv
 alert private—data 10
preprocessor dcerpc
 alert all
 limit memory 10000
preprocessor norm
 icmp
 ip base
 tcp base
preprocessor rep
 limit memory 1000
ruleset decoder
ruleset detection backdoor
ruleset detection botnet—cnc
ruleset detection chat
ruleset detection ddos
ruleset detection dns
ruleset detection exploit
ruleset detection file—identify
ruleset detection file—other
ruleset detection ftp
ruleset detection icmp
ruleset detection icmp—info
ruleset detection misc
ruleset detection nntp
ruleset detection phishing—spam
ruleset detection policy
ruleset detection policy—other
ruleset detection policy—social
ruleset detection pua—p2p
ruleset detection rpc
ruleset detection scan
ruleset detection server—mail
ruleset detection shellcode
ruleset detection smtp
ruleset detection snmp
```

```
ruleset detection spyware—put
ruleset detection sql
ruleset detection telnet
ruleset detection tftp
ruleset detection web—activex
ruleset detection web—attacks
ruleset detection web—cgi
ruleset detection web—client
ruleset detection web—coldfusion
ruleset detection web—iis
ruleset detection web—misc
ruleset detection web—php
ruleset detection x11
ruleset preprocessor
ruleset private—data
enable
```

35.25.4 Примеры правил

Рассмотрим далее различные полезные правила.

35.25.4.1 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
```

35.26 Установка базы данных Postgres

В локальной сети, в которой находится со службой IDS, необходимо установить базу данных Postgres. В данную базу данных служба IDS 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 IDS):

```
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 IDS
- 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 IDS.
- /etc/init.d/postgresql restart : перезапустите postgresql

35.27 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 в ключите заново систему аутентификации в /var/www/base/base_main.php: Use_Auth_System = 1;

36. Служба IDSSUR

Система имеет службу предупреждения и предотвращения вторжений idssur. Основное отличие от службы IDS - большая производительность.

36.1 Режимы

Служба IDSSUR может работать в двух режимах:

- ids - система обнаружения вторжений;
- ips - система обнаружения и предупреждения вторжений.

36.2 Переменные

В службе есть понятие именованных переменных, которые могут быть двух типов:

- ipvar - именованная переменная, хранящая IP-адреса;
- portvar - именованная переменная, хранящая порты.

Существуют предопределенные наборы системных переменных обоих типов, имеющие некоторые стандартные значения, например: «portvar ssh-ports 22 1022», т.е. переменная с именем ssh-ports равная набору чисел 22,1022, определяющих набор SSH-портов. Смысл переменной приведен не в ее определении, а в определении правила, использующего ее.

Системные переменные, показываемые по авто-дополнению команды ipvar/portvar, используются в системном наборе правил. Их можно изменять, но нельзя удалять.

Пользовательские переменные, в свою очередь, могут создаваться,удаляться и изменяться пользователем по своему усмотрению. Они могут быть использованы в пользовательских правилах.

Для определения своей, либо переопределения системной именованной переменной следует использовать следующие команды.

portvar <NAME> <{[!]PORT1[:PORT2]|any|NAME2},8>

Эта команда задает переменную NAME, определяющую набор портов: NAME2 - другая, заранее определенная portvar-переменная, PORT1:PORT2 - интервал портов, PORT1 - отдельный порт, знак “!” - отрицание набора портов/порта.

ipvar <NAME> <{[!]IP1[:IP2]|any|NAME2},8>

Эта команда задает переменную NAME, определяющую набор IP-адресов. Формат аналогичен команде portvar.

36.3 Начальная настройка

Вначале необходимо определиться с методом перехвата трафика: nfq или af-packet.

active-capmode <nfq|afp>

Эта команда задает используемый в службе метод перехвата трафика. Метод afp работает быстрее, чем nfq.

Далее необходимо задать интерфейсы, трафик которых необходимо анализировать.

36.3.1 Режим NFQ

В данном режиме перехвата трафика необходимо задать пары интерфейсов, которые будут анализироваться.

Для входа в настройки режима NFQ выполните команду:

capmode nfq

Если не задать ничего, будет анализироваться весь трафик всех интерфейсов:

iface-pair <IFACE_IN> <IFACE_OUT>

Интерфейс IFACE_IN задает интерфейс входящего трафика, IFACE_OUT - исходящего. Это относится как к прямым, так и транзитным пакетам, приходящим на интерфейс IFACE_IN и маршрутизируемым на интерфейс IFACE_OUT.

36.3.2 Режим AFP

В конфигурации режима afp есть настройки по умолчанию и настройки для конкретных интерфейсов, трафик которых подлежит перехвату и анализу. Специфичные настройки интерфейсов переопределяют настройки по умолчанию.

Для входа в настройки режима AFP выполните команду:

capmode afp

Для выхода в режим настройки интерфейса следует выполнить команду:

iface <IFACE>

Параметр IFACE задает интерфейс входящего трафика.

threads <NUM>

Эта команда задает число потоков получения трафика.

По умолчанию: 1.

36.3.2.1

Настройка интерфейса в режиме AFP

buffer-size <N>

Эта команда задает размер буфера приема трафика (Кб). Большее значение может увеличить производительность.

emergency-flush

Эта команда задает восстановление из фазы отбрасывания пакетов. Данная опция может помочь при высокой нагрузке.

cksum <yes|no|auto>

Эта команда предписывает включить (yes) или выключить(no) проверку контрольной суммы пакетов. В режиме auto служба сама определяет, что делать.

cluster-id <ID>

Эта команда задает идентификатор кластера. Данный параметр нужен для балансировки нагрузки на основе потока.

cluster-type <flow|cpu>

Эта команда задает режим балансировки нагрузки для кластера:

- flow - все пакеты данного потока направляются в один и тот же сокет;
- cpu - все пакеты данного процессора направляются в один и тот же сокет.

По умолчанию: режим балансировки Round-Robin.

copy-to <IFACE> <ips|ids>

Эта команда задает интерфейс, куда следует копировать трафик, полученный с интерфейса, в конфигурации которого находится данная команда.

Параметр ips говорит о том, что в случае срабатывания на пакет drop-правила, пакет далее не копируется на указанный интерфейс.

Параметр ids говорит о том, что в случае срабатывания на пакет drop-правила, пакет все равно копируется на указанный интерфейс. Т.е. блокирования подозрительных пакетов не происходит.

defrag <yes|no>

Эта команда определяет, включить или выключить дефрагментацию пакетов в ядре. Если включено (yes): ядро выполняет дефрагментацию перед посылкой пакета. Если выключено (no): ядро не выполняет дефрагментацию перед посылкой пакета.

no promisc

Эта команда предписывает выключить неразборчивый режим.

ring-size <VAL>

Эта команда устанавливает размер кольцевого буфера.

По умолчанию: размер кольцевого буфера равен (pp * 2 / nt), где

- pp - это значение, определяемое командой limit pending-packets;
- nt - это значение определяемое командой threads.

36.4 Общие настройки.

policy <POL> {<NET>,8}

Эта команда задает привязку типа обработки фрагментов и слежения за сессиями к сетевым адресам. Т.е. определяет политику подсистемы дефрагментации и подсистемы сессий.

Возможные политики POL: linux, macos, vista, windows2k3 и др.

runmode <auto|single|workers|autofp <round-robin|active-packets|hash> >

Эта команда устанавливает метод, по которому потоки выполнения, модули и очереди взаимодействуют вместе.

Возможные методы:

- auto - пакеты одного и того же сетевого потока могут быть обработаны любым потоком выполнения подсистемы обнаружения;
- single - однопоточный режим;
- workers - каждый поток выполнения делает всю работу, начиная от выборки пакетов и заканчивая ведением лога;
- autofp - пакеты из каждого сетевого потока обрабатываются специально потоком выполнения подсистемы обнаружения, при этом балансировка нагрузки может быть выполнена по алгоритмам: - round-robin - сетевые потоки назначаются для обработки потокам выполнения циклически; - active-packets - сетевые потоки назначаются для обработки тем потокам выполнения,которые имеют самое низкое число необработанных пакетов,т.е. наименее занятым потокам; - hash - сетевые потоки назначаются для обработки потокам выполнения на основе хэш IP-адресов, т.е. это по сути случайная баоансировка.

limit asn1-frames

Эта команда задает максимальное число отслеживаемых ASN1-фрэймов.

По умолчанию: 256.

limit packet-size

Эта команда задает заранее выделяемый размер памяти для пакета.

По умолчанию: 1514.

limit pending-packets

Эта команда задает максимальное число пакетов, ожидающих в очереди на обработку.

По умолчанию: 1024.

limit pcre

Эта команда задает параметры обработки с использованием регулярных выражений.

Параметры:

- match : максимальное число обрабатываемых совпадений при поиске по регулярным выражениям;
- match-recursion : задает максимальное число обрабатываемых рекурсивных совпадений при поиске по регулярным выражениям.

По умолчанию: 1500 и 3500.

36.5 Подсистема дефрагментации IP-пакетов.

Позволяет настроить опции подсистемы дефрагментации IP-пакетов.

Для входа в режим настройки опций следует ввести команду:

defragmentation

Далее следуют команды настройки подсистемы дефрагментации.

frags <NUM>

Эта команда задает максимальное число хранимых IP-фрагментов.

По умолчанию: 65535

hash-size <NUM>

Эта команда задает максимальное число записей хэш-таблицы подсистемы.

По умолчанию: 4096

memcap <VAL>

Эта команда задает максимальный объем памяти, выделяемый для подсистемы.

По умолчанию: 16мб.

timeout <NUM>

Эта команда задает таймаут (сек.),после которого будет прекращено слежение за фрагментом.

По умолчанию: 60 сек.

trakcers <NUM>

Эта команда задает максимальное число отслеживаемых дефрагментируемых потоков.

По умолчанию: 1000.

36.6 Подсистема сессий и соединений.

Позволяет настроить опции подсистемы сессий и соединений.

Для входа в режим настройки опций данной подсистемы следует ввести команду:

stream

Далее следуют команды настройки подсистемы.

tracking async

Эта команда предписывает следить за ассинхронными TCP-сессиями.

tracking cksum

Эта команда предписывает не обрабатывать пакеты с неверными контрольными суммами.

tracking memcap <VAL>

Эта команда задает максимальный размер памяти для подсистемы (байт).

По умолчанию: 32Мб.

tracking prealloc <NUM>

Эта команда задает число сессий, для которых заранее выделяется память.

По умолчанию: 32768.

tracking sessions <NUM>

Эта команда задает максимальное число одновременных сессий, которые отслеживаются.

По умолчанию: 262144.

36.7 Подсистема обнаружения.

Позволяет настроить опции подсистемы обнаружения IP-пакетов.

Для входа в режим настройки опций данной подсистемы следует ввести команду:

detection

Далее следуют команды настройки подсистемы.

max-recursion <NUM>

Эта команда задает предел рекурсии обнаружения, т.е. глубину обнаружения.

По умолчанию: 3000.

performance <low|medium|high>

Эта команда задает производительность подсистемы обнаружения.

Параметры:

- low : низкая производительность (малое число групп правил);
- medium : средняя производительность (среднее число групп правил);
- high : высокая производительность (большое число групп правил).

По умолчанию: medium

sgh-mpm-context <own|shared|auto>

Эта команда задает тип выделения MPM-контекстов для групп правил.

Параметры:

- own : каждая группа правил имеет свой MPM-контекст;
- shared : все группы правил разделяют один общий MPM-контекст
- auto : служба сама определяет тип выделения контекстов, исходя из MPM-алгоритма.

MPM-контекст - это контекст, описывающий MPM - подсистему поиска по образцам, используемую при анализе трафика на предмет попадания в правила обнаружения.

По умолчанию: auto

36.8 Подсистема потока.

Позволяет настроить опции подсистемы потока IP-пакетов.

Для входа в режим настройки опций следует ввести команду:

flow

Далее следуют команды настройки подсистемы.

emergency <prunes|recovery> <VAL>

Эта команда позволяет настроить поведение в аварийном режиме - когда максимальный размер памяти для подсистемы почти исчерпан.

Параметры:

- prunes - количество удаляемых потоков для каждого нового потока;
- recovery - процент потоков, которые нужно удалить, чтобы выйти из аварийного режима;
- VAL - количество или процент потоков.

По умолчанию: prunes:5, recovery:30.

memcap <VAL>

Эта команда задает максимальный объем памяти, выделяемый для подсистемы.

По умолчанию: 32мб.

hash-size <NUM>

Эта команда задает максимальное число записей хэш-таблицы подсистемы.

По умолчанию: 65536

prealloc <NUM>

Эта команда задает число потоков, для которых заранее выделяется память.

По умолчанию: 10000

timeout <default|tcp|udp|icmp> <standart|emergency> <new|established|closed> <VAL>

Эта команда задает таймауты переходов состояний потоков.

Параметры:

- tcp|udp|icmp - протокол;
- default - все остальные протоколы;
- standart - обычный режим работы;
- emergency - аварийный режим работы (почти вся память подсистемы занята);
- new - максимальный период установления соединения;
- established - максимальный период молчания после установления соединения;
- closed - максимальный период ожидания после закрытия соединения перед удалением потока.

36.9 Подсистема сетевых узлов.

Позволяет настроить опции подсистемы сетевых узлов.

Для входа в режим настройки опций данной подсистемы следует ввести команду:

host

Далее следуют команды настройки подсистемы.

memcap <VAL>

Эта команда задает максимальный объем памяти, выделяемый для подсистемы.

По умолчанию: 16мб.

hash-size <NUM>

Эта команда задает максимальное число записей хэш-таблицы подсистемы.

По умолчанию: 4096

prealloc <NUM>

Эта команда задает число сетевых узлов, для которых заранее выделяется память.

По умолчанию: 1000

36.10 Подсистема HTTP-парсера.

Позволяет настроить опции подсистемы HTTP-парсера.

Для входа в режим настройки опций следует ввести команду:

http-parser

Далее следуют команды настройки подсистемы. Они определяют поведение HTTP-парсера для случая, если пакет не попал не в одну из политик.

double-decode-path

Эта команда предписывает осуществлять двойное декодирование пути в URI.

double-decode-query

Эта команда предписывает осуществлять двойное декодирование запроса в URI.

max-request-size <VAL>

Эта команда задает максимальный размер HTTP-запроса для анализа службой.

По умолчанию: 3072 байта.

max-response-size <VAL>

Эта команда задает максимальный размер HTTP-ответа для анализа службой.

По умолчанию: 3072 байта.

default-policy <POL>

Эта команда устанавливает политику HTTP по умолчанию.

Возможные политики POL:

- iis40/50/51/60/70/75 - сервер IIS;
- apache/apache22 - сервер Apache;
- tomcat60 - сервер Tomcat;
- ids;

- generic;
- minimal.

policy <POL>

Эта команда осуществляет вход в настройки подсистемы для указанной политики HTTP.

Возможные политики см. выше в описании команды default-policy в текущем разделе.

36.10.1 Настройка политики HTTP-парсера.

address {<NET>,8}

Эта команда задает привязку политики к сетевым адресам. Трафик с указанных адресов будет обрабатываться данной политикой.

Остальные команды - такие же, как в настройке HTTP-парсера, однако они определяют его поведение конкретно для данной политики и для трафика с указанных командой address адресов.

36.11 Подсистема MPM.

Позволяет настроить опции подсистемы MPM (Multi Pattern Matcher).

MPM - подсистема поиска данных по множеству образцов.

Для входа в режим настройки опций следует ввести команду:

mpm

Далее следуют команды настройки опций подсистемы.

alg <ALG>

Эта команда позволяет задать алгоритм,используемый в MPM. Возможные алгоритмы: b2g, b2gc, b2gm, b3g, wumanber, ac, ac-gfbs.

algopt <ALG> search-algo

Эта команда позволяет задать алгоритм поиска SALG для алгоритма ALG.

Возможные алгоритмы ALG: b2g, b2gc,b2gm,b3g,wumanber.

Возможные алгоритмы поиска: B3gScan, B3gScanBNDMq, B3gSearchBNDMq, B3gSearch.

algopt <ALG> hash-size

Эта команда задает размер MPM хэш-таблицы для для алгоритма ALG. Значение lowest - минимальный размер. Значение max - максимальный размер.

algopt <ALG> bf-size

Эта команда задает размер MPM bloom-фильтра, используемого при проверке образца. Значение low - минимальный размер. Значение high - максимальный размер.

36.12 Подсистема потоков выполнения.

Позволяет настроить опции подсистемы потоков выполнения.

Для входа в режим настройки опций следует ввести команду:

threading

Далее следуют команды настройки подсистемы.

thread-set <TS>

Эта команда осуществляет вход в настройки многопоточности для потоков заданного типа TS. Типы потоков могут быть следующие:

- management - поток управления;
- receive - поток получения данных;
- decode - поток декодирования;
- stream - поток обработки сетевых потоков,сессий,соединений;
- detect - поток обнаружения;
- verdict - поток принятия решения (вердикт) о том, что делать с сетевым пакетом;
- reject - поток блокирования пакетов с посылкой ICMP-сообщений;
- outputs - поток вывода информации о предупреждениях в журнал.

36.12.1 Настройка CPU-affinity для потоков выполнения.

В данном разделе указывается привязка процессорных ядер и их приоритетов к потокам выполнения выбранного командой thread-set типа (далее: потоки).

cpu {<all|NUM|NUM1-NUM2>,8}

Эта команда устанавливает номера процессорных ядер, на которых могут выполняться потоки.

mode <balanced|exclusive>

Эта команда устанавливает режим исполнения потоков: номера ядер, выполняющих потоки.

Параметры:

- balanced - отдельные потоки могут обрабатываться любыми ядрами из указанных командой cpu;

- exclusive - каждому потоку назначается свой набор ядер из множества, определенного командой cpu.

prio <default <PRIO>> [low {<NUM|NUM1-NUM2>,8}] [medium {<NUM|NUM1-NUM2>,8}] [high {<NUM|NUM1-NUM2>,8}]

Эта команда устанавливает приоритеты потоков выполнения.

Параметры:

- PRIO - задает приоритет потоков по умолчанию:
  - low - низкий приоритет;
  - medium - средний приоритет;
  - high - высокий приоритет;
- NUM|NUM1-NUM2 - задает номера процессорных ядер, которые используются для обслуживания потоков с приоритетом: высокий(high), низкий(low) или средний(medium).

36.13 Работа с правилами

Для входа в группу правил следует выполнить команду:

ruleset <NAME>

(Следует использовать TAB для просмотра имеющихся в системе групп правил).

После входа в группу правил становятся доступными следующие команды.

on <all| <GID> <SID> >

Эта команда позволяет включить все или указанное правило данной группы. Правила идентифицируется по номерам GID/SID, которые можно узнать по команде show service ids rules.

off <all|<GID> <SID>

Эта команда позволяет выключить все или указанное правило данной группы. Правила идентифицируется по номерам GID/SID, которые можно узнать по команде show service ids rules

action <GID> <SID> <TYPE>

Эта команда изменяет действие встроенного правила под номером GID/SID на действие, определяемое параметром TYPE:

- alert : записать в журнал пакет и предупреждение;
- pass : разрешить прохождение пакета;
- drop : записать пакет в журнал и запретить пакет;
- reject : записать пакет в журнал и запретить пакет; одновременно послать “TCP reset” (для TCP) или “ICMP port unreachable” (для UDP).

36.14 Настройка журналов

Настройка журналов и предупреждений службы IDSSUR осуществляется командой log и ее подкомандами.

log service <emergency|critical|alert|error|warning|notice|info>

Эта команда включает журнал службы. Пишутся сообщения с указанным приоритетом и выше.

log text-alerts

Эта команда предписывает регистрировать предупреждения службы в текстовом виде.

log binary-alerts local [SZ]

Эта команда предписывает регистрировать предупреждения службы в бинарном виде и хранить их на локальной системе.

Максимальный размер файла предупреждений - SZ (Мб).

По умолчанию: 10Мб.

log binary-alerts postgresql <IFACE> <IP> [port <PORT>] [SZ] [user <USR>] [password <PSW>] [brief]

Эта команда предписывает регистрировать предупреждения службы в бинарном виде и пересылать их на удаленную базу данных Postgres.

Параметры:

- IFACE - интерфейс, с которого следует пересылать предупреждения на удаленную систему;
- IP - адрес сервера базы данных;
- PORT - порт сервера базы данных;
- SZ - максимальный размер бинарного файла предупреждений;
- USR - имя пользователя базы данных;
- PSW - пароль пользователя базы данных;
- brief - краткая информация о предупреждении.

По умолчанию: PORT:5432, SZ:10Мб.

37. MAILER - служба пересылки почтовых сообщений

37.1 Введение

Система имеет возможность отправки сообщений на указанный e-mail с помощью службы пересылки почтовых сообщений - MAILER

37.2 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
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
```

37.2.1 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
```

37.3 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)
<file|running-config|startup-config|default-config>	Отправка содержимого файла, running-config, startup-config или default-config. (Необязательный параметр)

37.4 Настройка service-watcher для отправки сообщений с помощью службы MAILER

Система имеет возможность отслеживания появления заданных сообщений в журнальных файлах системы и дальнейшую отправку сообщений на указанный e-mail.

Для этого в настройке службы service-watcher необходимо указать параметры для службы пересылки почтовых сообщений.

Ниже приведен пример для поиска и отправки новых сообщений, содержащих строку usb в kernel.log на адрес test@mail.com

```
adm@DionisNX(config—service—watcher—kernel)# mailer usb test@mail.com
```

Все параметры команды mailer для настройки service-watcher:

команда	параметр
<rx>	Регулярное выражения для поиска. (Обязательный параметр)
via <account>	Имя настроенной учетной записи службы mailer. (Необязательный параметр. Если не указан, произойдет отправка с учетной записи, установленной по умолчанию)
<addr1,addr2…>	Список e-mail-адресов для рассылки (Обязательный параметр)
subject <Text>	Тема сообщения (Необязательный параметр)

38. L2TP-туннели

38.1 Введение

Система 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.

Настройка L2TP в системе сводится к настройке динамических интерфейсов типа l2tps и l2tp.

Основные сущности протокола:

- LAC (концентратор доступа L2TP) - настраивается через интерфейс с типом l2tp;
- LNS (сетевой сервер L2TP) - настраивается через интерфейс с типом l2tps.
- Виртуальный IP-адрес - это IP-адрес конца туннеля, назначаемый при его создании.

Сообщения протокола:

- управляющие: создание, поддержка и удаление туннеля; состоят из множества AVP(пара «атрибут-значение»)-заголовков; в обрабатываются в режиме пользователя;
- информационные сообщения: передача данных по туннелю; в обрабатываются в режиме пользователя или в режиме ядра (по умолчанию), зависит от настройки.

При настроенном IPSec в возможна защищенная передача пакетов L2TP через туннели IPSec.

Кроме этого,возможно применения правил NAT и ACL для интерфейсов l2tp и l2tps.

38.2 Настройка LNS

Для настройки LNS необходимо войти в режим настройки l2tps-интерфейса:

```
(config)# interface l2tps 0
```

Данный интерфейс представляет собой серверный пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов (LAC).

Далее в данном разделе интерфейс l2tps будем называть серверным пулом (интерфейсов).

Формат реально создаваемого динамического интерфейса (туннеля) следующий:

```
ls<N>—<LNS>—<IFNUM>
```

Например: ls0-tomsk-123

Рассмотрим данный формат:

- N - это номер серверного пула (от 0 до 99);
- LNS - это имя LNS (длиной от 1 до 5 символов), которое используется в командах PPP-аутентификации как имя узла;
- IFNUM - динамический номер туннеля в данной LNS (от 0 до 9999), который определяется автоматически при создании туннеля.

Каждый серверный пул - это отдельный процесс. В серверном пуле может быть несколько LNS, каждая со своими L2TP- и PPP-параметрами.

38.2.1 Глобальные настройки серверного пула

После входа в режим настройки l2tps-интерфейса строка приглашения будет иметь следующий вид:

```
(config—if—l2tps0)#
```

38.2.1.1 enable

Эта команда включает серверный пул интерфейсов.

38.2.1.2 disable

Эта команда выключает серверный пул интерфейсов:

38.2.1.3 log <all | avp | network | tunnel | packet | state>

Эта команда включает журналы заданного типа для текущего серверного пула интерфейсов.

Типы журналов:

- network - сетевое взаимодействие;
- tunnel - туннели;
- packet - сетевые пакеты;
- state - состояние;
- avp - AVP-значения протокола;
- all - все пречисленные выше журналы.

По умолчанию: отключены

38.2.1.4 listen <IP> [PORT]

Эта команда задает сокет IP:PORT для принятия L2TP-запросов.

По умолчанию: 0.0.0.0:1701

38.2.1.5 userspace

Эта команда определяет, что обработка информационных сообщений L2TP будет производится в режиме пользователя.

По умолчанию: в режиме ядра.

38.2.1.6 [N] lns <NAME>

Эта команда создает LNS с именем NAME под номером N.

Пул интерфейсов без настройки LNS не имеет смысла.

Именно в LNS указывается основная часть информации по созданию динамических L2TP-туннелей.

Длина имени LNS - от 1 до 5 символов.

Имя NAME соответствует имени узла L2TP и может использоваться в командах, требующих задания имени узла (например pap/chap команды).

Номер N задает приоритет LNS: при попытке подсоединения клиента к LNS список LNS просматривается по порядку возрастания приоритета. Для создания туннеля с клиентом будет использована первая найденная LNS. Поиск LNS осуществляется по диапазону разрешенных IP-адресов LAC (команда ‘lac range’). Если соединение с найденной LNS не будет успешным, попытка соединения с другими LNS, которые подходят к данной LAC, не будет осуществлена.

38.2.2 Общие настройки для LAC и LNS

Далее будут рассмотрены общие настройки для LAC и LNS.

38.2.2.1 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.

38.2.2.2 hidden-avp

Эта команда включает скрытие поля данных в AVP, содержащих важную информацию управляющих сообщений, такую как пароль пользователя или его ID.

По умолчанию: выключено.

38.2.2.3 length-bit

Эта команда включает использование поля длины сообщения в заголовке пакета L2TP.

По умолчанию: выключено.

38.2.2.4 txspeed <VAL>

Эта команда устанавливает скорость отправления данных через туннель в значение VAL бит/сек.

По умолчанию: 10Мбит/сек.

38.2.2.5 rxspeed <VAL>

Эта команда устанавливает скорость приема данных через туннель в значение VAL бит/сек.

По умолчанию: 10Мбит/сек.

38.2.2.6 tun-rws <VAL>

Эта команда устанавливает максимальное число входящих неподтвержденных пакетов контрольного канала.

По умолчанию: 4.

38.2.3 Настройка LNS

Далее перечислены команды,специфичные для LNS.

38.2.3.1 localip <IP>

Команда устанавливает виртуальный IP-адрес концу туннеля на стороне LNS.

38.2.3.2 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.

Если не указано ни одного интервала и нет ни одной LNS с интервалами lac-range, то попытка соединения разрешена всем LAC.

38.2.3.3 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.

38.2.3.4 permit ip-range <IP_START> [IP_END]

Эта команда задает интервал виртуальных IP-адресов, которые разрешено назначать удаленному концу туннеля (L2TP клиенту).

38.2.3.5 deny ip-range <IP_START> [IP_END]

Эта команда задает интервал виртуальных IP-адресов, которые не разрешено назначать удаленному концу туннеля (L2TP клиенту).

38.2.4 Настройка PPP

Настройки опций протокола PPP задаются для LAC- и LNS-протокола L2TP,а также для протокола PPTP (см. «Протокол PPTP»).

Все команды для задания PPP-опций, кроме команд для задания PPP-аутенификации, начинаются со слова ppp.

38.2.4.1 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.

38.2.4.2 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.

38.2.4.3 ppp idle <N>

Эта команда задает максимальный период (в сек.) бездействия туннеля. По истечении данного периода времени туннель будет закрыт.

Доступно только для интерфейсов l2tps и pptps.

По умолчанию: 1800.

38.2.4.4 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.

38.2.4.5 ppp lcp-echo-interval <N>

Эта команда задает интервал времени в секундах между LCP Echo-запросами.

По умолчанию: 3.

38.2.4.6 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.

38.2.4.7 ppp mru <VAL>

Эта команда устанавливает размер MRU (Maximum receive unit) в байтах.

По умолчанию: 1500.

38.2.4.8 ppp mtu <VAL>

Эта команда устанавливает размер MTU (Maximum transmit unit) в байтах.

По умолчанию: 1500.

38.2.4.9 ppp ms-dns <IP>

Эта команда предписывает передавать указанный адрес DNS-сервера не-NX-клиентам (например Android- или Windows-клиент).

Доступно только для интерфейсов l2tps и pptps.

38.2.4.10 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.

38.2.4.11 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.

38.2.4.12 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.

38.2.4.13 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.

38.2.4.14 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.

38.2.4.15 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.

38.2.4.16 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.

38.2.5 L2TP аутентификация

Для настройки взаимной аутентификации LNS и LAC необходимо задать в их настройках опцию challange-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 l2tp или l2tps итерфейса необходимо задать общий секрет следующей командой:

38.2.5.1 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 , то:

- для LNS: LOCAL - это имя LNS; REMOTE - это имя LAC
- для LAC: LOCAL - это имя LAC; REMOTE - это имя LNS

Если LAC и/или LNS создаются не в системе , то передаваемые имена узлов LOCAL и REMOTE могут быть, например, доменными именами соответствующих узлов.

38.2.6 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:

38.2.6.1 <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, 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 «*».

Особенности задания HOST_PASSIVE и HOST_ACTIVE:

- для интерфейсов l2tp и l2pts: имена узлов соответсвуют именам LAC и LNS соответственно;
- для интерфейсов pptp и pptps: имена узлов задаются командой ppp localname.

Знак «*» - означает любой узел.

38.3 Настройка LAC

Для настройки LAC необходимо войти в режим настройки l2tp-интерфейса:

```
(config)# interface l2tp 0
```

Данный интерфейс представляет собой клиентский пул динамических интерфейсов, создаваемых в системе при создании туннеля с LNS.

Далее в данном разделе интерфейс l2tp будем называть клиентским пулом (интерфейсов).

Один динамических интерфейс будет создаваться на каждый LAC, описанный в данном пуле, если этот LAC успешно соединился с LNS.

Формат реально создаваемого динамического интерфейса (туннеля) следующий:

```
lc<N>—<LAC>
```

Например: lc0-to_moscow

Рассмотрим данный формат:

- N - это номер клиентского пула интерфейсов (от 0 до 99);
- LAC - это имя LAC (длиной от 1 до 10 символов); используется в командах PPP-аутентификации как имя узла.

Каждый интерфейс - это отдельный процесс. В интерфейсе может быть несколько LAC, каждая со своими L2TP- и PPP-параметрами.

38.3.1 Глобальные настройки клиентского пула

После входа в режим настройки l2tp-интерфейса строка приглашения будет иметь следующий вид:

```
(config—if—l2tp0)#
```

Глобальные настройки интерфейса l2tp аналогичны п. «Глобальные настройки серверного пула» раздела «Настройка LNS» главы «L2TP-туннели».

Перечислим только специфичные для данного интерфейса команды.

38.3.1.1 lac <NAME>

Эта команда создает LAC с именем NAME (длиной от 1 до 10 символов).

Можно создать несколько LAC в данном интерфейсе.

38.3.2 Общие настройки для LAC и LNS

См. п. «Общие настройки для LAC и LNS» раздела «Настройка LNS» главы «L2TP протокол».

38.3.3 Настройка LAC

38.3.3.1 srv <DOMAIN | IP[:PORT]>

Эта команда указывает имя или IP-адрес и порт LNS, с которым нужно создать туннель.

38.3.3.2 redial-interval <TO>

Эта команда задает интервал времени в секундах между попытками повторного соединения.

По умолчанию: 30 сек.

38.3.4 Настройка PPP

См. п. «Настройка PPP» раздела «Настройка LNS» главы «L2TP протокол».

38.3.5 L2TP аутентификация

См. п. «L2TP аутентификация» раздела «Настройка LNS» главы «L2TP протокол».

38.3.6 PPP аутентификация

См. п. «PPP аутентификация» раздела «Настройка LNS» главы «L2TP протокол».

38.4 Прочая работа

38.4.1 Просмотр журналов

Для просмотра журналов следует ввести команду:

38.4.1.1 show interface log <l2tp|l2tps> <IFNUM> <ARGS…>

Просмотр журналов интерфейса типа l2tp или l2tps с номером интерфейса IFNUM.

Остальные параметры команды (ARGS) аналогичны другим командам просмотра журналов.

38.4.2 Особенности привязки NAT и ACL к туннелю

Предположим, что создан список NAT с именем n1.

Этот созданный список NAT можно привязать к интерфейсу l2tp или l2tps.

Правила привязки ACL аналогичны нижерассмотренным правилам для NAT.

38.4.2.1 Привязка списка NAT к интерфейсу l2tps

Привязка NAT ко всем динамическим интерфейсам из серверного пула l2tps0: список NAT будет привязан ко всем интерфейсам с именем, начинающимся на ls0-.

```
(config)# interface l2tps 0
(config—if—l2tps0)# ip nat—group n1
```

Привязка NAT ко всем динамическим интерфейсам из LNS lns1 серверного пула l2tps0: список NAT будет привязан ко всем интерфейсам с именем, начинающимся на ls0-lns1-.

```
(config)# interface l2tps 0
(config—if—l2tps0)# lns lns1
(config—if—l2tps0—lns1)# ip nat—group n1
```

Привязка NAT к конкретному динамическому интерфейсу из LNS lns1 серверного пула l2tps0: список NAT будет привязан к интерфейсу с именем ls0-lns1-1.

```
(config)# interface l2tps 0
(config—if—l2tps0)# lns lns1
(config—if—l2tps0—lns1)# ip nat—group n1 1
```

38.4.2.2 Привязка списка NAT к интерфейсу l2tp

Привязка NAT ко всем динамическим интерфейсам из клиентского пула l2tp0: список NAT будет привязан ко всем интерфейсам с именем, начинающимся на lc0-.

```
(config)# interface l2tp 0
(config—if—l2tp0)# ip nat—group n1
```

Привязка NAT к динамическому интерфейсу, соответствующему LAC lac1 клиентского пула l2tp0: список NAT будет привязан к интерфейсу с именем lc0-lac1.

```
(config)# interface l2tp 0
(config—if—l2tp0)# lac lac1
(config—if—l2tp0—lac1)# ip nat—group n1
```

38.5 Пример настройки

38.5.1 L2TP-туннель

Рассмотрим пример настройки серверного и клиентского L2TP-интерфейсов.

Имеется два изделия :

- одно из них - L2TP-сервер сети (LNS) с адресом 10.0.0.1/24 из сети 10.0.0.0/24;
- 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.

Серверный пул интерфейсов L2TP:

```
(config)# interface l2tps 0
(config—if—l2tps0)# lns nx1
(config—if—l2tps0—nx1)# permit ip—range 192.168.1.2 192.168.1.100
(config—if—l2tps0—nx1)# localip 192.168.1.1
(config—if—l2tps0—nx1)# permit lac—range 10.0.0.2 10.0.0.100
(config—if—l2tps0—nx1)# ppp chap requrie
(config—if—l2tps0—nx1)# chap nx2 nx1 123
(config—if—l2tps0)# enable
```

Реальные ls0-nx1-* интерфейсы будут создаваться по мере успешного подключения L2TP-клиентов.

Клиентский интерфейс L2TP:

```
(config)# interface l2tp 0
(config—if—l2tp0)# lac nx2
(config—if—l2tp0—nx2)# srv 10.0.0.1
(config—if—l2tp0—nx2)# chap nx2 nx1 123
(config—if—l2tp0—nx2)# enable
```

Реальный lc0-nx2 интерфейс будет создан. Кроме этого на nx1 появится интерфейс ls0-nx1-0.

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 и IPSEC на Windows, используя, например, preshared-ключ;
- отключить использование IPSEC в ОС Windows: создать ключ типа dword-32bit с именем ProhibitIpSec и значением 1.

38.5.2 PPTP туннель

Рассмотрим пример настройки сереврного и клиентского PPTP-интерфейсов.

Имеется два изделия :

- одно из них - PPTP-сервер с адресом 10.0.0.1/24 из сети 10.0.0.0/24;
- другое - PPTP-клиент с адресом 10.0.0.2/24 из сети 10.0.0.0/24.

Цель: создать PPTP VPN 192.168.2.0/24 между сервером и клиентом в виде PPTP-туннеля.

Серверный пул интерфейсов PPTP:

```
(config)# interface pptps 0
(config—if—pptps0)# localip 192.168.2.1 10
(config—if—pptps0)# remoteip 192.168.2.20 10
(config—if—pptps0)# ppp chap require
(config—if—pptps0)# ppp localname nx1
(config—if—pptps0)# chap nx2 nx1 123
(config—if—pptps0)# enable
```

Реальные ps0-* интерфейсы будут создаваться по мере успешного подключения PPTP-клиентов.

Клиентский интерфейс PPTP:

```
(config)# interface pptp 0
(config—if—pptp0)# srv 10.0.0.1
(config—if—pptp0)# ppp localname nx2
(config—if—pptp0)# chap nx2 nx1 123
(config—if—pptp0)# enable
```

Реальный pc0-интерфейс будет создан. Кроме этого на nx1 появится интерфейс ps0-0.

39. PPTP-туннели

39.1 Введение

Система 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 L2TP в системе сводится к настройке динамических интерфейсов типа pptps (серверный пул интерфейсов) и pptp (клиентский интерфейс).

Для интерфейсов pptp и pptps возможно применения правил NAT и ACL.

39.2 Настройка серверного пула интерфейсов

Данный тип интерфейса (pptps) представляет собой пул динамических интерфейсов, создаваемых в системе по мере подключения клиентов PPTP, по аналогии с пулом l2tps. Отличие состоит в том, что в случае с pptps нет понятия LNS, т.е. уходит один уровень иерархии.

Далее в данном разделе интерфейс pptps будем называть серверным пулом интерфейсов.

Формат реально создаваемого динамического интерфейса (туннеля) следующий:

```
ps<N>—<IFNUM>
```

Например: ps0-123

Рассмотрим данный формат:

- N - это номер серверного пула интерфейсов (от 0 до 99);
- IFNUM - динамический номер туннеля (от 0 до 9999); определяется автоматически при создании туннеля.

Каждый пул - это отдельный процесс.

39.2.1 Настройка серверного пула интерфейсов

После входа в режим настройки pptps строка приглашения будет выглядеть следующим образом:

```
(config—if—pptps0)#
```

39.2.1.1 enable

Эта команда включает пул интерфейсов.

39.2.1.2 disable

Эта команда выключает пул интерфейсов.

39.2.1.3 log

Эта команда включает ведение журналов.

По умолчанию: отключены

39.2.1.4 listen <IP>

Эта команда задает сокет IP:1723 для принятия PPTP запросов.

По умолчанию: 0.0.0.0:1723.

39.2.1.5 localip <IP_START> <N>

Эта команда задает один или несколько IP-адресов, которые будут использоваться в качестве локальных адресов туннеля.

39.2.1.6 remoteip <IP_START> <N>

Эта команда задает один или несколько IP-адресов, которые будут использоваться в качестве удаленных адресов туннеля.

39.2.2 Настройка PPP

См. п. «Настройка PPP» раздела «Настройка LNS» главы «L2TP-протокол».

39.2.3 PPP аутентификация

См. п. «PPP аутентификация» раздела «Настройка LNS» главы «L2TP-протокол».

39.3 Настройка клиентского интерфейса

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 при построении туннеля с сервером PPTP.

Формат реально создаваемого динамического интерфейса (туннеля) следующий:

```
pc<N>
```

Например: pc0.

Рассмотрим данный формат:

- N - это номер интерфейса (от 0 до 99).

Каждый интерфейс - это отдельный процесс.

39.3.1 Настройка интерфейса

После входа в режим настройки динамического интерфейса строка приглашения будет иметь следующий вид:

```
(config—if—pptp0)#
```

39.3.1.1 enable

Эта команда включает интерфейс.

39.3.1.2 disable

Эта команда выключает интерфейс.

39.3.1.3 log

Эта команда включает ведение журналов.

По умолчанию: отключены.

39.3.1.4 redial-interval <TO>

Эта команда задает интервал времени в секундах между попытками повторного соединения.

По умолчанию: 15 сек.

39.3.1.5 srv <DOMAIN | IP>

Эта команда задает имя или IP-адрес PPTP-сервера, с которым нужно создать туннель.

39.3.2 Настройка PPP

См. п. «Настройка PPP» раздела «Настройка LNS» главы «L2TP-протокол».

39.3.3 PPP аутентификация

См. п. «PPP аутентификация» раздела «Настройка LNS» главы «L2TP-протокол».

39.3.4 Особенности привязки NAT и ACL к туннелю

Предположим, что создан список NAT с именем n1.

Этот созданный список NAT можно привязать к интерфейсу pptp или pptps.

Правила привязки ACL будут аналогичны нижерассмотренным правилам для NAT.

39.3.4.1 Привязка списка NAT к интерфейсу pptps

Привязка NAT ко всем динамическим интерфейсам из серверного пула pptps0 осуществляется при помощи следующих команд (список NAT будет привязан ко всем интерфейсам с именем, начинающимся на ps0-):

```
(config)# interface pptps 0
(config—if—pptps0)# ip nat—group n1
```

Привязка NAT к конкретному динамическому интерфейсу из серверного пула pptps0 осуществляется при помощи следующих команд (список NAT будет привязан к интерфейсу с именем ps0-1):

```
(config)# interface pptps 0
(config—if—pptps0)# ip nat—group n1 1
```

39.3.4.2 Привязка списка NAT к интерфейсу pptp

Привязка NAT к динамическому интерфейсу l2tp0 осуществляется при помощи следующих команд (список NAT будет привязан к интерфейсу с именем pc0):

```
(config)# interface l2tp 0
(config—if—pptp0)# ip nat—group n1
```

39.4 Пример настройки

Рассмотрим пример настройки серверного и клиентского PPTP-интерфейсов.

Имеются два изделия :

- одно из них - сервер PPTP с адресом 10.0.0.1/24 из сети 10.0.0.0/24;
- другое - клиент PPTP с адресом 10.0.0.2/24 из сети 10.0.0.0/24.

Цель: создать PPTP VPN 192.168.1.0/24 между сервером и клиентом в виде PPTP-туннеля.

Серверный пул интерфейсов PPTP создается при помощи команд:

```
(config)# interface pptps 0
(config—if—pptps0)# localip 192.168.1.1 1
(config—if—pptps0)# remoteip 192.168.1.2 100
(config—if—pptps0)# ppp chap requrie
(config—if—pptps0)# chap nx1 nx2 123
(config—if—pptps0)# ppp localname nx2
(config—if—pptps0)# enable
```

Реальные ps0-* интерфейсы будут создаваться по мере успешного подключения PPTP-клиентов.

Клиентский интерфейс PPTP создается при помощи команд:

```
(config)# interface pptp 0
(config—if—pptp0)# srv 10.0.0.2
(config—if—pptp0)# chap nx1 nx2 123
(config—if—pptp0)# ppp localname nx1
(config—if—pptp0)# enable
```

Реальный pc0-интерфейс будет создан,если команда enable успешно выполнится(будет занесена в конфигурацию).

40. Механизмы качества обслуживания (QoS)

Механизмы QoS Dinis-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;
- значения TOS/DSCP в пакетах.

40.1 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
```

Внимание!!! Порядок сопоставления трафика классам не определен. Это означает, что классы не должны пересекаться по своим выборкам!

40.2 Отображение CoS в класс для VLAN

В заголовке ethernet-фрейма 802.1Q присутствуют три бита (Class of Service (CoS)), которые могут использоваться для классификации и маркировки трафика. Для входящего трафика в VLAN-интерфейс можно задать отображение CoS в класс трафика (созданный с помощью class-map). Для этого используется команда ingress-qos-map;

```
DionisNX(config)# ip class—map low
DionisNX(config—cmap—low)# ip class—map high
DionisNX(config—cmap—high)# exit
DionisNX(config)# interface ethernet 0.2
DionisNX(config—if—ethernet0.2)# ingress—qos—map 1:low,5:high
```

В качестве параметра к команде ingress-qos-map задается список из максимум восьми отображений, разделенных запятыми, в виде: <значение CoS>:<имя класса class-map>

При этом, входящему трафику будет сопоставлен класс на основе значения поля CoS.

Для выходящего трафика из VLAN-интерфейса также можно задать отображение класса трафика в значение бит CoS, для этого используется команда egress-qos-map:

```
DionisNX(config)# interface ethernet 0.2
DionisNX(config—if—ethernet0.2)# engress—qos—map low:1,high:5
```

В качестве параметра к команде egress-qos-map задается список из максимум восьми отображений, разделенных запятыми, в виде: <имя класса class-map>:<значение CoS>

При этом, выходящему трафику будет сопоставлен приоритет CoS на основе класса трафика.

Для сброса отображений используются команды: no ingress-qos-map и no egress-qos-map.

40.3 Политика обслуживания

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 не обязательные параметры].

Порядок правил в политике не имеет значения, поэтому при работе со списком действуют следующие правила.

- команда class <имя класса> … добавляет правило для класса, если его еще не было в списке;
- команда class <имя класса> … изменяет существующее правило класса, если оно уже было в списке;
- команда no class <имя класса> удаляет правило для класса из списка.

Для просмотра редактируемого списка удобно воспользоваться командой: do show.

Следует отметить, что независимо от созданных классов трафика всегда существует класс default, к которому относятся те пакеты, к которым не были применены правила обслуживания политики. Другими словами, класс default по смыслу означает «весь остальной трафик».

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)
tos значение/маска	Установка битов TOS (для всех классов, кроме класса default)
dscp значение/символьное имя	Установка DSCP (для всех классов, кроме класса default)

Если в правиле не указан параметр ceil, то пропускная способность заданного класса не будет превышать rate, то-есть, отсутствие параметра ceil это синоним задания ceil равному rate.

Если в правиле указан параметр ceil, то в политике обслуживания необходимо задать общую пропускную способность, из которой и будет распределяться незанятая полоса. Общая пропускная способность задается с помощью команды bandwidth:

```
DionisNX(config)# ip policy—map outworld
DionisNX(config—pmap—outworld)# bandwidth 100mbit
```

Команда no bandwidth удаляет информацию о пропускной способности канала. (При этом неявно считается, что она составляет 10Гбит).

Также, существует возможность ограничения входящего трафика. Для этого используется команда ingress: ingress rate <скорость>.

```
DionisNX(config)# ip policy—map outworld
DionisNX(config—pmap—outworld)# ingress rate 10mbit
```

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 tos 0x00/0xe0
 class prt1 rate 1kbit ceil 10000mbit priority 6 tos 0x20/0xe0
 class prt2 rate 1kbit ceil 10000mbit priority 5 tos 0x40/0xe0
 class prt3 rate 1kbit ceil 10000mbit priority 4 tos 0x60/0xe0
 class prt4 rate 1kbit ceil 10000mbit priority 3 tos 0x80/0xe0
 class prt5 rate 1kbit ceil 10000mbit priority 2 tos 0xa0/0xe0
 class prt6 rate 1kbit ceil 10000mbit priority 1 tos 0xc0/0xe0
 class prt7 rate 1kbit ceil 10000mbit priority 0 tos 0xe0/0xe0
```

Для просмотра списка/списков политик обслуживания следует использовать команду show ip policy-map <имя|*> [config|zero], доступную из enable-режима. Например:

```
DionisNX(config)# do show ip policy—map * config
```

При этом, если задан параметр config – будет показана конфигурация из running-config. Если не задан параметр config – будет показана действующая конфигурация, загруженная в ядро, для заданного класса (параметр <имя>) или всех классов (параметр *). Следует отметить, что класс становится действующим только в том случае, если реально используется. Тем самым, недействующие классы, на самом деле, не участвуют в классификации, несмотря на то, что находятся в running-config.

40.4 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
```

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
```

40.5 Туннельный трафик

Классификация трафика с помощью ip class-map осуществляется до первого удачного сопоставления с классом. Обычно этот нюанс не имеет значения при администрировании, однако в случае туннельного трафика это поведение иногда необходимо учитывать.

Например, рассмотрим трафик GRE туннеля. Пусть имеется класс:

```
DionisNX(config—cmap—gre)# match gre
```

Тогда при приходе трафика протокола GRE он будет классифицирован. Затем, из туннеля будет извлечен внутренний пакет (предположим, UDP) и этот пакет уже не будет классифицироваться – с ним будет по прежнему связан уже ранее сопоставленный класс gre.

На самом деле, обычно нет никакого смысла классифицировать трафик до того, как он будет извлечен из туннеля, но иногда это все-таки необходимо.

Кроме того, можно привести более наглядный пример:

```
DionisNX(config—cmap—udp)# match udp
```

Предположим, что UDP-пакеты заворачиваются в gre-туннель на выходе из маршрутизатора. Тогда класс GRE-пакетов останется тем же, что при классификации его ранее, как трафика UDP. Это может быть нежелательным в некоторых случаях.

В рассматриваемых ситуациях могут оказаться полезными классы типа tunnel. Для того, чтобы отметить class-map как туннельный, в режиме редактирования класса нужно выполнить команду:

```
DionisNX(config—cmap—gre)# tunnel
```

Для отмены режима туннеля:

```
DionisNX(config—cmap—gre)# no tunnel
```

Туннельный класс работает следующим образом:

1. Для обычного трафика, не пришедшего из туннеля, сопоставление с классом работает обычным образом.
2. Для туннельного трафика, он сопоставляется с классом до извлечения из туннеля и (в случае успешного первого сопоставления) после извлечения из туннеля.
3. Для туннелируемого трафика (который будет завернут в туннель) выполняется сопоставление до заворачивания и (в случае успешного 1-го сопоставления) после упаковки в туннель.
4. В обоих случаях (2 и 3) при втором сопоставлении с классом трафик уже промаркирован этим классом, но он может быть исключен из класса с помощью exclude.
5. При втором сопоставлении с классом, если трафик оказывается исключенным из класса (на шаге 4), происходит классификация обычными классами (без режима tunnel).

Например, рассмотрим несколько вариантов применения классов:

```
DionisNX(config—cmap—gre)# tunnel
DionisNX(config—cmap—gre)# match gre
DionisNX(config—cmap—gre)# exclude icmp
```

Здесь в класс gre попадут:

- gre-датаграммы;
- весь трафик внутри gre-датаграмм, кроме icmp-датаграмм.

```
DionisNX(config—cmap—esp)# tunnel
DionisNX(config—cmap—esp)# exclude
DionisNX(config—cmap—esp)# match esp
```

Здесь в класс esp попадут:

- весь esp-трафик;
- все содержимое из esp-туннелей будет исключено из класса esp.

```
DionisNX(config—cmap—gre)# tunnel
DionisNX(config—cmap—gre)# exclude
DionisNX(config—cmap—gre)# match gre
DionisNX(config—cmap—gre)# match udp
```

Здесь в класс gre попадут:

- весь gre-трафик;
- udp-трафик из gre-туннелей;
- другой трафик, извлеченный из туннелей gre, будет исключен из класса gre;

41. 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 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, например:

```
DionisNX(config)# no ip route default
DionisNX(config)# ip policy—route gateway 192.168.33.1
```

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
```

42. Динамическая маршрутизация

42.1 Списки

Этот раздел в настоящее время отсутствует.

См. документацию Cisco и Quagga.

42.2 RIP

42.2.1 Описание протокола RIP

RIP обеспечивает маршрутизацию внутри автономной системы (AC), 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.

42.2.2 RIP-маршрутизатор

42.2.2.1 Включение RIP-маршрутизатора (router rip)

Первое, что требуется для начала настройки RIP, это включить RIP-маршрутизатор. Сделать это можно командой:

```
(config)# router rip
```

42.2.2.2 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
  enable
  ip address 192.168.0.1/24
!
interface ethernet 1
  enable
  ip address 192.168.1.1/24
!
interface ethernet 0
  enable
  ip address 192.168.2.1/24
!
```

```
(config—rip)# network 192.168.1.1/24
```

42.2.2.3 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.

42.2.2.4 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.

42.2.2.5 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 2|(1 [2])
```

ip rip receive – версия для приема.

ip rip send - версия для передачи.

42.2.2.6 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
```

42.2.2.7 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
```

42.2.3 Метрики

42.2.3.1 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».

42.2.3.2 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
```

42.2.3.3 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.

42.2.4 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.

42.2.4.1 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
!
```

42.2.4.2 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
!
```

42.2.5 Анонсирование

42.2.5.1 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.

42.2.5.2 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
```

42.2.5.3 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>
```

42.2.6 Фильтрация анонсов

42.2.6.1 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.

42.2.6.2 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.

42.3 OSPF

В данном разделе приводится предельно краткая информация о настройке протокола динамической маршрутизации OSPF на Dionis NX C 1.2-10 Hand UTM. Перед прочтением раздела администратору насто- ятельно рекомендуется изучить соответствующую подробную литературу о протоколе OSPF (в частности, документацию от компании Cisco).

42.3.1 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.

42.3.2 Базовая настройка

Активация и настройка службы OSPF на узле

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 . 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.

Все 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).

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>
```

42.3.3 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>
```

42.3.4 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.

42.3.5 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»).

42.3.6 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
```

42.3.7 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>.

42.3.8 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 (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>
```

42.3.9 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.

42.3.10 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.

42.3.11 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
```

42.3.12 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>
```

42.3.13 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, …

42.3.14 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
```

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.

42.3.15 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>	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])]	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.

42.4 BGP

42.4.1 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.

42.4.2 BGP-маршрутизатор

42.4.2.1 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.

42.4.2.2 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
```

42.4.3 BGP-соединение

42.4.3.1 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
```

42.4.3.2 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
```

42.4.3.3 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
```

42.4.3.4 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
```

42.4.3.5 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.

42.4.3.6 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
```

42.4.3.7 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
```

42.4.3.8 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)# neighbor <ip>|<group> password <string>
```

42.4.3.9 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
```

42.4.3.10 IPv6 (activate)

Для разрешения обмена с адресами IPv6 предназначена команда:

```
(config—bgp—65001)# neighbor <ip>|<group> activate
```

Эту команду не следует использовать для IPv4.

42.4.3.11 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
```

42.4.3.12 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
```

42.4.3.13 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
```

42.4.4 Таймеры

42.4.4.1 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>]
```

42.4.4.2 Интервал сканирования BGP-таблиц (scan-time)

BGP-маршрутизатор просматривает свои таблицы на предмет изменений с определенным интервалом (по умолчанию 60 секунд). Для изменения частоты обновления можно использовать команду:

```
(config—bgp—65001)# bgp scan—time <t>
```

<t> - Время в секундах (по умолчанию 60 секунд).

Уменьшение времени сканирования приводит к росту нагрузке на процессор маршрутизатора.

Пример изменения интервала сканирования BGP-таблицы:

```
(config—bgp—65001)# bgp scan—time 5
```

Посмотреть текущие значения интервала обновления можно командой

```
# show bgp scan
```

42.4.4.3 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
```

42.4.4.4 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
```

42.4.5 Анонсирование

42.4.5.1 Анонсирование сети (network)

При первоначальной настройке BGP-маршрутизатор не анонсирует для соседей ничего, кроме своего идентификатора. Анонсировать определенную сеть можно командой:

```
(config—bgp—65001)# 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)# network 10.0.0.0/8
```

Удалить анонсированную ранее сеть можно командой:

```
(config—bgp—65001)# no network <ip/m>
```

42.4.5.2 Проверка наличия маршрута в анонсируемую сеть (import-check)

По умолчанию BGP-маршрутизатор не проверяет существование в таблице IP-маршрутизации маршрута к сети, которую он анонсирует соседям командой network. Проверку существования маршрута можно включить командой:

```
(config—bgp—65001)# bgp network import—check
```

Отменить эту проверку можно командой:

```
(config—bgp—65001)# no bgp network import—check
```

42.4.5.3 Перераспределение (redistribute)

Помимо явного анонсирования сетей при помощи команды network, возможно анонсировать сети путем перераспределения маршрутов из таблицы IP-маршрутизатора в BGP-маршрутизатор. Сделать это можно командой:

```
(config—bgp—65001)# 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)# redistribute connected
```

Удалить перераспределение можно командой:

```
(config—bgp—65001)# no redistribute kernel|connected|static|rip|ospf
```

42.4.5.4 Суммарный путь (aggregate-address)

Суммарный путь для нескольких подсетей анонсируется соседям с целью уменьшения объемов передаваемой информации. Сделать это можно командой:

```
(config—bgp—65001)# 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)# 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)# 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)# no aggregate—address <ip/m>
```

42.4.5.5 Выборочный анонс подавленных путей (unsuppress-map)

Иногда следует разрешить некоторые пути из суммарного пути для определенного соседа. Сделать это можно командой:

```
(config—bgp—65001)#  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)# aggregate—address 192.168.1.0/24 summary—only
(config—bgp—65001)# 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)# no neighbor <ip>|<group> unsuppress—map
```

42.4.5.6 Маршрут по умолчанию (default-originate)

Маршрут по умолчанию можно сообщить соседу и при этом его не нужно создавать в таблице IP-маршрутизации. Сделать это можно следующей командой:

```
(config—bgp—65001)# 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)# 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
```

42.4.6 Фильтрация анонсов

42.4.6.1 Distribute-list

Фильтрация при помощи distribute-list использует списки доступа маршрутизатора. Включить такую фильтрацию можно командой:

```
(config—bgp—65001)# 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)# neighbor 192.168.1.2 distribute—list myacl in
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001)#no neighbor <ip>|<group> distribute—list <racl_name>|<racl_num> in|out
```

42.4.6.2 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)# 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)# neighbor 192.168.1.2 filter—list myasacl out
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001)#no neighbor <ip>|<group> filter—list <apath_name> in|out
```

42.4.6.3 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)# 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)# neighbor 192.168.1.2 distribute—list myplist in
```

Отменить использование этого фильтра можно командой:

```
(config—bgp—65001)#no neighbor <ip>|<group> prefix—list <prlist_name> in|out
```

42.4.6.4 Maximum-prefix

Установить ограничение на количество префиксов (сетей), получаемых от соседа при переполнении памяти, можно командой:

```
(config—bgp—65001)# 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:

```
no neighbor <ip>|<group> maximum—prefix
```

42.4.6.5 Route-map

Фильтрацию при помощи карты маршрутов можно включить командой:

```
(config—bgp—65001)# neighbor <ip>|<group> route—map <rmap_name> in|out|import|export
```

in|out – применяет карту к входящим либо исходящим путям.

import|export – применяет карту к путям, помещаемым в таблицы клиентов сервера маршрутов. Используется совместно с командой route-server-client.

Отменить использование карты маршрутов можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> route—map <rmap_name>
        in|out|import|export
```

42.4.7 Замена полносвязного графа BGP

42.4.7.1 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>
```

42.4.7.2 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)# 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
```

42.4.7.3 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)# neighbor <ip>|<group> route—server—client
```

42.4.7.3.1 Фильтрация при помощи карт маршрутов (route-map import/export, match peer) Сервер создает для каждого клиента свою таблицу с маршрутами, полученными от других клиентов этого сервера. Возможно управлять импортом маршрутов в эту таблицу и экспортом из нее при помощи карты маршрутов. Команда для клиента:

```
(config—bgp—65001)# neighbor <ip>|<group> route—map <rmap_name> import|export
```

Route-map import - фильтрует маршруты, которые приходят в таблицу клиента от других клиентов сервера.

Route-map export – фильтрует маршруты, которые приходят от этого клиента в таблицы других клиентов сервера.

В правилах карты можно использовать критерий выбора адреса соседа:

```
(config—route—map—myrmap)# match peer local|<ip>
```

local – устанавливает совпадение адреса соседа с локальным статическим или перераспределенным адресом.

<ip> - IP-адрес соседа.

Пример использования карты n3mapi для разрешения импорта в таблицу клиента 192.168.0.3 анонсов только от соседа 192.168.0.2 (из AS 65002).

```
(config—bgp—65001)# neighbor 192.168.0.2 remote—as 65002
(config—bgp—65001)# neighbor 192.168.0.2 route—server—client
(config—bgp—65001)# neighbor 192.168.0.3 remote—as 65003
(config—bgp—65001)# neighbor 192.168.0.3 route—server—client
```

```
(config—bgp—65001)# neighbor 192.168.0.3 route—map n3mapi import
(config)# router route—map n3mapi permit 10
(config—route—map—n3mapi)# match peer 192.168.0.2
```

Удалить выбор адреса соседа можно командой:

```
(config—route—map—n3mapi)# no match peer
```

42.4.8 Перемещение BGP-путей в IP-маршрутизатор

42.4.8.1 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.

42.4.8.2 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).

42.4.8.3 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)# 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)# neighbor 192.168.1.2 weight 40000
```

Посмотреть вес можно командой:

```
# show bgp
```

Вернуть значения по умолчанию можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> weight
```

42.4.8.4 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
```

42.4.8.5 Атрибут AS_PATH

42.4.8.5.1 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)#  neighbor <ip>|<group> allowas—in [<n>]
```

Вернуть поведение по умолчанию можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> allowas—in
```

42.4.8.5.2 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
```

42.4.8.5.3 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
```

42.4.8.5.4 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)# neighbor <ip>|<group> remove—private—AS
```

Отменить удаление можно командой:

```
(config—bgp—65001)# no neighbor <ip>|<group> remove—private—AS
```

42.4.8.5.5 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
```

42.4.8.6 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)# 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)# no neighbor <ip>|<group> next—hop—self
```

42.4.8.7 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
```

42.4.8.8 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
```

42.4.8.9 Контроль атрибутов через карту маршрутов

42.4.8.9.1 Вес пути (set weight) Установить вес можно следующей командой:

```
(config—route—map—myrmap)# set weight <n>
```

<n> - вес пути

42.4.8.9.2 Атрибут LOCAL_PREF (set local-preference) Установить атрибут можно следующей командой:

```
(config—route—map—myrmap)# set local—preference <n>
```

<n> - новое значение атрибута

42.4.8.9.3 Атрибут AS_PATH (match/set as-path, pathlimit) Критерий отбора по спискам доступа по AS-путям можно установить следующей командой:

```
(config—route—map—myrmap)#match as—path <as_path_acl>
```

Критерий отбора по атрибуту AS_PATHLIMIT (draft-ietf-idr-as-pathlimit) можно установить следующей командой:

```
(config—route—map—myrmap)#match pathlimit as <n>
```

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.

Установить атрибут AS_PATHLIMIT (draft-ietf-idr-as-pathlimit) можно следующей командой:

```
set pathlimit ttl <n>
```

42.4.8.9.4 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
```

42.4.8.9.5 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
```

42.4.8.9.6 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>
```

42.4.8.10 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)# 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)# no neighbor <ip>|<group> attribute—unchanged
```

42.4.8.11 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)# 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)# distance bgp 200 200 200
```

Возврат значений по умолчанию выполняется командой:

```
(config—bgp—65001)# no distance bgp
```

Если требуется настроить значения AD для путей от конкретного соседа, это выполняется командой:

```
(config—bgp—65001)# distance (<n> <ip> [<racl>])
```

<n> - значение AD.

<ip> - 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. 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)# distance 200 195.220.1.2 myadracl
```

Для удаления изменений AD и возврата значений по умолчанию используется команда:

```
(config—bgp—65001)# no distance <ip>
```

42.4.8.12 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)# 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.

42.4.9 Сообщества

42.4.9.1 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)# 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.

42.4.9.2 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.

42.4.9.3 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>
```

42.4.9.4 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
```

42.4.10 Дополнительные возможности BGP-маршрутизатора

42.4.10.1 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
```

42.4.10.2 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)# neighbor <ip>|<group> capability orf prefix—list send|receive|both
```

42.4.10.3 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]
```

42.4.10.4 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)# neighbor <ip>|<group> soft—reconfiguration inbound
```

После этого можно пользоваться clear bgp soft in.

42.4.10.5 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
```

42.4.10.6 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
```

42.5 Карты маршрутов

42.5.1 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.

42.5.2 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>
```

42.5.3 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, 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
```

42.5.4 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
```

42.5.5 Критерии сравнения

42.5.5.1 Интерфейс

Для ZEBRA, RIP, OSPF, BGP.

Сравнение интерфейса, с которого доступен маршрут.

```
(config—route—map—myrmap)# match interface <iface>
```

<iface> - название интерфейса, например ethernet0.

42.5.5.2 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.

42.5.5.3 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>)
```

42.5.5.4 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>)
```

42.5.5.5 Метрика маршрута (metric). RIP, BGP

```
(config—route—map—myrmap)# match metric <n>
```

<n> - метрика маршрута.

42.5.5.6 Тег маршрута

Для RIP.

```
(config—route—map—myrmap)# match tag \<n\>
```

<n> - тег маршрута.

42.5.6 Установки

42.5.6.1 Метрика маршрута

Для OSPF, BGP.

Устанавливает метрику, либо изменяет существующую

set metric [+|-]<seq>

<seq> - изменение метрики

42.5.6.2 Источник маршрута

Для ZEBRA.

IP-адрес источника маршрута

set src <ip>

42.5.6.3 Адрес шлюза

Для RIP, BGP.

set ip next-hop <ip>

42.5.6.4 Тег маршрута

Для RIP.

set tag <n>

42.5.7 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
```

42.5.8 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».

43. Криптография

43.1 Ключ доступа

Для начала работы с криптографическими средствами 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);
- Совместно используемые (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-индикатора. 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-индикатора.

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 ;
- Один узел может одновременно иметь только один ключ доступа;
- Ключ доступа может быть сохранён только один раз и на одном носителе (флэш, дискета или LCD) после генерации или замены;
- На одном носителе может быть только один ключ доступа.

43.1.1 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 крипто-системы 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.

43.1.2 Генерация нового КД и инициализация ДСЧ

Вставьте внешний носитель с начальным заполнением ДСЧ.

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 flash
```

Данная команда выведет доступные контейнеры в корневой директории флэш-накопителя (для дискет используйте floppy). 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 flash subdir1/subdir2/subdir3...
```

Для генерации КД следует ввести команду:

```
# crypto access key init
```

Данная команда выполнит поиск первого внешнего носителя и попытку чтения контейнера random.ini/gk.db3 (также необходим файл uz.db3). Если контейнер данного типа отсутствует, то будет произведён поиск контейнера PKCS#15 (в корневой директории). Если в корневой директории отсутствуют контейнеры или найдено несколько контейнеров PKCS#15, то необходимо выполнить команду с точным указанием имени файла контейнера. Например:

```
# crypto access key init flash subdir1/file2
```

Если контейнер PKCS#15 защищён паролем, то потребуется ввести пароль.

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».

43.1.3 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-индикатора.

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 flash
```

(или floppy 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.

Далее необходимо выполнить пункт «Загрузка КД».

43.1.4 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 copy running—config startup—config
```

Команда «crypto access key load» осуществляет поиск КД на внешних носителях в следующем порядке: флэш, дискета, ОЗУ LCD, ПЗУ LCD. Если требуется загрузка с определённого носителя, то можно явно его указать. Например:

```
(config)# crypto access key load flash
```

При загрузке КД осуществляется попытка расшифровать начальное заполнение ДСЧ, сохранённое на внутреннем носителе. Неудачное расшифрование означает несоответствие данного КД данному узлу . 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 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 copy running—config startup—config
```

43.1.5 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):

```
# crypto access key clear lcd
# crypto access key clear flash
# crypto access key clear floppy
```

Для **безопасного** удаления ключей с внешних носителей рекомендуется использовать команду “clear removable” (см. раздел “Обслуживание”).

43.1.6 Плановая замена 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-индикатора выполните команду:

```
# crypto access key replace lcd
```

Для замены КД с сохранением на внешнем носителе вставьте носитель со старым КД и выполните команду:

```
# crypto access key replace flash
```

(или floppy для дискеты).

Будет предложено защитить новый КД паролем. Если защита не нужна, нажмите 2 раза Enter.

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.

43.2 Туннели Disec

Система имеет возможность создавать туннели DISEC.

43.2.1 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 flash
# crypto access key store lcd
(config)# crypto access key load lcd
```

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 flash
```

Поддерживаемые внешние ключевые носители: flash, floppy, all (искать ключи как на flash, так и на floppy). 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.

43.2.2 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 ip 1.1.1.1
(config—disec—t1)# remote ip 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 ip: задает IP-адрес локального конца туннеля;
- remote ip: 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 *
```

43.2.3 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, прнимается 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.

43.2.4 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
```

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 A не сможет отключиться, будет предпринята попытка включения успешно отключенных туннелей,если таковые были до момента сбоя в отключении туннеля A. Таким образом, в случае успешности данной команды ВСЕ туннели будут отключены. В случае неуспешности - ничего не изменится.

43.2.5 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.

43.2.6 Работа с ключами

В данном разделе описывается просмотр,удаление и добавление ключей и абонентов.

43.2.6.1 Просмотр ключей

Чтобы посмотреть установленные в систему крипто-ключи, следует использовать команду:

```
# show crypto disec keys
```

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.

43.2.6.2 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
```

43.2.6.3 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 flash
```

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
```

43.2.6.4 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”).

43.2.6.5 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.

43.2.6.6 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.

43.2.6.7 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.

43.2.7 Работа с туннелями

43.2.7.1 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
```

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
```

43.2.7.2 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
```

43.2.7.3 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
```

43.2.8 Прочая работа с DISEC

43.2.8.1 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.

43.2.8.2 Блокирование DISEC трафика

Чтоб полностью заблокировать трафик через все включенные туннели, следует использовать команду:

```
(config—disec)# crypto disec noipcrypto
```

Чтоб разблокировать трафик через все включенные туннели, следует использовать команду:

```
(config—disec)# no crypto disec noipcrypto
```

Трафик будет разблокирован только для тех туннелей, которые не заблокированны индивидуально командой **block**.

43.2.8.3 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
```

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.

43.2.8.4 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
```

43.3 PSK (совместно используемые ключи)

Совместно используемые ключи (Pre-shared Keys, PSK) представляют из себя конфиденциальную последовательность байт (от 8 до 256) 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).

43.3.1 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> <носитель> @<номер_абонента>
```

Номер абонента задаётся в десятичном виде и может быть от 1 до «Number of abonents» включительно. Перед номером обязательно нужно указать символ «@».

Пример:

```
# show crypto psk keys flash
DSRF key container on ’/dev/sdb1’ device:
Zone: 1
Serial: 1234
Abonent: 1
Number of abonents: 9999

# crypto psk set key dsrf_key @9000
Info: Found possible DSRF container on ’/dev/sdb1’ 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> <носитель> <путь_к_файлу>
```

Всё содержимое файла воспринимаются как ключ. Файл должен иметь длину от 8 до 256 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  256  Jul 11 17:42  key2

# crypto psk set key psk1 flash /psks/key1
Info: Read 32 bytes of pre—shared key.
Info: Saving the key with internal name 'psk1'.

# crypto psk set key psk2 flash /psks/key2
Info: Read 256 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
```

43.3.2 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 «* *».

43.3.3 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”).

43.4 PKI (закрытые ключи, сертификаты, COC)

43.4.1 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 СОС.

OCSP - (Online Certificate Status Protocol) - протокол немедленного выяснения действительности сертификата. Данный протокол позволяет отзывать сертификаты более оперативно, чем СОС. 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.

43.4.2 Полная очистка PKI

Если необходимо удалить все закрытые ключи, сертификаты и СОС из системы, следует выполнить команду привилегированного режима:

```
# crypto pki clear all
```

43.4.3 Управление закрытыми ключами

Закрытые ключи устанавливаются в систему с внешних носителей.

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).

Поддерживаются форматы контейнеров закрытых ключей Фактор-ТС и PKCS#15.

43.4.3.1 Импорт ключей

Если на внешнем носителе находится один закрытый ключ (в корневой директории), то для установки его в систему достаточно выполнить команду привилегированного режима:

```
# crypto pki import key
```

Найденный ключ сохранится в системе с тем же именем файла, который имеет контейнер на внешнем носителе. Если в системе уже существует ключ с таким именем, то будет выдано предварительное уведомление (о перезаписывании). Если необходимо дать импортируемому ключу другое системное имя, то следует выполнить команду с параметром «to»:

```
# crypto pki import key to <новое_имя>
```

Если на внешнем носителе находится несколько ключей (или они находятся в поддиректориях), то для просмотра имён контейнеров следует использовать команду:

```
# show crypto pki keys flash|floppy [<путь_к_директории>]
```

Далее следует импортировать ключ с указанием пути к контейнеру. Общий формат import-команды:

```
# crypto pki import key [flash|floppy] [from <путь_к_контейнеру>] [to <новое_имя>]
```

Пример:

Вставляем внешний флэш-носитель и выполняем команду просмотра:

```
# show crypto pki keys flash
certs/
crls/
keys/
pkcs15/
```

Видно, что на внешнем носителе в корневой директории контейнеров с ключами не содержится. Поэтому следует вывести содержимое поддиректории keys.

```
# show crypto pki keys flash keys
key1.nam    oldfactor
key2.nam    oldfactor
```

Видно, что в данной директории находятся два ключа в формате Фактор-ТС. Принимается решение импортировать ключ key1.nam с внутреннем именем old_key1. Для этого следует выполнить команду:

```
# crypto pki import key from keys/key1.nam to old_key1
```

Поскольку предполагается, что в поддиректории pkcs15 тоже могут содержаться контейнеры с ключами, её следует также просмотреть:

```
# show crypto pki keys flash pkcs15
cred1.p15/
cred2.p15/
```

В выводе команды видны две «поддиректории». На самом деле, этими «поддиректориями» могут оказаться PKCS#15-контейнеры, которые отображаются как поддиректории, потому что они могут содержать в себе несколько ключей. Контейнеры PKCS#15 можно просматривать так же, как поддиректории:

```
# show crypto pki keys flash pkcs15/cred1.p15
@00000001   pkcs15
mykey       pkcs15   CN=Иванов Иван Иванович, O=Хорошая организация, C=RU
```

Действительно, «поддиректория» cred1.p15 оказалась файлом контейнера PKCS#15, который содержит два закрытых ключа. В контейнерах PKCS#15 с закрытым ключом могут быть ассоциированы текстовая метка и X500-имя субъекта, которому этот ключ принадлежит. В данном примере ключ mykey имеет эти атрибуты. Если же такие атрибуты у ключа отсутствуют, то он отображается по своему внутреннему числовому идентификатору (в данном примере - ключ @00000001), и о владельце данного ключа можно только догадываться.

Принимается решение импортировать оба ключа (с внутренними именами unknown_key1 и ivan_key соответственно):

```
# crypto pki import key from pkcs15/cred1.p15/@00000001 to unknown_key1
# crypto pki import key from pkcs15/cred1.p15/mykey to ivan_key
```

Если контейнер PKCS#15 защищён паролем, то потребуется его ввести.

43.4.3.2 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
```

43.4.3.3 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”).

43.4.4 Управление сертификатами

В 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, PKCS#15, 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.

43.4.4.1 Импорт сертификатов

Команды просмотра и импорта сертификатов имеют синтаксис, похожий на команды просмотра и импорта ключей.

Чтобы просмотреть сертификаты на внешнем носителе, нужно выполнить команду (в привилегированном режиме):

```
# show crypto pki certs flash|floppy [<путь>]
```

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.

Следующие команды выполняют импорт соответственно корневых, промежуточных, клиентских и OCSP-сертификатов:

```
# crypto pki import root ca cert [<внешний_носитель>] [from <путь_к_файлу>] [to
      <новое_имя>]
# crypto pki import ca cert [<внешний_носитель>] [from <путь_к_файлу>] [to <новое_имя>]
# crypto pki import cert [<внешний_носитель>] [from <путь_к_файлу>] [to <новое_имя>]
# crypto pki import ocsp cert [<внешний_носитель>] [from <путь_к_файлу>] [to <новое_имя>]
```

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 keys flash
certs/
crls/
keys/
pkcs15/
```

Просмотрим директорию «certs»:

```
# show crypto pki certs flash certs
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. Мы принимаем решение его установить в систему, так как доверяем данной организации и сертификатам, выпущенным её УЦ:

```
# crypto pki import root ca cert from 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 certs/ocsp.cer to right_org_ocsp_cert
```

Мы знаем что директория «pkcs15» содержит контейнеры PKCS#15 с ключами и необходимыми цепочками сертификатов. Просматриваем эти контейнеры:

```
# show crypto pki certs flash pkcs15
cred1.p15/
cred2.p15/

# show crypto pki certs flash pkcs15/cred1.p15
@00000001   user   CN=Петров Пётр Петрович, O=Хорошая организация, C=RU
@00000002   root   CN=Главный УЦ, O=Хорошая организация, C=RU
@00000003   ca     CN=УЦ первого филиала, O=Хорошая организация, C=RU
mycert      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. Сертификатам УЦ не присвоена текстовая метка в PKCS#15 контейнере, поэтому мы видим их идентификаторы.

Также мы видим сертификат Петрова Петра Петровича, который имеет внутренний идентификатор 0x00000001. В формате PKCS#15 есть требование о том, что если в контейнере хранится закрытый ключ и соответствующий ему сертификат, то их идентификаторы должны совпадать. Мы вспоминаем, что мы импортировали некий неизвестный нам ключ с идентификатором 0x00000001 (в примере с закрытыми ключами - см. выше) и присвоили ему внутреннее имя «unknown_key1». Теперь мы видим, что этот ключ принадлежит Петрову Петру Петровичу. Мы понимаем, что его ключ и сертификат нам не нужны, и принимаем решение удалить этот ключ:

```
# crypto pki clear key unknown_key1
```

Мы импортируем сертификаты наших удостоверяющих центров и сертификат Иванова Ивана Ивановича:

```
# crypto pki import root ca cert from pkcs15/cred1.p15/@00000002 to our_main_ca_cert
# crypto pki import ca cert from pkcs15/cred1.p15/@00000003 to our_local_ca_cert
# crypto pki import cert from pkcs15/cred1.p15/mycert 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
```

Контейнеры P7B, как и контейнеры PKCS#15, отображаются как «директории», и могут содержать сертификаты и COC. Пример:

```
# show crypto pki certs flash
cacer.p7b/
```

Файл «cacer.p7b» отображается как директория, потому что он содержит хотя бы один сертификат. Просматриваем сертификаты в контейнере:

```
# show crypto pki certs flash cacer.p7b
cert01   root   CN=Главный УЦ, O=Хорошая организация, C=RU
cert02   ca     CN=УЦ первого филиала, O=Хорошая организация, C=RU
```

Внутри контейнера P7B сертификаты идентифициируются по порядковым номерам. Импортируем сертификаты:

```
# crypto pki import root ca cert from cacer.p7b/cert01 to our_main_ca_cert
# crypto pki import ca cert from cacer.p7b/cert02 to our_local_ca_cert
```

43.4.4.2 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>
```

Просмотреть отпечатки (fingerprints) открытых ключей сертификатов можно с помощью команд:

```
# show crypto pki root ca cert <имя> keyids
# show crypto pki ca cert <имя> keyids
# show crypto pki cert <имя> keyids
# show crypto pki ocsp cert <имя> keyids
```

43.4.4.3 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
```

43.4.5 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 COC 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, PKCS#15, P7B.

43.4.5.1 Импорт COC

Импорт списков отозванных сертификатов похож на импорт закрытых ключей. (Если контейнер PKCS#15 защищён паролем, то ввод пароля не потребуется, потому что COC не является конфиденциальной информацией).

Для просмотра COC на внешних носителях и импорта следует использовать команды привилегированного режима:

```
# show crypto pki crls flash|floppy [<путь_к_директории>]
# crypto pki import crl [flash|floppy] [from <путь_к_файлу>] [to <новое_имя>]
```

Контейнеры P7B помимо самих сертификатов могут содержать списки отзыва сертификатов. Если контейнер содержит хотя бы один COC, то следующая команда отобразит его как «директорию»:

```
# show crypto pki crls flash
cacer.p7b/
```

Пытаемся импортировать COC из P7B:

```
# crypto pki import crl from 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 flash 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. Импортируем оба списка:

```
# crypto pki import crl from cacer.p7b/crl01 to our_main_ca_crl
# crypto pki import crl from cacer.p7b/crl02 to our_local_ca_crl
```

43.4.5.2 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>
```

43.4.5.3 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
```

43.5 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 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).

В 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.

43.5.1 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.

43.5.1.1 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 43.1 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;

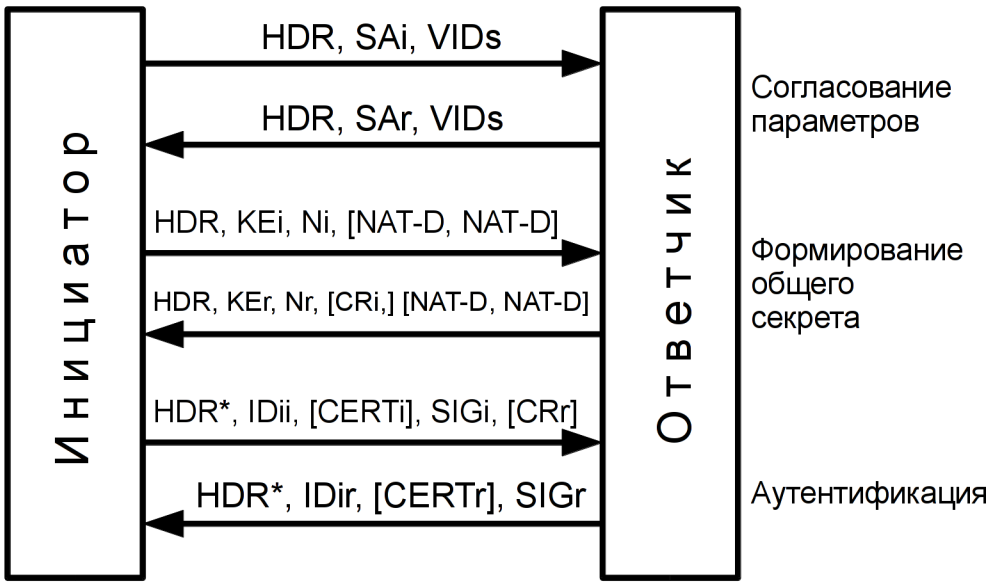

Рис. 43.1: Фаза 1 (PKI)

- VIDs - идентификаторы VendorID, уведомляющие о дополнительных возможностях: Dead Peer Detection (RFC3706), NAT Traversal (RFC3947), GOST - IPsec по стандарту ООО «КриптоПро»;
- 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;
- 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. 43.2).

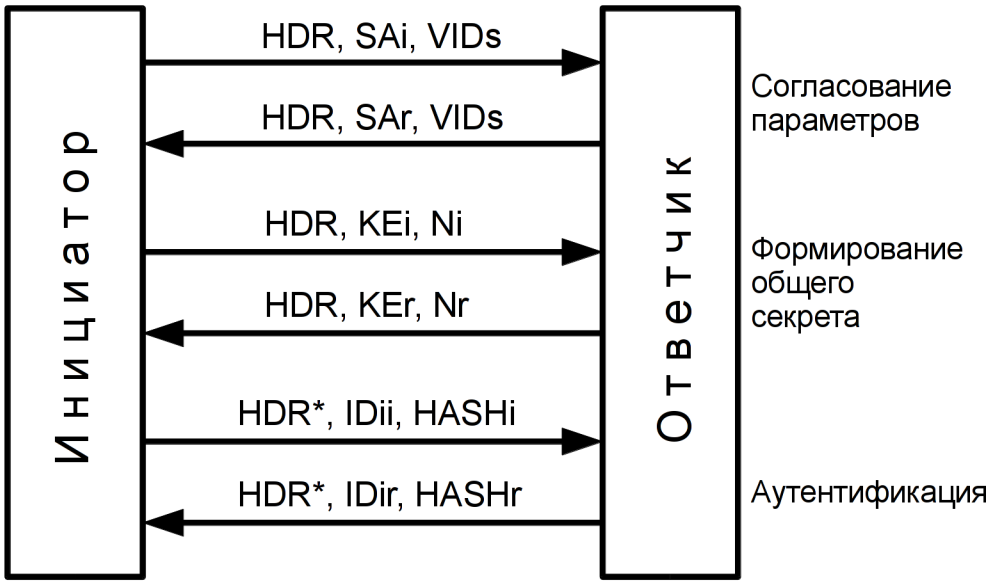

Рис. 43.2: Фаза 1 (PSK)

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.

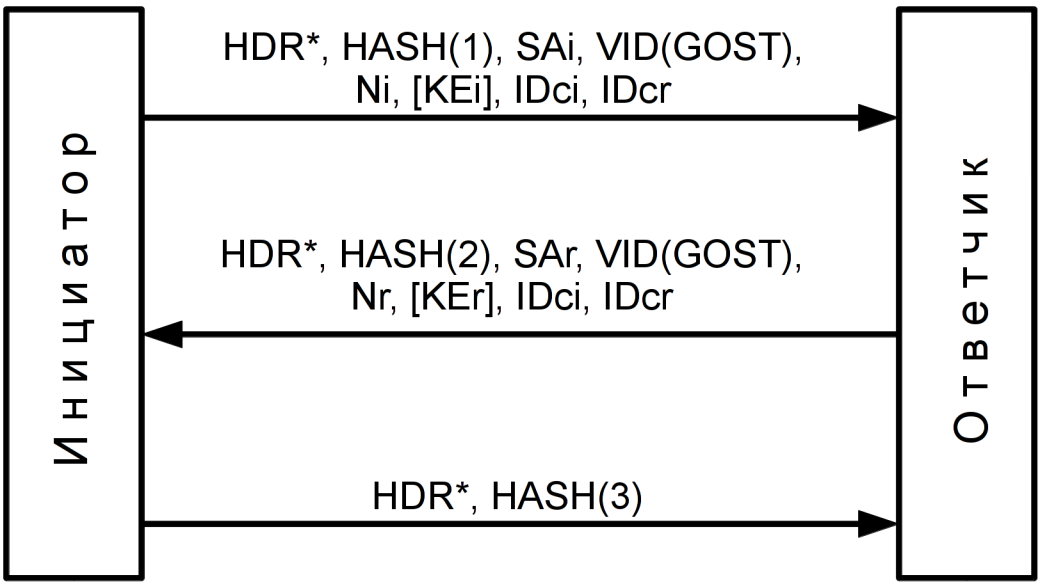

Рис. 43.3: Фаза 2

43.5.1.2 Фаза 2

Фаза 2 показана на рисунке 43.3.

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;
- 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.

43.5.1.3 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. 43.4), которая происходит между фазой 1 и фазой 2.

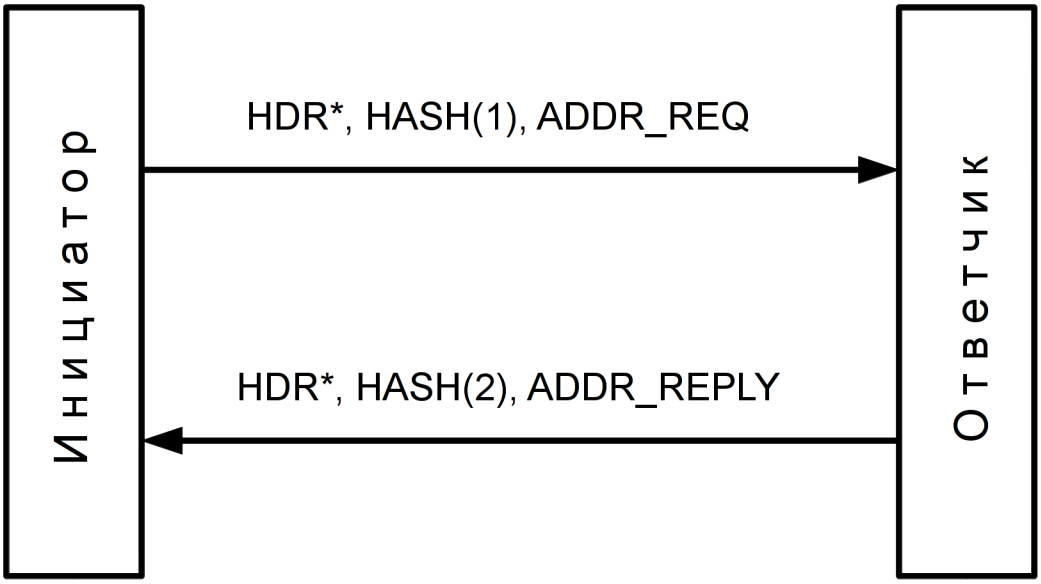

Рис. 43.4: Фаза ModeConfig

Условные обозначения:

- ADDR_REQ - запрос клиентского адреса у сервера;
- ADDR_REPLY - назначение адреса клиенту.

43.5.1.4 Уведомление сторон

Стороны могут слать друг другу асинхронные уведомления на любом этапе установления туннеля или в процессе работы туннеля (см. рис. 43.5).

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.

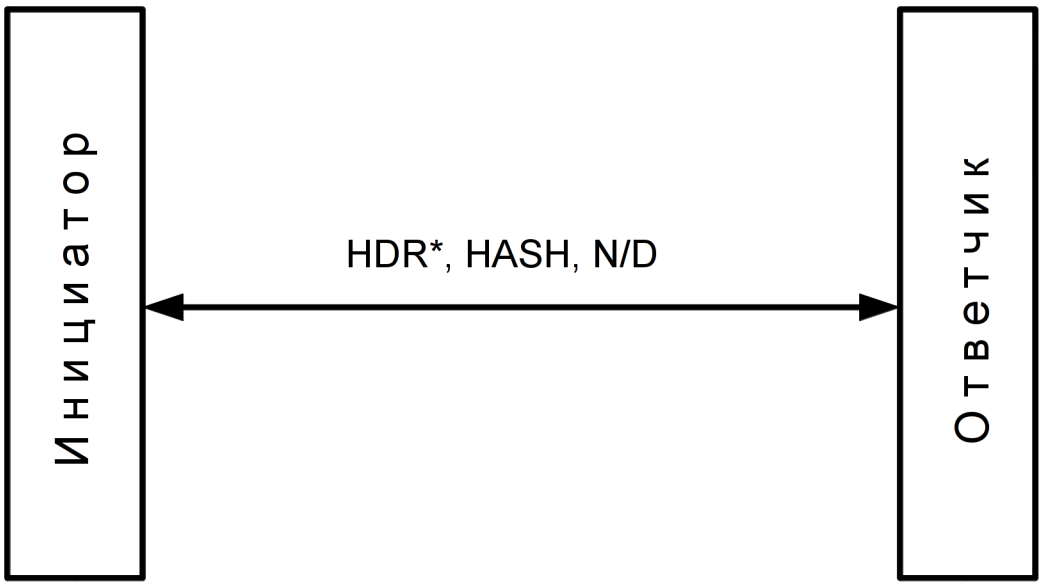

Рис. 43.5: Сообщения уведомлений

43.5.2 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.

43.5.3 Соединения IPsec

Необходимо ввести понятие «соединения IPsec».

Одно соединение IPsec - это одна из следующих конфигураций сети (см. рис. 43.6).

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.

43.5.4 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 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 keys/router1.nam
# crypto pki import root ca cert from certs/ca.cer
# crypto pki import cert from 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:

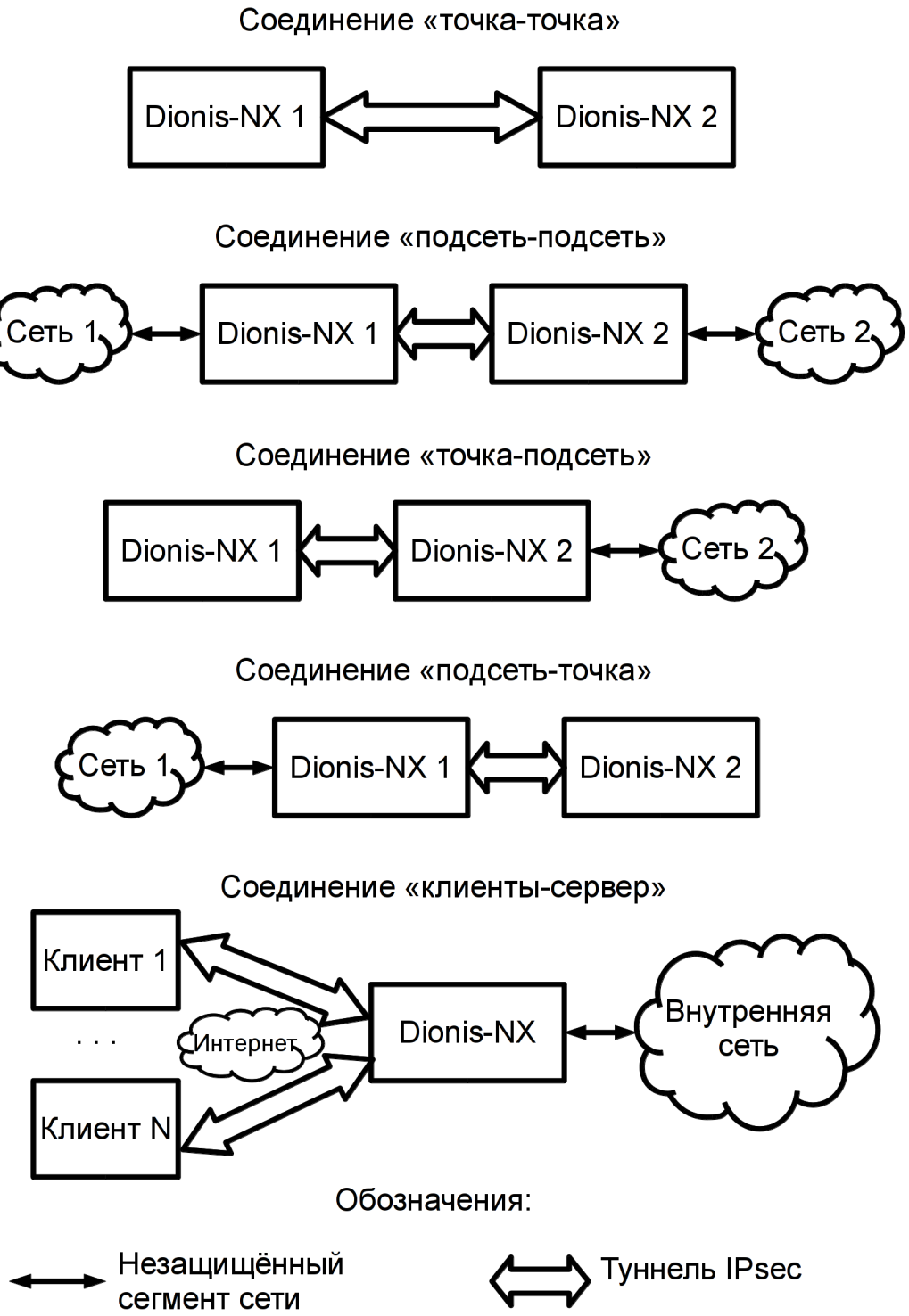

Рис. 43.6: Виды соединений IPsec

```
(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:

```
(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 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 keys/router2.nam
# crypto pki import root ca cert from certs/ca.cer
# crypto pki import cert from 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».

43.5.5 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, из файла /psks/key1):

```
# crypto psk set key psk1 flash /psks/key1
```

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 flash /psks/key1
# 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
```

43.5.6 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.

Например:

```
# configure terminal
(config)# ip access—list ipsec_only
(config—acl—ipsec_only)# permit udp sport 500 dport 500
(config—acl—ipsec_only)# permit udp sport 4500 dport 4500
(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
```

В данном примере 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.

43.5.7 Управление и диагностика службы IKE

Управление туннелями IPsec осуществляется через службу IKE.

43.5.7.1 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.

43.5.7.2 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
```

43.5.7.3 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
```

43.5.7.4 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).

43.5.7.5 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.

43.5.8 Управление и диагностика состояний соединений

43.5.8.1 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).

43.5.8.2 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.

43.5.8.3 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
t4 [2] online
```

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>
```

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.

43.5.8.4 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.

43.5.8.5 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.

43.5.8.6 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».

43.5.8.7 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).

43.5.8.8 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».

43.5.8.9 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.

43.5.8.10 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 165 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.

43.5.8.11 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
```

43.5.8.12 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);
- 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.

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).

43.5.8.13 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».

43.5.8.14 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»);
- 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»).

43.5.8.15 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> 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.

43.5.8.16 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]
```

43.5.8.17 Вывод всех настроек соединения

Чтобы вывести полную конфигурацию соединения вместе с настройками по умолчанию, нужно выполнить команду:

```
# show crypto ike conn <имя> config
```

43.5.9 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»).

43.5.10 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»).

43.5.11 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».

43.5.12 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».

43.5.13 Соединения “клиенты-сервер”

43.5.13.1 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
t1  [2]  online
t1  [3]  pending2
```

В этом случае команда «crypto ike close conn» закроет сразу все порождённые соединения:

```
# crypto ike close conn t1
# show crypto ike conn t1
t1        listen
```

43.5.13.2 Шаблон IP-адреса клиента

Следующая опция позволяет устанавливать туннель с любого IP-адреса:

```
(config-ike-conn-<имя>)# remote ip *
```

43.5.13.3 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.

43.5.13.4 “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
```

43.5.13.5 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
```

43.5.13.6 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.

43.5.13.7 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.

43.5.13.8 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.

43.5.13.9 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>]
```

Удалить опции «local/remote source ip» из конфигурации соединения можно с помощью соответствующих команд «no».

43.5.13.10 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.

43.5.14 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–…»);
- Если используется аутентификация по pre-shared ключам (не рекомендуется), то на узле, противоположном оппоненту, и находящимся за NAT, необходимо ассоциировать PSK с «*» (совпадает с опцией «remote ip *»).

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.

43.5.15 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 опция принудительной инкапсуляции в UDP работает корректно только для соединений типа «клиент-сервер». То есть на стороне сервера обязательно должна присутствовать опция «remote ip *».

43.5.16 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».

43.5.17 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
```

43.5.18 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.

43.5.18.1 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.

43.5.18.2 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). 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.

43.5.18.3 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

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.

43.5.18.4 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

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
```

43.5.18.5 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.

43.5.18.6 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).

43.5.18.7 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 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.

43.5.19 Продление и закрытие туннелей. Таймеры

43.5.19.1 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).

43.5.19.2 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.

43.5.19.3 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).

43.5.19.4 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».

43.5.19.5 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
```

43.5.20 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.

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.

43.5.21 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.

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».

43.5.22 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 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

43.5.23 COC 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 (COC). Протокол OCSP имеет больший приоритет, как более оперативный.

43.5.23.1 Ручная загрузка COC

Списки отозванных сертификатов могут быть загружены в систему вручную с помощью команды «crypto pki import crl» (см. выше). При запуске службы IKE, все COC, находящиеся в локальном хранилище, загружаются в оперативную память. Если при работающей службе IKE в локальное хранилище были загружены дополнительные COC, то для того, чтобы они вступили в силу, требуется выполнить команду режима enable:

```
# crypto ike reload
```

43.5.23.2 “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 COC). Если OCSP-информация недоступна и нет соответствующего действительного COC, то сертификат считается заведомо недействительным.

При нестрогой политике сертификат проверяется на отзыв, если есть соответствующая информация (OCSP или COC). 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
```

43.5.23.3 Загрузка новых COC по сети

Помимо статически загруженных COC, служба IKE может динамически получать COC по протоколам HTTP/FTP/LDAP.

ВАЖНО: По умолчанию динамическая загрузка COC 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.

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.

43.5.23.4 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 <uri>	Дополнительная точка распространения COC (HTTP/FTP/LDAP)
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)

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)# 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.

43.5.23.5 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
```

43.5.23.6 Кэширование COC

Иногда бывает полезно сохранять динамически загруженные COC 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 COC можно таким же образом, как и загруженными вручную, то есть с помощью команд «*crypto pki * crl*» (см. выше).

Выключить сохранение загружаемых COC можно соответствующей опцией «no».

```
(config—ike)# no crl cache
```

43.5.23.7 Настройка протокола OCSP

В 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 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 требуется обрабатывать большое количество соединений от разных оппонентов.

43.5.23.8 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.

43.5.23.9 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.

43.5.24 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
```

43.5.25 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 flash
DSRF key container on ’/dev/sdb1’ device:
Zone: 1
Serial: 1234
Abonent: 1
Number of abonents: 9999

# crypto psk set key psk2 @9000
Info: Found possible DSRF container on ’/dev/sdb1’ 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 @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 @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 канал с ним должен восстановиться.

ВНИМАНИЕ: Удалённое переключение ключа узла 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 :

```
# 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-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-ТС,C=RU”
...
```

Действия на узле 1:

```
# show crypto pki keys flash
user1.p15/
# crypto pki import key from user1.p15 to user1.key.2
# show crypto pki certs flash
user1.cer   user  CN=user1,O=Фактор-ТС,C=RU
# crypto pki import cert from 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 установ-

лены 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-TC,C=RU
# show crypto pki root ca certs
ca.cer      CN=УЦ,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”
...
```

Импорт нового сертификата УЦ/узла, нового закрытого ключа:

```
# show crypto pki certs flash
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 ca.cer to ca.cer.2
# crypto pki import cert from user1.cer to user1.cer.2
# show crypto pki keys flash
user1.p15/
# crypto pki import key from 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
```

43.5.26 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.

43.5.27 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).

43.5.28 Защита от replay-атак

В реализации протокола ESP в системе 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>
```

43.5.29 Лицензии на соединения

В некоторых конфигурациях 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).

43.5.30 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
```

43.5.31 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.

44. 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.

44.1 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.

44.2 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.

44.2.1 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
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

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.

44.2.2 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
```

44.2.3 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.

45. Отказоустойчивый кластер

Типичная схема использования отказоустойчивого кластера на основе маршрутизаторов представлена на рисунке:

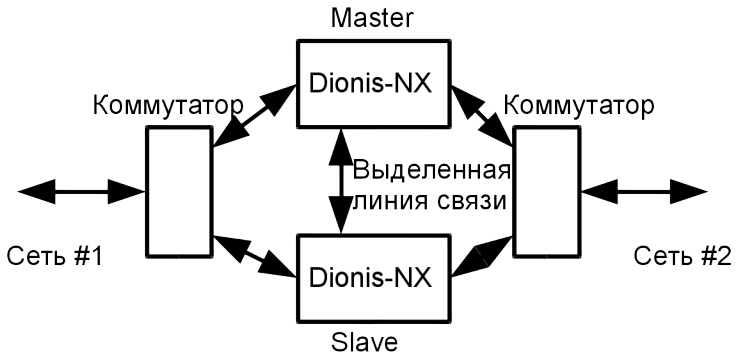

Рис. 45.1: Отказоустойчивый кластер

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.

45.1 Требования к оборудованию

Для организации отказоустойчивого кластера на основе системы 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 .

45.2 Подготовка к организации кластера

Перед объединением двух маршрутизаторов в кластер, на каждый из них необходимо установить одинаковую версию системы 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. 5.2. 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 автоматического формирования IP-адресов основного и резервного маршрутизатора, а также для уникальной идентификации кластера. Для кластеров, находящихся в одной физической сети, выбранные подсети выделенной линии должны отличаться.

Для установки параметров кластера необходимо выполнить следующую команду в режиме конфигурирования:

```
Router(config)# cluster
```

Конфигурирование кластеров осуществляется с использованием команды aux. Она указывает, по какому интерфейсу и по какой подсети организуется выделенная линия. Для минимальной предварительной настройки основного маршрутизатора, необходимо выполнить следующие команды в режиме конфигурирования кластера:

```
Router(config—cluster)# aux interface ethernet 2
Router(config—cluster)# aux network 192.168.10.0/24
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. Для минимальной предварительной настройки резервного маршрутизатора, необходимо в режиме конфигурирования кластера выполнить следующие команды :

```
Router(config—cluster)# slave
Router(config—cluster)# aux interface ethernet 2
Router(config—cluster)# aux network 192.168.10.0/24
Router(config—cluster)# enable
```

Команда «slave» в данном примере говорит о том, что данный маршрутизатор является резервным. По-умолчанию, маршрутизатор считается основным.

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.

45.3 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. 45.2.

Для кластера может быть настроен интервал между посылками пакета жизни 45 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 милисекундах. 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.

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. Интервал времени между операциями обмена информацией можно установить командой:

```
Router(config—cluster)# conntrack poll 10
```

Интервал времени выражается в секундах и по умолчанию равен 5 секундам.

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.

45.4 Получение информации о кластере

Получить информацию о текущем состоянии кластера можно с помощью следующей команды, выполненной из привилегированного режима:

```
Router# show cluster

State:          enable
Mode:           master
Interface:      ethernet2
Network:        192.168.77.0/24
Advert period:  500 ms (default)
Advert timeout: 1500 ms (default)
ConnTrack poll: 5 sec (default)

Current config settings (not active):
Mode:           master
Interface:      ethernet2
Network:        192.168.77.0/24
Advert period:  1000 ms
Advert timeout: 1500 ms (default)
ConnTrack poll: 10 sec

Warning: The current settings is not equal to active settings.
Warning: Disable cluster and then enable it to fix differencies.
```

В данном примере показан самый полный вывод команды «show cluster» - в случае рассинхронизации настроек работающего кластера и настроек в running-config. В начале вывода показаны настройки работающего кластера. Далее показаны измененные настройки из текущей конфигурации. В завершении выводятся предупреждения о рассинхронизации.

Если кластер запущен и рассинхронизации настроек нет, то будут показаны только настройки работающего кластера. Если кластер остановлен, будут показаны только настройки из текущей конфигурации.

Свойство «State» показывает, запущен ли кластер:

- enable - кластер запущен;
- disable - кластер остановлен.

Свойство «Mode» отображает текущий режим работы маршрутизатора:

- master - маршрутизатор является основным;
- slave - маршрутизатор является резервным;
- slave (temp master) - маршрутизатор является резервным, но выполняет функции основного, так как основной вышел из строя.

Свойства «Interface» и «Network» отображают настройки интерфейса и подсети для выделенной линии связи между основным и резервным маршрутизаторами.

Свойства «Advert period» и «Advert timeout» отображают период посылки пакета жизни основным маршрутизатором и тайм-аут, по истечении которого основной маршрутизатор считается неработоспособным.

Свойство «ConnTrack poll» отображает период времени между операциями обмена информацией об установленных соединениях.

45.5 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.

FIXME TODO - Написать о разных типах ключей. С.Каличев

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, предусмотрена команда:

```
Router# cluster connect
```

К имени хоста в приглашении командной строки на резервном маршрутизаторе будет добавлена строка »-slave» для того, чтобы администратор легко мог отличить основной маршрутизатор от резервного.

```
Router—slave#
```

45.6 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
```

46. Обновление системы

Администратор имеет возможность производить обновление системы. 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.

46.1 DIP-пакеты

Обновления системы 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, образ корневой файловой системы

.

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
...
```

46.2 Инфраструктура DIP

На маршрутизаторе может быть одновременно установлено несколько экземпляров операционной системы. Это могут быть и разные версии ОС,и несколько экземпляров системы одной версии. Возможность использования нескольких версий ОС нужна для безопасного обновления системы, а также для организации отката (fallback) к работоспособному экземпляру системы при возникновении сбоев работы текущей работающей системы.

Все установленные экземпляры операционной системы (будем называть их пакетами ОС или OS package) доступны только для чтения. Дополнительные данные для операционных систем (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 ОС 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 ОС, не имеющие своего слота данных, а также слоты данных, не привязанные ни к одному пакету.

К примеру, вновь установленное обновление ОС 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 ОС идентифицируется уникальным именем. Каждый слот данных также идентифицируется уникальным именем. Используя эти уникальные имена, администратор системы может производить различные действия над пакетами ОС и слотами данных. Операции над текущим (активным) пакетом ОС и активным слотом данных ограничены, так как невозможно, к примеру, удалить текущий слот данных, не нарушив работу маршрутизатора.

Команды для пакетов ОС:	
Команда	Назначение
os install	Установка нового пакета ОС. Источником является DIP-пакет
os remove	Удаление существующего пакета ОС. Невозможно для активного пакета
os rename	Переименование существующего пакета ОС. Меняется уникальное имя пакета в системе
os export	Экспорт существующего пакета ОС. Будет создан DIP-пакет
os bind	Привязка пакета ОС к существующему слоту данных. Невозможно для активного пакета ОС. Невозможно для уже привязанного слота данных
os bind	Отвязывание пакета ОС от слота данных. Невозможно для активного пакета ОС
show os	Получение списка установленных пакетов ОС
show os info	Получение подробной информации об установленных пакетах ОС

Команды для слотов данных:

Команда	Назначение
os data create	Создание нового пустого слота данных
os data clone	Клонирование слота данных. Создается новый слот, дублирующий содержимое исходного слота
os data remove	Удаление существующего слота данных. Невозможно для слотов, привязанных к какому-либо пакету ОС
os data rename	Переименование существующего слота данных
os data backup	Создание резервной копии данных на основании существующего слота данных. Создается файл - резервная копия

os data restore	Восстановление слота данных на основании резервной копии данных. Невозможно для активного слота данных
schedule backup	Безопасное создание резервной копии активного слота с перезагрузкой
schedule restore	Восстановление текущего слота данных на основании резервной копии
schedule migrate	Миграция на другой пакет ОС с сохранением текущего слота данных. Требуется перезагрузка
show os data	Получить список существующих слотов данных

Команды загрузки системы:

Команда	Назначение
boot default	Задать пакет ОС, который будет загружаться по умолчанию
boot fallback	Задать пакет ОС, который будет загружен в случае необходимости отката к предыдущей версии (fallback).
boot experimental	Установить для пакета ОС признак того, что эта ОС является экспериментальной
show boot	Получить текущую конфигурацию загрузчика

Общие операции:

Команда	Назначение
show os summary	Получить сводку состояния DIP-инфраструктуры

46.3 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. 28.4). 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. Каждая установленная версия системы 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 уста-

новленных ОС.

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 . 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 ОС может быть переименован. Новое название должно быть уникально:

```
Router# os rename dionisnx—1.0—0 mysystem
```

После этого пакет ОС в системе идентифицируется новым именем «mysystem».

Если какой-либо пакет ОС устарел и не используется, его можно удалить:

```
Router# os remove dionisnx—0.9—0
```

46.4 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 ОС 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 тайм-аута, 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
```

46.5 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 ОС 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.

46.6 Привязка данных

Слот данных может быть привязан к установленному пакету ОС. Это означает, что при загрузке этой ОС для хранения конфигурации и данных будет использован именно привязанный слот данных.

Привязка слота данных к пакету ОС производится следующей командой:

```
Router# os bind dionisnx—1.0—0 data~1
```

Где «dionisnx-1.0-0» - установленный пакет ОС, 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 ОС.

Операция по привязке слота данных «os bind» не может быть выполнена для слота данных, который уже привязан к какому либо пакету ОС. Такой слот необходимо сначала отвязать и только потом использовать:

```
Router# os bind dionisnx—1.0—0
```

Данная команда (без указания слота данных) отвяжет пакет ОС 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 ОС и текущим слотом данных смотрите раздел «Миграция ОС».

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 ОС. Первое поле - идентификатор (имя) системы. Поле в фигурных скобках показывает версию системы. Поле в квадратных скобках указывает на привязанный слот данных. После установленных пакетов ОС перечислены существующие слоты данных. Первое поле - имя слота данных. Поле в квадратных скобках - пакет ОС, к которому привязан слот.

46.7 Миграция ОС

Часто возникает ситуация, когда администратор хочет установить новую версию ОС, 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 ОС 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 ОС. Эта задача также решается через перезагрузку.

```
Router# schedule rebind data_new
```

Команда планирует привязку слота данных «data_new» к текущему пакету ОС. После перезагрузки будет загружена старая система с новым слотом данных.

46.8 Резервная копия пакета ОС

Администратор может создать DIP-пакет из уже установленного пакета ОС. 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.

47. Обслуживание

47.1 Резервное копирование

В системепредусмотрено 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.

47.1.1 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:», а также на флеш-носитель (например flash0.1:). В конце команды могут быть введены следующие опции:

- config-only - Сохранять только основные файлы конфигурации. Самый быстрый способ;
- no-log - Не сохранять файлы журналирования;
- 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 no—log
```

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”. В данном случае использована опция «no-log».

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.

47.1.2 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
```

47.1.3 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 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.

47.2 Проверка файловых систем

Несмотря на то, что файловая система EXT3 (системный раздел и раздел данных системы является журналируемой, в некоторых нештатных ситуациях требуется принудитель- ная 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.

47.3 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.

47.4 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.

47.5 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.

48. Приложение

48.1 Примеры конфигураций

48.1.1 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;
- Созданы (но не применены) классы QoS, соответствующие классам изделия Dionis-LX;
- Сервис протоколирования настроен по умолчанию;
- Минимально настроен (но выключен) сервис DNS;
- Минимально настроен (но выключен) сервис NTP;
- Включен сервис SSH для оператора cli;
- Включена маршрутизация пакетов.

```
!
timezone MSK−4
!
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 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 tos 0x00/0xe0
 class prt1 rate 1kbit ceil 10000mbit priority 6 tos 0x20/0xe0
 class prt2 rate 1kbit ceil 10000mbit priority 5 tos 0x40/0xe0
 class prt3 rate 1kbit ceil 10000mbit priority 4 tos 0x60/0xe0
 class prt4 rate 1kbit ceil 10000mbit priority 3 tos 0x80/0xe0
 class prt5 rate 1kbit ceil 10000mbit priority 2 tos 0xa0/0xe0
 class prt6 rate 1kbit ceil 10000mbit priority 1 tos 0xc0/0xe0
 class prt7 rate 1kbit ceil 10000mbit priority 0 tos 0xe0/0xe0
!
interface ethernet 0
 enable
 ip address 192.168.1.1/24
!
ip access—list no—invalid
 deny state invalid
!
service log
 log
 trace all
 size 262144 131072
 alert beep
!
service dns
 log all—first info
 listen localhost
 view default
  auto—local—zones
  zone .
   auto static
!
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
!
service ssh
 enable
!
ip forwarding
```

48.1.2 Пример файерволла

```
!
! $System: DionisNX
! $Version: 1.0.0d
! $Date: 2012—06—20 11:29:10
!
timezone MSK—4
session timeout adm none
!
hostname raul
!
ip path—mtu—discovery
ip tcp ecn server—mode
ip tcp selective—ack
ip tcp syncookies
ip tcp timestamps
!
interface ethernet 0
 enable
 ip address 83.220.32.68/27
 ip access—group outside in
 ip nat—group masq
!
interface ethernet 1
 enable
 ip address 192.168.16.58/24
 ip access—group in16 in
!
interface ethernet 2
 enable
 ip address 192.168.33.254/24
 ip access—group int33 in
 ip nat—group squid
!
ip route 0.0.0.0/0 83.220.32.65
ip route 192.168.0.0/24 192.168.16.1
ip route 192.168.32.0/24 192.168.16.1
```

```
!
ip resolver domainname cuba.int
!
ip resolver nameserver 192.168.33.254
!
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 83.220.32.68 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 83.220.32.68
!
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
!
service log
 log
 trace
 size 262144 131072
 alert beep
!
service dns
 log all—first info
 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
 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
  auto—local—zones
  zone .
   auto weekly
  zone forward 16.168.192.in—addr.arpa.
   forwarders 192.168.16.3 192.168.16.4
  zone forward factor—ts.int.
   forwarders 192.168.16.3 192.168.16.4
  zone forward factor—ts.net.
   forwarders 192.168.16.3 192.168.16.4 192.168.16.1
  zone forward factor—ts.ru.
   forwarders 192.168.16.3 192.168.16.4 192.168.16.1
  zone master 33.168.192.in—addr.arpa.
   update
   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.
  zone master cuba.int.
   update
   ttl  604800
   soa master raul admin root@raul.cuba.int negttl 604000
   a 192.168.33.1 domain fidel
   a 192.168.33.160 domain havana
   a 192.168.33.254 domain raul
   a 192.168.33.3 domain pkunistan
   a 192.168.33.6 domain vova—ipsec
   ns raul
 enable
!
service dhcp
 default—lease—time 345600
 listen ethernet 2
 max—lease—time 400000
```

```
 min—lease—time 86400
 broadcast—address 192.168.33.255
 domain—name cuba.int
 router 192.168.33.254
 search cuba.int
 search factor—ts.int
 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
 listen 3127 192.168.33.254 intercept
 cache replacement—policy disk gdsf
 cache replacement—policy memory gdsf
 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
 admin—email admin@factor—ts.ru
 cache disk—size 4096
 cache limit disk max 500
 cache limit disk min 0
```

```
 cache limit memory max 16
 cache memory—size 16
 cache type aufs
 log access
 log cache high
 log—access deny nolog
 log—access permit all
 http—access deny bad1
 http—access permit net33
 http—access deny all
 caching deny lan
 caching permit all
 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
 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
```

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					