

Лабораторная работа №1. Установка и ознакомление с EVE-NG (UNETLAB).

1.1 Цель работы

Целью работы является ознакомление с эмулятором EVE-NG, уяснение процесса установки эмулятора и получение базовых навыков по установке образов оборудования.

1.2 Теоретические сведения

В настоящее время большой популярностью пользуется сетевой симулятор Cisco Packet Tracer, который разработан компанией Cisco, выпускающей профессиональное сетевое оборудование и предназначенный для использования при изучении сетевых технологий, а также для моделирования сетей различного масштаба перед их физическим построением с целью выявления возможных недостатков и создания оптимальной конфигурации оборудования.

Помимо Cisco Packet Tracer стоит отметить другой популярный проект – UNetLab. Это мульти-вендорная и многопользовательская платформа для моделирования различных сетей с применением широкого спектра оборудования: коммутаторов, маршрутизаторов и т.п.

На данный момент проект UNetLab претерпел некоторые изменения, в частности было изменено и название: EVE-NG (Emulated Virtual Environment Next Generation). EVE-NG имеет ряд ключевых особенностей:

- Моделирование различных топологий в интерактивном режиме
- Поддержка многопользовательского проектирования путем размещения проектов в режиме ограниченного доступа
- Поддержка пользовательских ядер для протоколов уровня L2
- Оптимизация памяти
- Аппаратное ускорение
- Интеграция и взаимодействие с реальными сетями
- Удобный интерфейс на HTML5
- Возможность быстрого переноса реальной топологии в виртуальную среду и множество других особенностей.

Выделим три основных составляющих EVE-NG:

- **Dynamips** – позволяет осуществлять запуск оборудования серии 7200, 3725, 1710 и другие образы. Однако, для перечисленных моделей обеспечивается наибольшая поддержка и производительность;
- **IOL** – обеспечивает поддержку оборудования L2 и L3;
- **QEMU** – позволяет запускать огромное количество устройств Cisco ASA, Fortinet, Juniper.

По большому счёту, в EVE-NG есть возможность запустить всё, что поддаётся виртуализации, что открывает большой простор для действий.

EVE-NG поддерживает большое количество образов сетевого оборудования различных производителей, в том числе Cisco, Alcatel, Juniper, Fortinet.

Для большей наглядности проведём сравнение EVE-NG с GNS3. Основные отличия приведены в таблице 1.

EVE-NG	GNS3
Для управления используется веб-интерфейс, т.е. не требуется установка каких-либо клиентов.	Необходима установка клиента, т.е. самого GNS3.
Идентичность работы эмулятора: как в ОС Windows, так и в Linux.	Отдельные клиенты для каждой ОС.
Отсутствие ограничений по используемой оперативной памяти при применении QEMU.	Ограничение по ОП в 2 Гб.
Отсутствие ограничений по сетевым подключениям при использовании QEMU.	Не более 16 подключений.
Многопользовательское решение.	Продукт для одного пользователя.

Таблица 1 – Основные отличия EVE-NG от GNS3.

Рассмотрим процесс установки EVE-NG. Стоит отметить, что установку можно произвести как на физический сервер, так и на виртуальную машину (именно этот вариант и будет предложен для разбора).

Образ EVE-NG можно загрузить с официального сайта проекта совершенно бесплатно. Для установки образа потребуется также VMware Player, который также можно загрузить с сайта компании-разработчика. После загрузки образа и установки VMware Player перейдем непосредственно к установке EVE-NG.

Для большего удобства приведем список программного обеспечения, которое может потребоваться вам при установке эмулятора. Это:

- *WinSCP* – графический клиент протоколов SCP и SFTP для Windows. Обеспечивает защищённое копирование файлов между компьютерами и серверами, поддерживающими данные протоколы;
- *PuTTY* – свободно распространяемый клиент для различных протоколов удалённого доступа, таких как SSH, Telnet, rlogin. Позволяет подключиться к удалённому узлу и управлять им;

Помимо этого, вам потребуются образы сетевого оборудования, так как EVE-NG поставляется в виде чистой системы, в которую не входит ни один экземпляр какого-либо сетевого оборудования. Поэтому, просто установив эмулятор на сервер или виртуальную машину, вы сможете всего лишь ознакомиться с его интерфейсом, не более.

1.2.1 Порядок установки:

1. Запустить VMware Player, выбрать пункт «Create a New Virtual Machine». В открывшемся выбрать вариант «I will install the operating system later». В следующем окне необходимо выбрать соответствующие пункты для установки: EVE-NG в виде образа для виртуальной машины основана на Ubuntu 64-bit. В дальнейшем производится выбор папки для установки, ввод названия для виртуальной машины и подбор объёма дискового пространства (20 Гб будет вполне достаточно для обучения)

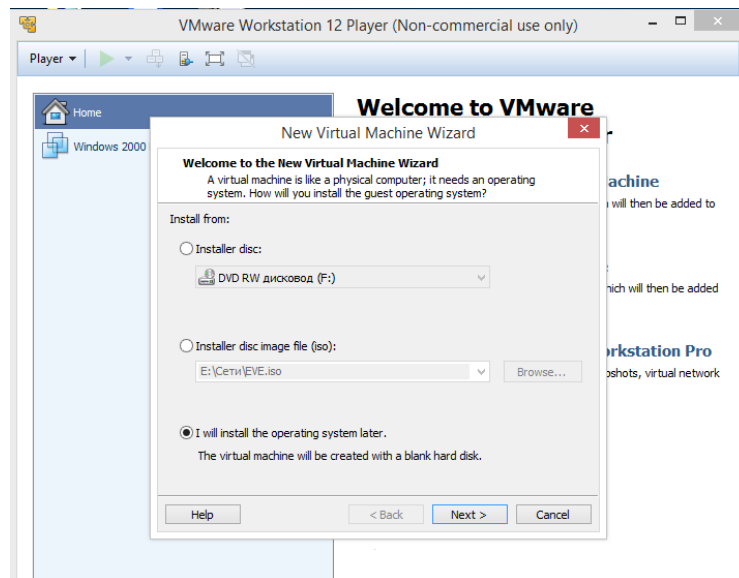


Рисунок 1.2.1.1 – Выбор варианта установки в окне VMware Player

2. Виртуальная машина будет успешно создана с параметрами по умолчанию: 1 Гб ОЗУ, 1 ЦП и 1 сетевой интерфейс. Для решения простейших задач этого вполне достаточно, но для серьезных стендов их необходимо увеличить – всё зависит от сложности стендов и задач, которые предстоит решать.
3. Произведем запуск созданной виртуальной машины. После выбора языка произведем установку EVE VM.

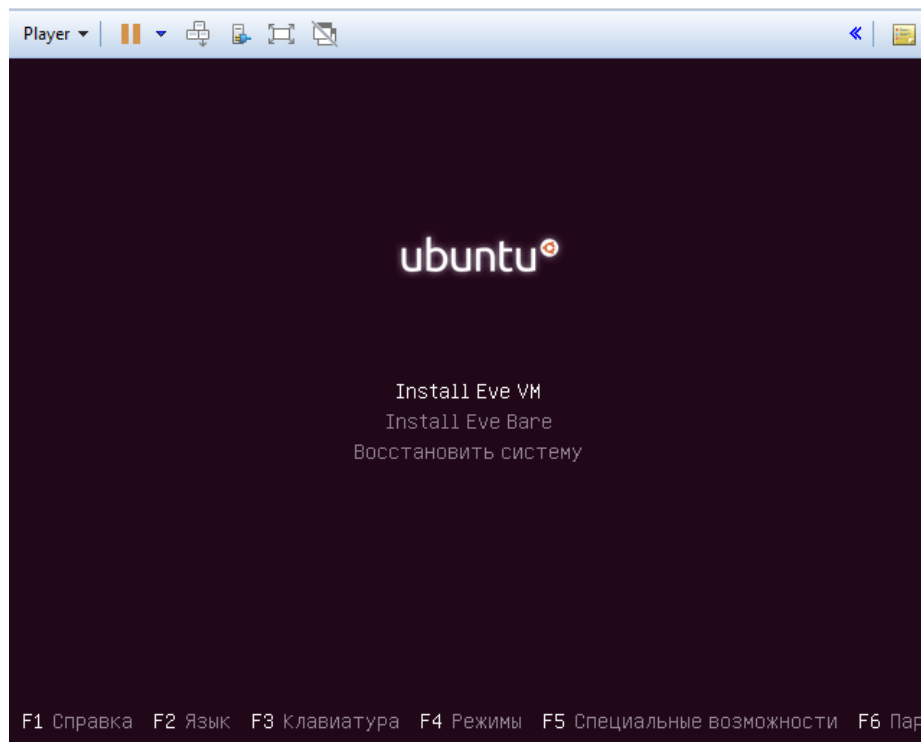


Рисунок 1.2.1.2 – Меню выбора вариантов установки

В процессе установки будут возникать стандартные диалоговые окна: выбор региона, часового пояса, сочетания клавиш для переключения раскладки клавиатуры, ввода сетевого имени и некоторых других свойств. Эти параметры вы можете настроить по своему усмотрению.

4. После успешной установки виртуальная машина перезагрузится и произведет установку и распаковку некоторых дополнительных файлов, необходимых для работы. Как только система будет полностью готова к работе, будет предложено ввести имя пользователя и пароль: root и eve соответственно.

```
Eve-NG (default root password is 'eve')
Use http://192.168.182.128/

WARNING: neither Intel VT-x or AMD-V found

eve-ng login: _
```

Рисунок 1.2.1.3 – Вход в систему

5. Следующими пунктами повторим ввод пароля в появившейся строке, назначим имя машины, имя DNS.

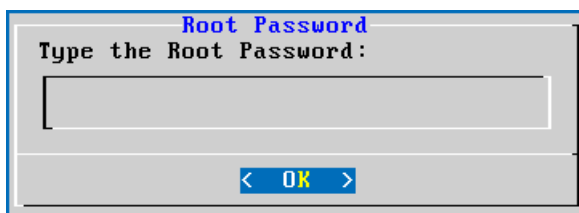


Рисунок 1.2.1.4 – Ввод пароля для пользователя root

6. При настройке типа IP-адреса выберем вариант «Статический». Переключение между пунктами осуществляется при помощи стрелок, выбор пункта – нажатием клавиши «Пробел».

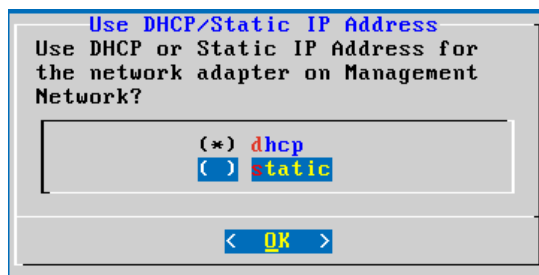


Рисунок 1.2.1.5 – Выбор используемого типа IP-адреса

7. Далее производим назначение IP-адреса, маски (24), сетевого шлюза. Для установки введите IP-адрес, по которому вы будете обращаться к интерфейсу управления. Обратите внимание, что в качестве шлюза по умолчанию всегда используется второй адрес сетевого адаптера виртуальной машины.

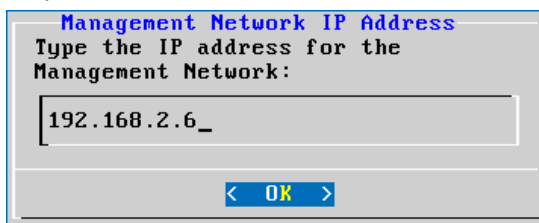


Рисунок 1.2.1.6 – Ввод IP-адреса

8. Будет предложено ввести соответствующую маску, сетевой шлюз. При вводе сетевого шлюза обратите внимание на то, что требуется ввести его для виртуальной машины, а не для компьютера, на который она установлена (обратитесь к меню настроек виртуальной машины для уточнения информации).

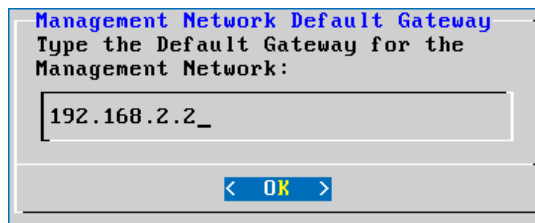


Рисунок 1.2.1.7 – Ввод сетевого шлюза

9. Адрес первичного DNS-сервера, который требуется ввести далее, совпадает с адресом сетевого шлюза по умолчанию. В качестве адреса вторичного DNS-сервера можно ввести 0.0.0.0. Пропускаем ввод данных в следующем окне, и выбираем конфигурацию для подключения виртуальной машины к интернету (в зависимости от вашего подключения).

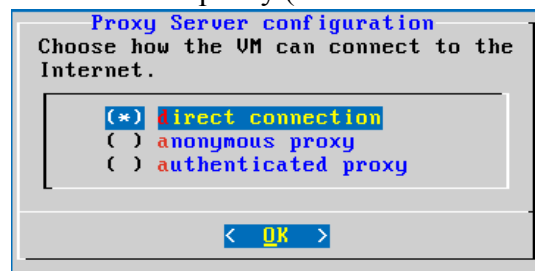


Рисунок 1.2.1.8 – Конфигурация прокси-сервера

На этом установка окончена. Виртуальная машина перезагрузится, необходимо будет повторно ввести учетные данные. Для проверки соединения с интернетом воспользуйтесь командой *ping* (целесообразно также проверить доступность виртуальной машины непосредственно из внешней среды – используйте консоль Windows).

```
Eve-NG (default root password is 'eve')
Use http://192.168.182.130/

WARNING: neither Intel VT-x or AMD-V found

eve-ng login: root
Password:
Welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.14-eve-ng-ukms+ x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

The programs included with the Ubuntu system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by
applicable law.

root@eve-ng:~# ping www.google.com
PING www.google.com (173.194.122.212) 56(84) bytes of data:
64 bytes from www.google.com (173.194.122.212): icmp_seq=1 ttl=128 time=7.55 ms
64 bytes from www.google.com (173.194.122.212): icmp_seq=2 ttl=128 time=7.00 ms
64 bytes from www.google.com (173.194.122.212): icmp_seq=3 ttl=128 time=9.67 ms
^C
--- www.google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2006ms
rtt min/avg/max/mdev = 7.003/8.078/9.679/1.158 ms
root@eve-ng:~#
```

Рисунок 1.2.1.9 – Проверка доступа EVE к сети.

Если вы хотите получить последние официальные обновления, используйте команду *apt-get update*. Таким образом вы только загрузите доступные обновления, для их установки используется другая команда – *apt-get install eve-ng*. Информацию о конфигурации интерфейсов можно просмотреть при помощи команды *ip address*. Как и в любой версии Ubuntu, все настройки хранятся по пути *etc/network/interfaces*. В том случае, если в будущем

вам будет необходимо изменить какие-либо параметры, это достаточно легко осуществить при помощи редактирования данного файла.

1.2.2 Работа с эмулятором

1.2.2.1 Обзор рабочей среды

Если установка и настройка произведены надлежащим образом, вы можете приступать к работе с эмулятором. Для этого введите IP-адрес, который принадлежит виртуальной машине, в строку вашего браузера. Логин – *admin*, пароль – *eve*.

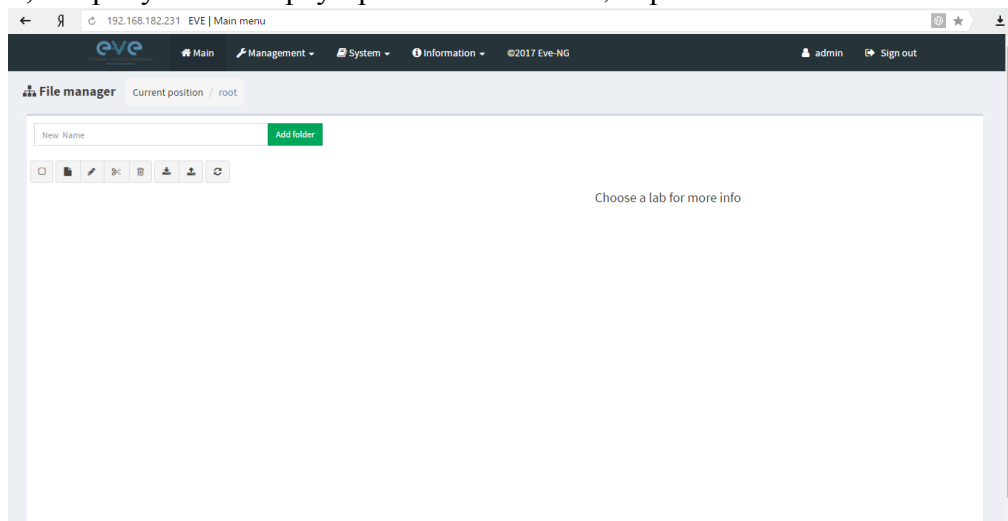


Рисунок 1.2.2.1.1 – Рабочая область эмулятора

Проведем краткий обзор вкладок:

- *Main* – основная рабочая область эмулятора. Представляет собой некоторое подобие файлового менеджера и позволяет создать новую папку, новую лабораторию, удалить элементы, переместить, экспортировать или импортировать лаборатории.
- *Management* – вкладка управления пользователями. С её помощью вы можете не только добавить и удалить пользователя, сменить учётные данные и пароли, роли и права – это полноценное средство администрирования. Вы можете просматривать такую информацию, как время последнего сеанса пользователя, IP-адрес сеанса, текущую папку, текущую лабораторию.
- *System* – вкладка системной информации. Разделена на три пункта: *System status* – отображает информацию по использованию аппаратных ресурсов, таких как объём используемого дискового пространства, загрузку ЦП, оперативной памяти, количество используемых в данный момент экземпляров оборудования по категориям; *System logs* – служит для просмотра файлов журнала, которые содержат различную информацию о произошедших событиях; *Stop all nodes* – приостановка работы всего оборудования, запущенного на данный момент в эмуляторе.
- *Information* – данная вкладка содержит несколько пунктов, носящих сугубо информационный характер. Среди них присутствует ссылка на форум и канал на YouTube, что может быть полезно при возникновении сложных вопросов по работе эмулятора.

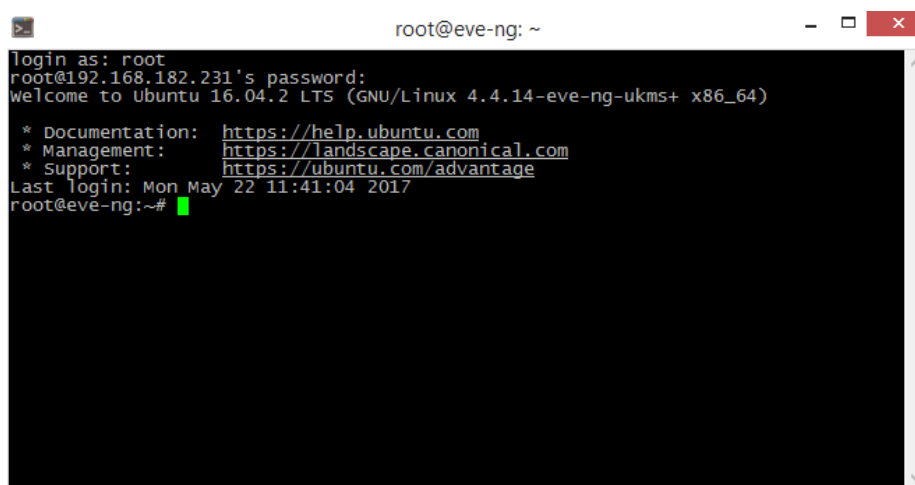
1.2.2.2 Установка образов оборудования

Как уже было упомянуто выше, EVE-NG поступает без образов сетевого оборудования. Это означает, что для работы с эмулятором необходимо загрузить и установить образы оборудования. При работе с файловой системой виртуальной машины предлагается

использовать файл-менеджер WinSCP, который можно бесплатно загрузить с официальной страницы продукта. Для подключения к виртуальной машине используйте её IP-адрес и стандартные данные для входа.

Образы оборудования в формате **.bin* при помощи WinSCP необходимо скопировать по пути *opt/unetlab/addons/iol/bin*. Работа с WinSCP аналогична работе с любым файл-менеджером, но больше всего похожа на использование TotalCommander: в левой части окна вы можете видеть файловую систему вашего компьютера, в правой – виртуальной машины, к которой вы подключены. Копирование файлов осуществляется простым перетаскиванием при помощи мыши из одной области в другую.

После этого подключимся к виртуальной машине при помощи PuTTY (используя те же данные, что и ранее).

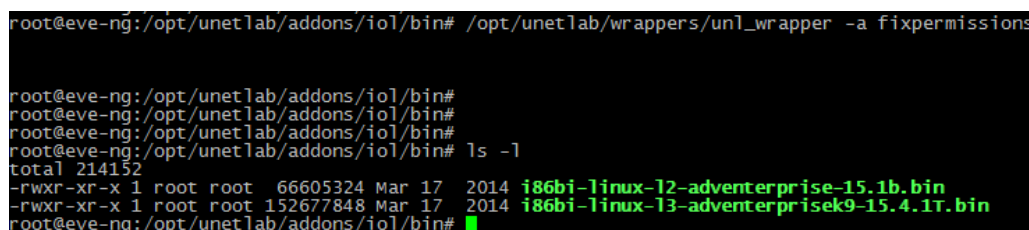


```
root@eve-ng: ~
login as: root
root@192.168.182.231's password:
welcome to Ubuntu 16.04.2 LTS (GNU/Linux 4.4.14-eve-ng-ukms+ x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage
Last login: Mon May 22 11:41:04 2017
root@eve-ng:~#
```

Рисунок 1.2.2.2.1 – Терминал EVE-NG при подключении через PuTTY

Используя стандартные команды, перейдём в каталог *opt/unetlab/addons/iol/bin* и убедимся в том, что образы оборудования успешно скопированы. При помощи команды */opt/unetlab/wrappers/unl_wrapper -a fixpermissions* настроим права доступа к файлам образов. В конечном итоге у вас должно получиться следующее:



```
root@eve-ng:/opt/unetlab/addons/iol/bin# /opt/unetlab/wrappers/unl_wrapper -a fixpermissions
root@eve-ng:/opt/unetlab/addons/iol/bin#
root@eve-ng:/opt/unetlab/addons/iol/bin#
root@eve-ng:/opt/unetlab/addons/iol/bin# ls -l
total 214152
-rwxr-xr-x 1 root root 66605324 Mar 17 2014 i86bi-linux-12-adventureprise-15.1b.bin
-rwxr-xr-x 1 root root 152677848 Mar 17 2014 i86bi-linux-13-adventureprise9-15.4.1t.bin
root@eve-ng:/opt/unetlab/addons/iol/bin#
```

Рисунок 1.2.2.2.2 – Полная информация о файлах образов с настроенными правами доступа

В каталоге с образами также должен находиться файл *iourc*, который содержит информацию о лицензии (лицензионный ключ), без которого использование загруженных образов не будет возможно. Его также необходимо перенести в папку при помощи WinSCP. Чтобы убедиться в том, что установленные образы будут работать корректно, примените следующие команды:

- *Touch NETMAP;*

- *LD_LIBRARY_PATH=/opt/unetlab/addons/iol/lib /opt/unetlab/addons/iol/bin/*полное имя образа устройства* ID устройства (может быть любым числом, к примеру – 100)*

После применения вышеописанных команд в консоли терминала должна появиться информация о лицензии, наподобие той, что представлена на рисунке 13.

```
Restricted Rights Legend

Use, duplication, or disclosure by the Government is
subject to restrictions as set forth in subparagraph
(c) of the Commercial Computer Software - Restricted
Rights clause at FAR sec. 52.227-19 and subparagraph
(c) (1) (ii) of the Rights in Technical Data and Computer
Software clause at DFARS sec. 252.227-7013.

-----
cisco systems, Inc.
170 West Tasman Drive
San Jose, California 95134-1706

Cisco IOS Software, Linux Software (I86BI_LINUX-ADVENTERPRISEK9-M), Version 15.4
(1)T, DEVELOPMENT TEST SOFTWARE
Technical Support: http://www.cisco.com/techsupport
Copyright (c) 1986-2013 by Cisco Systems, Inc.
Compiled Sat 23-Nov-13 03:28 by prod_rel_team

This product contains cryptographic features and is subject to United
States and local country laws governing import, export, transfer and
use. Delivery of Cisco cryptographic products does not imply
third-party authority to import, export, distribute or use encryption.
Importers, exporters, distributors and users are responsible for
compliance with U.S. and local country laws. By using this product you
agree to comply with applicable laws and regulations. If you are unable
to comply with U.S. and local laws, return this product immediately.

A summary of U.S. laws governing Cisco cryptographic products may be found at:
http://www.cisco.com/wwl/export/crypto/tool/stqrg.html

If you require further assistance please contact us by sending email to
export@cisco.com.
```

Рисунок 1.2.2.2.3 – Информация о лицензии

1.2.3 Создание лабораторного стенда

1.2.3.1 Создание топологии

Теперь можно приступить к созданию лабораторного стенда. Для этого откроем веб-интерфейс эмулятора и при помощи кнопки «Add new lab» создадим новую работу, заполнив все необходимые поля формы, изображенной на рисунке 14.

Рисунок 1.2.3.1.1 – Форма создания новой работы

После нажатия на кнопку «Save» произойдет загрузка страницы редактирования топологии, где вы можете добавить устройства из установленных образов (они будут подсвечены зелёным в списке оборудования). Добавление оборудования осуществляется в последовательности «Add an object – Node».

Рассмотрим процесс добавления устройства L2 – коммутатора. В открывшемся окне производится выбор групп оборудования (*Template*), образа оборудования (*Image*), количества добавляемых экземпляров (*Number of nodes to add*), ввод названия, выбор отображаемой иконки, количества групп портов (по 4 Ethernet и Serial порта в группе), стартовой конфигурации, задержки.

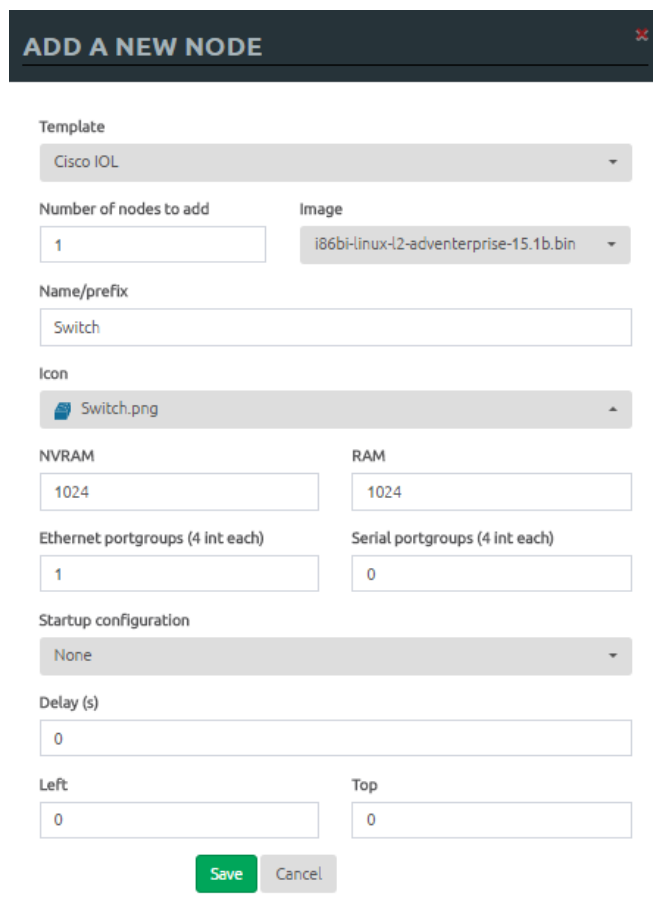


Рисунок 1.2.3.1.2 – Добавление нового экземпляра оборудования

Для соединения экземпляров оборудования между собой необходимо подвести курсор мыши к выбранному экземпляру оборудования, после чего рядом с ним появится пиктограмма, изображающая электрическую вилку (см. Рисунок 16). После нажатия на эту пиктограмму вам предоставляется выбор интерфейсов для подсоединения, остается только указать на то устройство, к которому вы хотите подключиться.

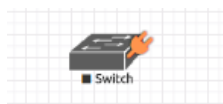


Рисунок 1.2.3.1.3 – Создание соединения

На рисунке 17 показано окно настройки соединения. В данном окне вы можете произвести выбор интерфейсов устройств, между которыми будет создано соединение.

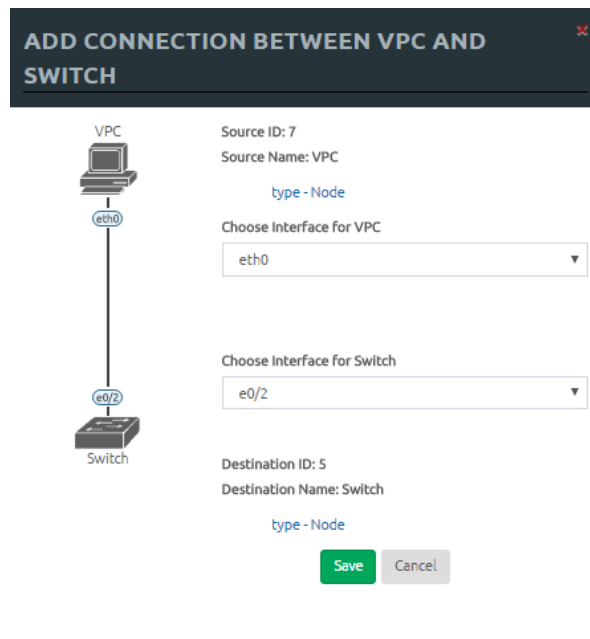


Рисунок 1.2.3.1.4 – Форма создания соединения

Добавляя оборудование и соединяя его между собой, создадим топологию, изображенную на рисунке 18. В данной топологии применяются только L2 устройства Cisco и VPC – Virtual PC.

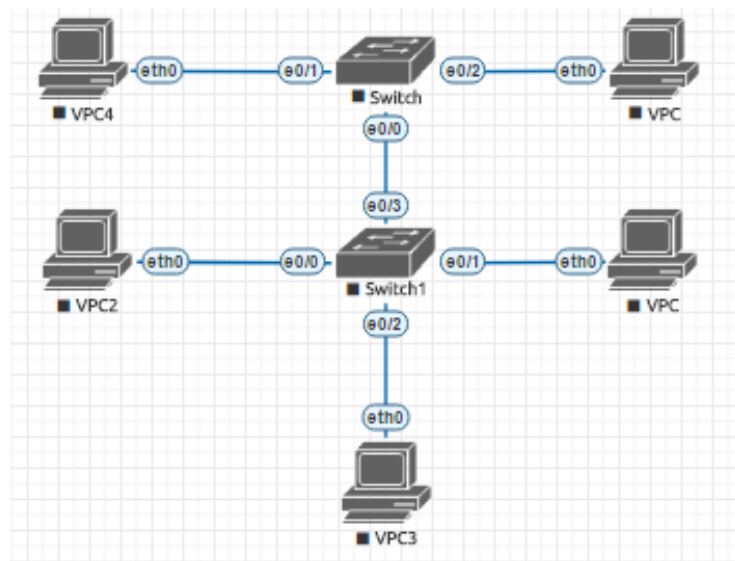


Рисунок 1.2.3.1.5 – Реализуемая топология

1.2.3.2 Настройка VPC

После того, как топология сети создана, необходимо произвести настройку оборудования. Следует назначить всем компьютерам в сети IP-адреса и маски таким образом, чтобы они находились в одной подсети. На коммутаторах необходимо отключить неиспользуемые порты.

Для успешного подключения к оборудованию и его настройки необходимо включить экземпляры оборудования. Это осуществляется из контекстного меню, появляющегося при нажатии на выбранный экземпляр правой кнопкой мыши – пункт *Start*.

Настройка оборудования производится из консоли устройств. Доступ к ней можно получить при помощи стандартного Telnet-клиента Windows, SecureCRT или PuTTY. По умолчанию, при щелчке левой кнопкой мыши на экземпляре оборудования, будет предложено открыть окно стандартного Telnet-клиента Windows, но ввиду того, что в современных версиях системы он не предустановлен, удобнее воспользоваться PuTTY. Для этого необходимо открыть программу, после чего ввести требуемые для подключения данные – IP-адрес, порт и выбрать тип соединения Telnet. Удобнее открывать соединение в новом окне: это позволит избежать повторного запуска PuTTY и даст возможность одновременно наблюдать за работой и конфигурировать несколько экземпляров оборудования. Пример настройки показан на рисунке 19.

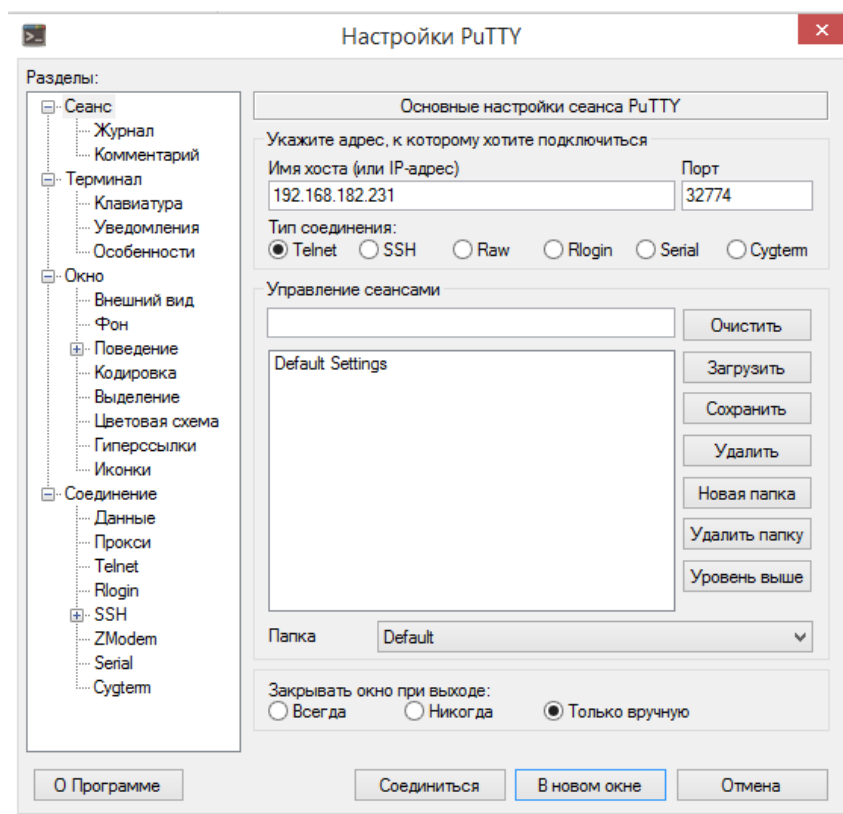


Рисунок 1.2.3.2.1 – Окно создания подключения к экземпляру оборудования в PuTTY

Подключимся к одному из VPC. Открывшееся окно терминала показано на рисунке 20.

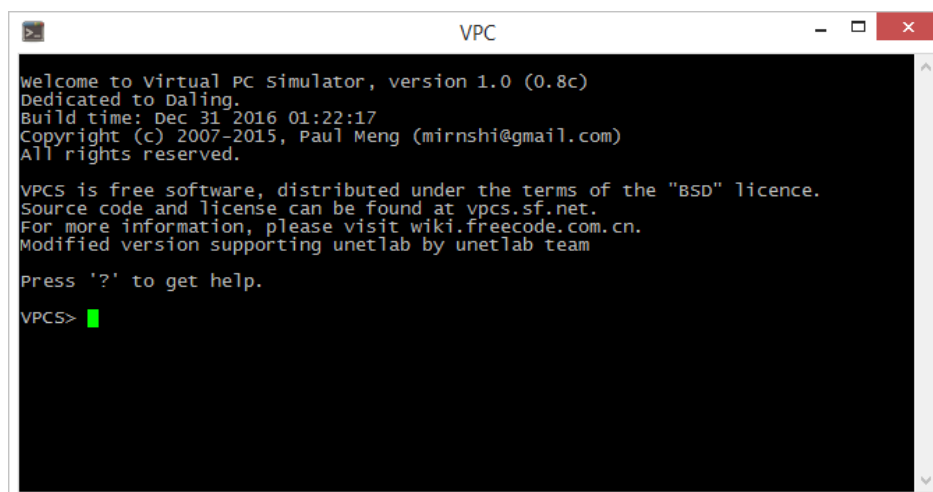
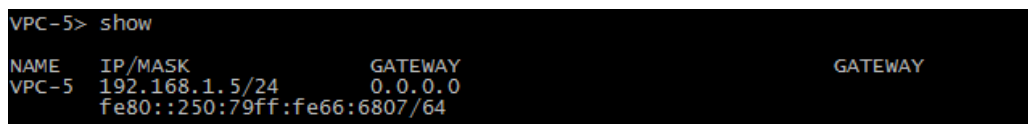


Рисунок 1.2.3.2.2 – Окно настройки устройства VPC.

Получить подсказки по командам, используемым для настройки и управления устройством поможет команда `?`. Произведем настройку всех устройств VPC. Для просмотра результатов воспользуемся командой `show` – результат её выполнения показан на рисунке 21.



```
VPC-5> show
```

NAME	IP/MASK	GATEWAY	GATEWAY
VPC-5	192.168.1.5/24	0.0.0.0	
	fe80::250:79ff:fe66:6807/64		

Рисунок 1.2.3.2.3 – Основная информация о VPC-5.

Мы рассмотрели работу с оборудованием образца VPC, которые поставляются по открытой лицензии и доступны в эмуляторе изначально. После того, как они полностью сконфигурированы, перейдём к настройке коммутаторов Cisco.

1.2.3.4 Настройка коммутаторов Cisco

Оборудование Cisco использует в качестве операционной системы Cisco IOS – Internetwork Operating System, на которую возлагаются функции коммутации, маршрутизации, передачи данных. Система имеет индивидуальный интерфейс командной строки CLI – Command Line Interface, включающий впечатляющий и достаточный для конфигурирования оборудования набор команд, о которых вы можете узнать более подробно из специализированной литературы.

Подключение к коммутатору осуществляется таким же образом, как и к VPC, благодаря чему вам не составит труда его инициировать. В открывшемся при успешном подключении окне терминала вы сможете наблюдать обыкновенный интерфейс CLI (разве что только на черном фоне). Один из коммутаторов уже был заранее сконфигурирован, это сразу заметно по наличию баннера *message-of-the-day* на рисунке 22.

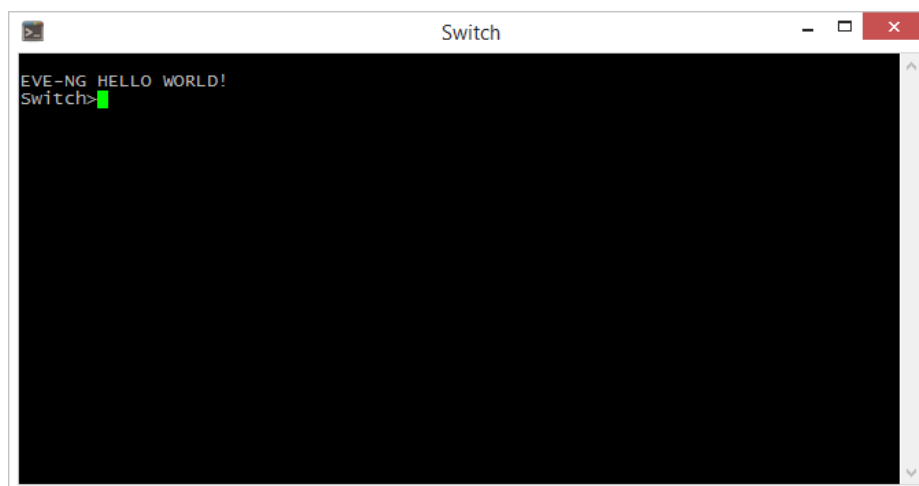


Рисунок 1.2.3.4.1 – Окно CLI коммутатора Switch.

Все настройки оборудования проводятся ровно таким же образом, так же, как и в Cisco Packet Tracer.

После окончательного конфигурирования оборудования необходимо проверить доступность каждого из узлов при помощи команды `ping`.

1.3 Практическая часть

Данная лабораторная работа выполняется в эмуляторе EVE-NG, первоначально не установленном в качестве виртуальной машины. Все необходимые действия приведены в порядке выполнения.

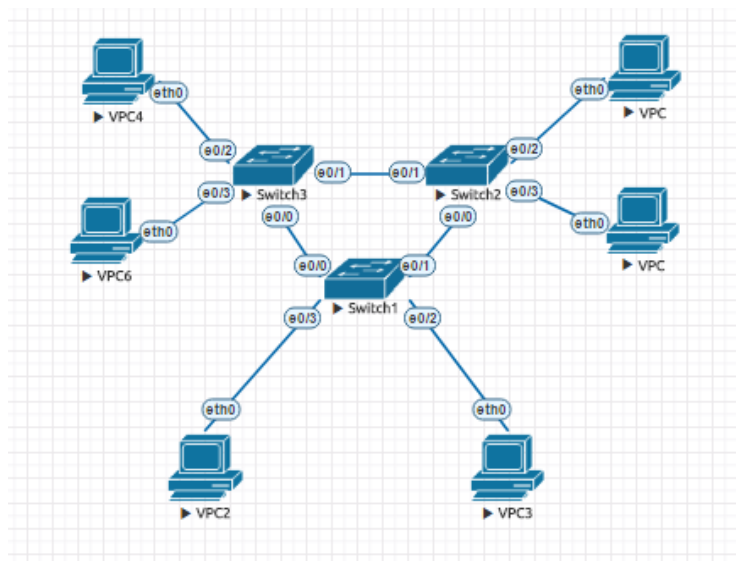


Рисунок 1.3.1 – Используемая топология

1.3.1 Порядок выполнения работы

- 1.3.1.1 Выполнить установку EVE-NG на виртуальную машину.
- 1.3.1.2 Установить образы оборудования.
- 1.3.1.3 Построить топологию в соответствии с приведённой в качестве задания.
- 1.3.1.4 Произвести конфигурирование оборудования. Для каждого VPC задать name, IP-адрес, маску. Для каждого коммутатора задать баннер message-of-the-day, установить пароли для console, присвоить hostname, отключить неиспользуемые порты и перевести используемые в требуемый режим.

Лабораторная работа №2. VLAN

1.1. Теоретические сведения

VLAN – от английского *Virtual Local Area Network* – логическая (виртуальная) локальная сеть. Она представляет собой группу устройств, которые могут взаимодействовать друг с другом на канальном уровне, хотя при этом могут быть подключены к различным коммутаторам. Иными словами, устройства взаимодействуют друг с другом таким образом, как если бы они были подключены к широковещательному домену. В то же время, устройства, находящиеся в различных VLAN и при этом подключенные к одному коммутатору, не могут возможности взаимодействия друг с другом на канальном уровне. Связь между ними возможна на сетевом и более высоких уровнях.

VLAN – один из наиболее распространенных механизмов логической сегментации сетей, защиты сетей от *ARP-spoofing* 'a (одна из техник атаки, применяемая в сетях, использующих протокол ARP, позволяет перехватывать трафик между узлами, взаимодействующими в пределах одного широковещательного домена). Кроме того, VLAN используют для сокращения объёма широковещательного трафика в сети.

1.2. Назначение VLAN

С использованием VLAN открывается следующий ряд возможностей:

- **сокращение объёма широковещательного трафика в сети** – в связи с тем, что VLAN представляет собой отдельный широковещательный домен, при его создании на двух различных коммутаторах, их порты в совокупности будут образовывать один широковещательный домен. В то же время, при помощи VLAN возможно разбить порты одного коммутатора на несколько широковещательных доменов;
- **повышение степени безопасности сети** – при разбиении сети на VLAN появляется возможность применения политик безопасности ко всей подсети сразу, а не к устройствам по отдельности. Помимо этого, переход от одной VLAN к другой предполагает использование L3-устройства, на котором могут применяться политики, которые разрешают или запрещают доступ;
- **гибкое разделение устройств по подсетям** – в силу того, что VLAN не привязана к местоположению устройств, находящиеся на дальнем расстоянии друг от друга устройства, подключенные к различным коммутаторам, могут находиться внутри одной VLAN.

1.3. Тегирование трафика

Тегирование – процесс добавления метки (тега) VLAN к кадру. Как правило, конечные устройства (к примеру, компьютеры пользователей) не тегировать трафик. Эта задача возлагается на коммутаторы, работающие в сети. Более того, конечные устройства «не представляют», в каком VLAN они находятся. Таким образом, трафик в разных VLAN особо ничем не различается.

Однако, если через порт коммутатора может проходить трафик нескольких VLAN одновременно, его необходимо каким-то образом дифференцировать. Именно по этой причине каждый кадр должен быть определённым образом помечен, т.е. иметь информацию о принадлежности к VLAN. Наибольшее распространение получила технология, описанная в спецификации IEEE 802.1Q, но существуют и другие протоколы, к примеру, ISL (Cisco Systems).

- *ISL – Inter Switch Link, протокол межкоммутационного канала* – проприетарный протокол компании Cisco Systems, предназначенный для передачи информации о принадлежности трафика к VLAN. Данный протокол инкапсулирует исходный кадр, добавляя заголовок, содержащий информацию о принадлежности к VLAN. ISL был разработан до стандарта IEEE 802.1Q и какое-то время поддерживался на сетевом оборудовании Cisco наряду с 802.1Q, но в настоящее время протокол является устаревшим.
- *IEEE 802.1Q* – открытый стандарт, описывающий процедуру тегирования трафика для передачи информации о принадлежности его к VLAN. Данный стандарт, в отличие от ISL, не изменяет заголовки кадра, благодаря чему устройства, его не поддерживающие, могут передавать трафик без учёта его принадлежности к VLAN. 802.1Q помещает внутрь кадра тег, который передаёт информацию о принадлежности трафика к VLAN. Размер тега – 4 байта, состоит он из следующих полей (рисунок 1):
 - **Tag Protocol Identifier (TPID)** — Идентификатор протокола тегирования. Размер поля — 16 бит. Указывает, какой протокол используется для тегирования. Для 802.1q используется значение 0x8100.

- **Tag Control Information (TCI)** - поле, инкапсулирующее в себе поля приоритета, канонического формата и идентификатора VLAN:
 - **Priority** — приоритет. Размер поля — 3 бита. Используется стандартом IEEE 802.1p для задания приоритета передаваемого трафика.
 - **Canonical Format Indicator** — индикатор канонического формата. Размер поля — 1 бит. Указывает на формат MAC-адреса. 0 — канонический (кадр Ethernet), 1 — не канонический (кадр Token Ring, FDDI).
 - **VLAN Identifier** — идентификатор VLAN. Размер поля — 12 бит. Указывает, какому VLAN принадлежит фрейм. Диапазон возможных значений VID от 0 до 4094.

В стандарте 802.1Q существует понятие *Native VLAN*. По умолчанию это VLAN 1. Трафик, передающийся в этом VLAN, не тегировается.

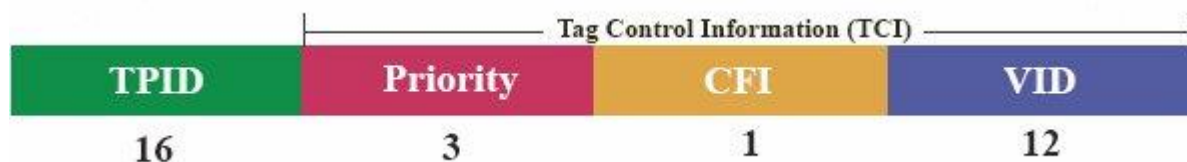


Рисунок 1 – Формат тега 802.1Q

- *802.1ad (Stacked VLANs, Q-in-Q)* – открытый стандарт, описывающий двойной тег. Его основное отличие от 802.1Q заключается в наличии двух VLAN: внешнего и внутреннего, что позволяет разбить сеть не на 4095 VLAN, а на 4095². Кроме того, наличие двух тегов позволяет организовывать более гибкие и сложные сети. Например, использование данного стандарта позволяет организовать соединение на сетевом уровне для двух разных устройств, расположенных в разных городах, но посылающих трафик с одинаковыми тегами.

1.4. Виды портов

Порты коммутаторов, поддерживающие VLAN, можно разделить на две категории:

- *Тегированные (trunk) порты* – предназначены для организации передачи через один порт данных, принадлежащих к различным VLAN, и, соответственно, получения трафика различных VLAN на один порт.
- *Нетегированные (access) порты* – предназначены для тегирования нетегированного трафика, которое происходит при его прохождении через данный порт.

Существует два подхода к назначению порта коммутатора в определённый VLAN:

- *Статическое назначение* – принадлежность порта к VLAN задаётся администратором при настройке оборудования.
- *Динамическое назначение* – принадлежность порта к VLAN определяется в процессе работы коммутатора при помощи определённых процедур, описанных в специальных стандартах, например, в 802.1X. Согласно протоколу 802.1X доступ к сети получают только клиенты, прошедшие аутентификацию, если аутентификация

не была пройдена, доступ с соответствующего порта будет запрещен. По результатам аутентификации порт коммутатора размещается в том или ином VLAN.

Trunk-порты имеют несколько различных режимов работы:

- *auto* – порт находится в автоматическом режиме и будет переведён в режим *trunk*, только в том случае, если порт на другой стороне находится в режиме *desirable* или *on*.
- *desirable* – порт находится в режиме готовности к переходу в *trunk*, происходит передача кадров DTP соседнему порту, запрашивая порт перейти в состояние *trunk*. Режим *trunk* будет установлен в том случае, если соседний порт находится в режимах *on*, *desirable*, *auto*.
- *trunk* – порт постоянно находится в режиме *trunk*, даже если соседний порт не поддерживает данный режим.
- *nonegotiate* – порт готов перейти в режим *trunk*, но при этом не передает кадры DTP на другом конце. Данный режим используется для предотвращения конфликтов с оборудованием других производителей.

1.5. Протокол VTP

VLAN Trunking Protocol – проприетарный протокол компании Cisco Systems, предназначенный для создания, удаления и переименования VLAN на сетевых устройствах. Данный протокол имеет свободный аналог – *GVRP*.

Сообщения VTP делятся на 4 группы

- *Summary advertisements* – рассылается каждые 5 минут сервером VTP, содержит в себе сведения о текущей версии конфигурации, имени домена и другой детальной информации о конфигурации.
- *Subset advertisements* – рассылается немедленно при изменении следующих параметров конфигурации: создании, удалении, переименовании, приостановки и активации VLAN.
- *Advertisement requests* – осуществляется в случае изменения имени домена, обновлении конфигурации, перезагрузки коммутатора.
- *VTP Join Messages* – предназначены для отсечения (*pruning*).

На коммутаторах протокол VTP может работать в следующих режимах:

Server – режим по умолчанию:

- Можно создавать, изменять и удалять VLAN из CLI.
- Происходит генерирование объявлений VTP и передача объявлений от других коммутаторов.
- Может обновлять свою базу VLAN при получении информации не только от других VTP-серверов, но и от других VTP-клиентов в одном домене, с более высоким номером ревизии.
- Сохраняет информацию о настройках VLAN в файле *vlan.dat*.

Client:

- Нет возможности создания, изменения и удаления VLAN из CLI.
- Происходит только передача объявлений от других коммутаторов.
- Синхронизируется база данных VLAN при получении информации VTP.
- Сохраняет информацию о настройках VLAN в файле *vlan.dat*.

Transparent:

- Можно создавать, изменять и удалять VLAN из CLI, но только для локального коммутатора.
- Не генерирует объявления VTP.
- Передает объявления от других коммутаторов.
- Не обновляет свою базу данных VLAN при получении информации по VTP.
- Сохраняет информацию о настройках VLAN в NVRAM.

Off – данный режим работы был добавлен в третьей версии протокола, аналогичен режиму Transparent, но, в отличие от него, не передает объявления VTP. В таблице ниже показана общая информация по диапазонам VLAN.

Таблица 1 – Диапазоны VLAN

VLANs	Диапазон	Использование	Передается VTP
0, 4095	Reserved	Только для системного использования.	--
1	Normal	VLAN по умолчанию. Можно использовать, но нельзя удалить.	Да
2-1001	Normal	Для VLANов Ethernet. Можно создавать, удалять и использовать.	Да
1002-1005	Normal	Для FDDI и Token Ring. Нельзя удалить.	Да
1006-4094	Extended	Только для VLANов Ethernet.	Только в версии 3

1.6. Native VLAN

Native VLAN - это понятие в стандарте 802.1Q, которое обозначает VLAN на коммутаторе, где все кадры идут без тега, т.е. трафик передается нетегированным. По умолчанию это VLAN 1. В некоторых моделях коммутаторов, например, Cisco, это можно изменить, указав другой VLAN как native. Если коммутатор получает нетегированные кадры на порту *trunk*, он автоматически причисляет их к Native VLAN. И точно так же кадры, генерируемые с нераспределенных портов, при попадании в trunk причисляются к Native VLAN.

Трафик, который принадлежит другим VLAN, тегируется с указанием соответствующего VLAN ID внутри тега.

Пример настройки Native VLAN на коммутаторах Cisco

Настройка VLAN 2 как native:

```
sw1(config)# interface f0/10
sw1(config-if)# switchport trunk native vlan 2
```

Теперь весь трафик, принадлежащий VLAN 2 будет передаваться через trunk нетегированным, а весь пришедший на trunk нетегированный трафик будет промаркирован как принадлежащий VLAN 2 (по умолчанию VLAN 1).

Проверить какой VLAN настроен как Native:

```
Switch#sh int e0/1 trunk
```

Port	Mode	Encapsulation	Status	Native vlan
Et0/10	on	802.1q	trunking	2

Port	Vlans allowed on trunk
Et0/10	2,101-102

Port	Vlans allowed and active in management domain
Et0/10	2,101-102

Port	Vlans in spanning tree forwarding state and not pruned
Et0/10	2,101-102

Из соображений безопасности (например, для защиты от *VLAN Hopping*) рекомендуется в trunk выполнять тегирование даже для Native VLAN. Включить тегирование фреймов для Native VLAN глобально можно с помощью команды `vlan dot1q tag native`, посмотреть текущий статус тегирования можно используя команду `show vlan dot1q tag native`.

```
Switch(config)#no vlan dot1q tag native
Switch#sho vlan dot1q tag native
dot1q native vlan tagging is disabled
```

1.7. Конфигурирование с настройкой маршрутизации между VLAN

В этом разделе приведены конфигурационные файлы коммутаторов для изображенной схемы. На коммутаторе sw3 настроена маршрутизация между VLAN, поэтому в данной схеме hosts могут общаться как в пределах одного VLAN, так и между различными VLAN.

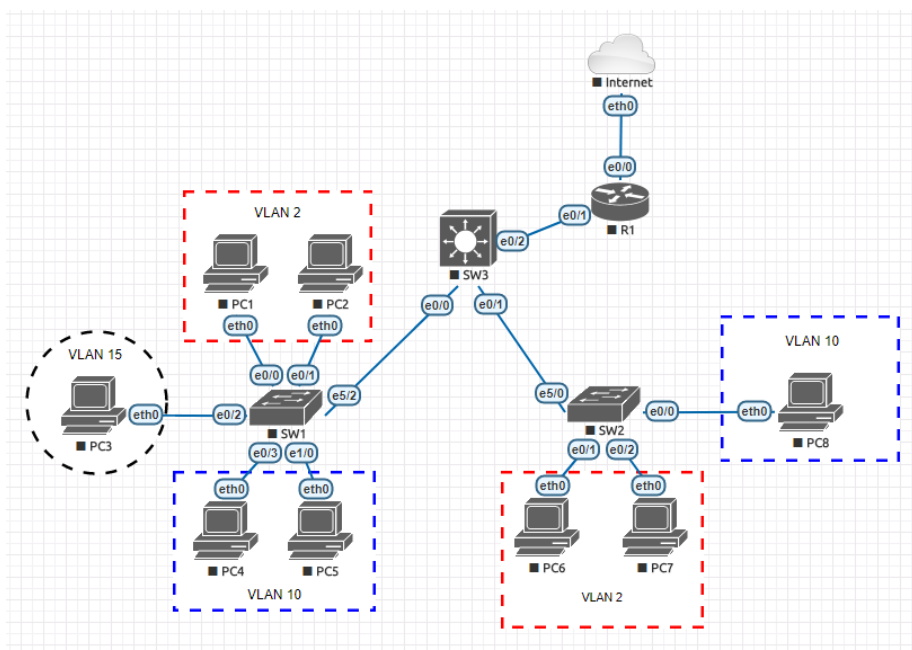


Рисунок 2 – Используемая топология

Здесь рассмотрим конфигурации коммутаторов для изображенной на рис.2 схемы. На коммутаторе SW3 настроена маршрутизация между VLAN, поэтому в рамках данной схемы устройства могут обмениваться информацией не только в пределах одной VLAN, но и между различными VLAN.

Таким образом, устройства, подключенные к коммутатору SW1 и находящиеся в рамках VLAN 2, могут обмениваться информацией между собой и с устройствами в VLAN 2, подключенными к коммутатору SW2. Но кроме того, они также могут взаимодействовать с устройствами в других VLAN на коммутаторах SW1 и SW2.

Конфигурация SW1:

```
interface Eth0/0
  switchport mode access
  switchport access vlan 2
interface Eth0/1
  switchport mode access
  switchport access vlan 2
interface Eth0/2
  switchport mode access
  switchport access vlan 15
interface Eth0/3
  switchport mode access
  switchport access vlan 10
interface Eth1/0
  switchport mode access
  switchport access vlan 10
interface Eth5/2
  switchport trunk encapsulation dot1q
  switchport mode trunk
  switchport trunk allowed vlan 1,2,10,15
```

Конфигурация SW2:

```
interface Eth0/0
  switchport mode access
  switchport access vlan 10
interface Eth0/1
  switchport mode access
  switchport access vlan 2
interface Eth0/2
  switchport mode access
  switchport access vlan 2
interface Eth5/0
  switchport trunk encapsulation dot1q
  switchport mode trunk
  switchport trunk allowed vlan 1,2,10
```

Конфигурация SW3:

```
ip routing
vlan 2,10,15
interface Eth0/0
  switchport mode trunk
  switchport trunk allowed vlan 1,2,10,15
interface Eth0/1
  switchport mode trunk
  switchport trunk allowed vlan 1,2,10
interface Eth0/2
  no switchport
  ip address 192.168.1.2 255.255.255.0
interface Vlan2
```

```

ip address 10.0.2.1 255.255.255.0
interface Vlan10
ip address 10.0.10.1 255.255.255.0
interface Vlan15
ip address 10.0.15.1 255.255.255.0
ip route 0.0.0.0 0.0.0.0 192.168.1.1

```

Конфигурирование коммутаторов завершено, а проверить его корректность можно при помощи команды *show vlan brief* на любом из коммутаторов. Таким образом, была создана топология, в которой реализован механизм маршрутизации между VLAN.

2. Практическая часть

1. Построить приведённую на рис.3 топологию сети в эмуляторе EVE-NG.
2. Присвоить устройствам IP-адреса.
3. Создать требуемые VLAN.
4. Настроить коммутаторы для реализации механизма маршрутизации между VLAN (на рис.3 зелеными областями выделены те VLAN, между которыми должна быть организована маршрутизация).
5. Проверить результаты настройки оборудования, убедиться в корректности конфигурации.

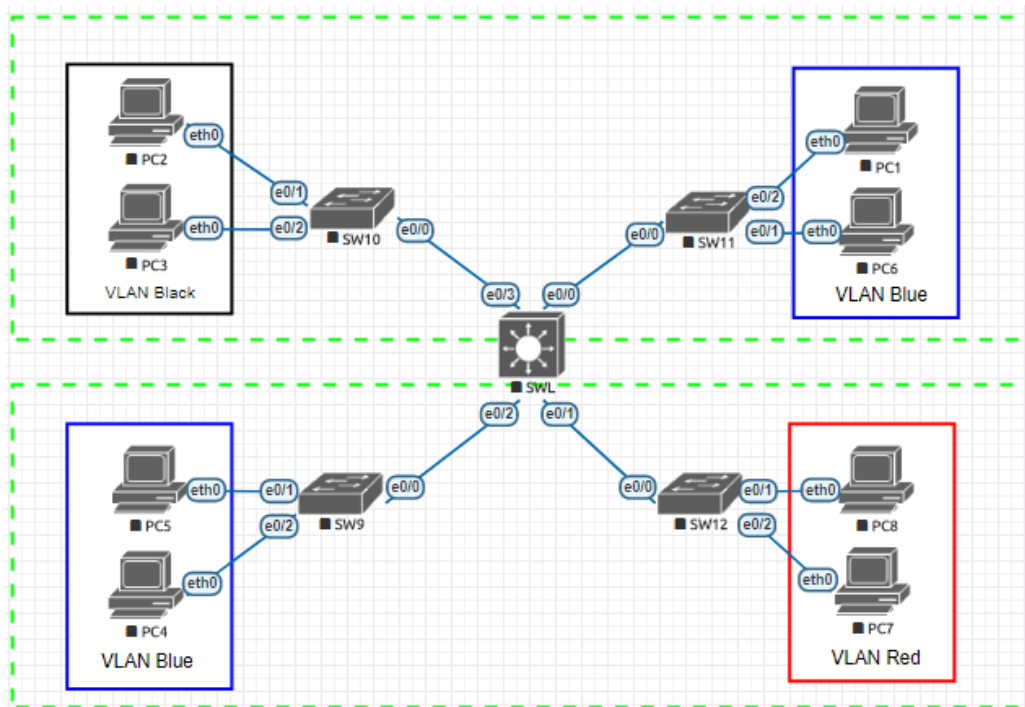


Рисунок 3 – Используемая топология

Лабораторная работа №3. NAT

1. Теоретическая часть

1.1. Технология NAT

NAT (от английского *Network Address Translation*) – механизм преобразования сетевых адресов. Этот механизм в TCP/IP позволяет преобразовывать IP-адреса транзитных пакетов. Преобразование адреса при помощи метода NAT может производиться практически любым маршрутизирующим устройством – маршрутизатором, сервером доступа, межсетевым экраном. Наибольшей популярностью пользуется SNAT, механизм которого состоит в замене адреса источника при прохождении пакета в одну сторону и обратной замене адреса назначения в обратном пакете.

При приёме пакета от локального компьютера, маршрутизатор смотрит на IP-адрес назначения: если это локальный адрес, то пакет пересылается другому локальному компьютеру, в противном случае маршрутизатор транслирует обратный IP-адрес пакета на свой внешний IP-адрес и меняет номер порта. Комбинацию, необходимую для обратной подмены, маршрутизатор сохраняет у себя во временной таблице. По прошествии некоторого количества времени данная запись стирается за сроком давности.

Технология NAT имеет ряд преимуществ:

- Позволяет сэкономить IP-адреса (только в случае использования NAT в режиме PAT), транслируя несколько внутренних IP-адресов в один внешний публичный IP-адрес (или в несколько, но меньшим количеством, чем внутренних). По такому принципу построено большинство сетей в мире: на небольшой район домашней сети местного провайдера или на офис выделяется 1 публичный (внешний) IP-адрес, за которым работают и получают доступ интерфейсы с приватными (внутренними) IP-адресами.
- Позволяет предотвратить или ограничить обращение снаружи ко внутренним хостам, оставляя возможность обращения изнутри наружу. При инициации соединения изнутри сети создаётся трансляция. Ответные пакеты, поступающие снаружи, соответствуют созданной трансляции и поэтому пропускаются. Если для пакетов, поступающих снаружи, соответствующей трансляции не существует (а она может быть созданной при инициации соединения или статической), они не пропускаются.
- Позволяет скрыть определённые внутренние сервисы внутренних хостов/серверов. По сути, выполняется та же указанная выше трансляция на определённый порт, но возможно подменить внутренний порт официально зарегистрированной службы (например, 80-й порт TCP (HTTP-сервер) на внешний 54055-й). Тем самым, снаружи, на внешнем IP-адресе после трансляции адресов на сайт (или форум) для осведомлённых посетителей можно будет попасть по адресу <http://example.org:54055>, но на внутреннем сервере, находящемся за NAT, он будет работать на обычном 80-м порту.

Разумеется, наряду с преимуществами технология имеет некоторые недостатки:

- *Старые протоколы.* Протоколы, разработанные до массового внедрения NAT, не в состоянии работать, если на пути между взаимодействующими хостами есть трансляция адресов. Некоторые межсетевые экраны, осуществляющие трансляцию IP-адресов, могут исправить этот недостаток, соответствующим образом заменяя IP-

адреса не только в заголовках IP, но и на более высоких уровнях (например, в командах протокола FTP).

- *Идентификация пользователей.* Из-за трансляции адресов «много в один» появляются дополнительные сложности с идентификацией пользователей и необходимость хранить полные логи трансляций.
- *Иллюзия DoS-атаки.* Если NAT используется для подключения многих пользователей к одному и тому же сервису, это может вызвать иллюзию DoS-атаки на сервис (множество успешных и неуспешных попыток. Частичным решением проблемы является использование пула адресов (группы адресов), для которых осуществляется трансляция.
- *Пиринговые сети.* В NAT-устройствах, не поддерживающих технологию Universal Plug & Play, в некоторых случаях, необходима дополнительная настройка при работе с пиринговыми сетями и некоторыми другими программами, в которых необходимо не только инициировать исходящие соединения, но также принимать входящие.

1.2. Концепции трансляции адресов

Существует 3 основных концепции трансляции адресов:

- *Статический NAT* — Отображение незарегистрированного IP-адреса на зарегистрированный IP-адрес на основании один к одному. Особенно полезно, когда устройство должно быть доступным снаружи сети.
- *Динамический NAT* — Отображает незарегистрированный IP-адрес на зарегистрированный адрес из группы зарегистрированных IP-адресов. Динамический NAT также устанавливает непосредственное отображение между незарегистрированным и зарегистрированным адресом, но отображение может меняться в зависимости от зарегистрированного адреса, доступного в пуле адресов, во время коммуникации.
- *Перегруженный NAT (NAPT, NAT Overload, PAT, маскарадинг)* — форма динамического NAT, который отображает несколько незарегистрированных адресов в единственный зарегистрированный IP-адрес, используя различные порты. Известен также как PAT (*Port Address Translation*). При перегрузке каждый компьютер в частной сети транслируется в тот же самый адрес, но с различным номером порта.

1.3. Типы NAT

- *Симметричный NAT (Symmetric NAT)* — Трансляция, при которой каждое соединение, инициируемое парой «внутренний адрес - внутренний порт» преобразуется в свободную уникальную случайно выбранную пару «публичный адрес - публичный порт». При этом инициация соединения из публичной сети невозможна.
- *Cone NAT, Full Cone NAT* — Однозначная (взаимная) трансляция между парами «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт». Любой внешний хост может инициировать соединение с внутренним хостом (если это разрешено в правилах межсетевого экрана).
- *Address-Restricted cone NAT, Restricted cone NAT* — Постоянная трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт».

Любое соединение, инициированное с внутреннего адреса, позволяет в дальнейшем получать ему пакеты с любого порта того публичного хоста, к которому он отправлял пакет(ы) ранее.

- *Port-Restricted cone NAT* — Трансляция между парой «внутренний адрес: внутренний порт» и «публичный адрес: публичный порт», при которой входящие пакеты проходят на внутренний хост только с одного порта публичного хоста — того, на который внутренний хост уже посылал пакет.

1.4. Технология NAT Looback

Смысл технологии *NAT loopback* (или *NAT hairpinning*) прост: если пакет приходит из внутренней сети на внешний IP-адрес маршрутизатора, он считается пришедшим извне — а значит, работают правила брандмауэра, относящиеся ко внешним соединениям. И если пакет успешно пройдёт сквозь брандмауэр, сработает NAT, взяв на себя посредничество между двумя внутрисетевыми машинами. Это даёт две вещи:

- Прямо изнутри локальной сети можно проверить, как настроены сетевые службы.
- Доступ к серверу, находящемуся в локальной сети, по доменному имени. Без NAT loopback пришлось бы править файл `hosts` на каждой машине для каждого задействованного домена и поддомена.

Недостатком NAT loopback можно считать повышенную нагрузку на хаб и маршрутизатор (по сравнению с прямым доступом к серверу).

1.5. NAT Traversal

NAT Traversal (прохождение или автонастройка NAT) — это набор возможностей, позволяющих сетевым приложениям определять, что они находятся за устройством, обеспечивающим NAT, узнавать внешний IP-адрес этого устройства и выполнять сопоставление портов для пересылки пакетов из внешнего порта NAT на внутренний порт, используемый приложением; все это выполняется автоматически, пользователю нет необходимости вручную настраивать сопоставления портов или вносить изменения в какие-либо другие параметры. Однако существуют меры предосторожности в доверии к таким приложениям — они получают обширный контроль над устройством, появляются потенциальные уязвимости.

1.6. Пример настройки NAT

Создадим топологию по схеме, приведённой на рисунке 1. Данные об IP-адресах, масках и VLAN приведены в таблице 1.

Таблица 1 – Данные для настройки

PC	IP/Mask	VLAN	Gateway
4	192.168.3.3/24	Administration	192.168.3.1
5	192.168.3.2/24		
7	192.168.2.2/24	PCs	192.168.2.1
8	192.168.2.3/24		

На коммутаторе создать требуемые VLAN, перевести порты коммутатора SW3 в требуемые режимы работы.

При настройке маршрутизатора требуется разделить интерфейс `e0/0` на два субинтерфейса, каждый из которых будет относиться к определённой VLAN. Их конфигурация производится следующим образом:

```

int e0/0.2
encapsulation dot1q 2
ip address 192.168.2.1 255.255.255.0

```

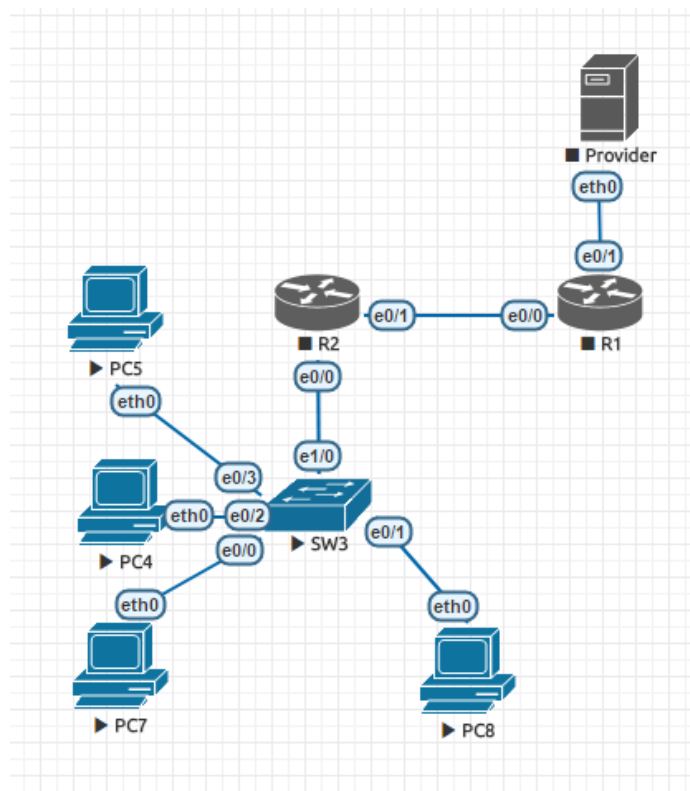


Рисунок 1 – Используемая топология

Инкапсуляция применяется в том случае, когда необходимо транслировать несколько VLAN через один порт. В коммутаторах Cisco используется два типа инкапсуляции: *dot1q* и *isl*. Стоит отметить, что для корректной работы необходимо использовать один и тот же тип инкапсуляции. В приведённой команде после типа инкапсуляции указывается номер VLAN, к которому относится данный интерфейс, следующей командой задаётся его IP и маска.

Предположим, что появилась необходимость подключить имеющуюся сеть к Интернету, т.е. к оборудованию провайдера (маршрутизатор провайдера R1 и серверное оборудование Provider). Порту доступа (e0/0) провайдер присвоил «белый» IP-адрес 200.100.10.1 и маску 255.255.255.252, сервер провайдера также имеет свой IP-адрес 200.100.20.2 и маску 255.255.255.252.

Для того чтобы подключиться к сети провайдера, необходимо настроить маршрутизатор R2, точнее говоря – интерфейс e0/1. Ему необходимо присвоить тот IP-адрес, который выдаётся провайдером, в данном случае подойдет IP-адрес 200.100.10.2 с той же маской, в качестве шлюза по умолчанию используется IP-адрес провайдера.

После произведённых настроек можно убедиться в доступности оборудования провайдера при помощи команды *ping*, результат выполнения которой показан на рисунке 2.

```

Router>ping 200.100.20.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.100.20.2, timeout is 2 seconds:
!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 5/5/7 ms

```

Рисунок 2 – Результат выполнения команды ping, выполняемой с маршрутизатора R2

Однако, подобный результат будет наблюдаться только в случае с маршрутизатором R2. Это происходит в силу того, что компьютеры локальной сети имеют «серые» IP-адреса, в связи с чем не могут подключиться к оборудованию провайдера напрямую. В этот момент приходит на помощь технология NAT, благодаря которой появляется возможность обеспечить компьютерам доступ в Интернет, в данном случае – доступ к серверу Provider.

При помощи команд *ip nat outside* и *ip nat inside* на маршрутизаторе R2 настроим внешние и внутренние интерфейсы. При помощи команд *ip access-list* и *permit* создадим access-list, включающий обе VLAN.

- *ip nat <outside/inside>* - команда для перевода интерфейсов во внешний или внутренний режимы;
- *ip access-list <num> <название>* - команда для создания access-list. Списки контроля доступа бывают *стандартными (standard)* и *расширенными (extended)*. Стандартные списки доступа позволяют указывать только адрес отправителя, расширенные – IP-адреса отправителя и получателя при указании протоколов IP, ICMP и др., порты отправителя и получателя при указании протоколов TCP и UDP.
- *permit* – служит для добавления записей в список контроля доступа. Запись, занесённая с данным ключом, попадёт в список доступа как разрешённая. Существует альтернатива – *deny*. После данного ключа следует запись IP-адреса и wildcard-маски.

После выполнения всех настроек, сервер Provider должен быть доступен с любого из PC's. Пример результата проверки показан на рисунке 3.

```
VPCS> ping 200.100.20.2
84 bytes from 200.100.20.2 icmp_seq=1 ttl=62 time=23.769 ms
84 bytes from 200.100.20.2 icmp_seq=2 ttl=62 time=2.363 ms
84 bytes from 200.100.20.2 icmp_seq=3 ttl=62 time=1.519 ms
84 bytes from 200.100.20.2 icmp_seq=4 ttl=62 time=1.475 ms
84 bytes from 200.100.20.2 icmp_seq=5 ttl=62 time=1.679 ms
```

Рисунок 3 – Проверка выполненных настроек

При помощи команды *show ip nat translations* можно посмотреть то, каким образом происходит трансляция локальных адресов компьютеров (рисунок 4).

```
Router#sh ip nat translations
Pro Inside global      Inside local      Outside local     Outside global
icmp 210.232.10.2:5312 192.168.2.2:5312 210.232.30.2:5312 210.232.30.2:5312
icmp 210.232.10.2:5568 192.168.2.2:5568 210.232.30.2:5568 210.232.30.2:5568
icmp 210.232.10.2:5824 192.168.2.2:5824 210.232.30.2:5824 210.232.30.2:5824
icmp 210.232.10.2:6080 192.168.2.2:6080 210.232.30.2:6080 210.232.30.2:6080
icmp 210.232.10.2:6336 192.168.2.2:6336 210.232.30.2:6336 210.232.30.2:6336
```

Рисунок 4 – Трансляция адресов

2. Практическая часть

Данная лабораторная работа выполняется в эмуляторе EVE-NG, первоначально не установленном в качестве виртуальной машины. Все необходимые действия приведены в порядке выполнения.

2.1. Порядок выполнения

2.1.1. Построить топологию сети в соответствии с заданием на рисунке 5.

2.1.2. Произвести настройку оборудования, разделить компьютеры на 4 VLAN, задать IP-адреса, маски, шлюз по умолчанию. Диапазон IP-адресов для каждой подсети выбрать по своему усмотрению.

2.1.3. Произвести настройку NAT на маршрутизаторе сети, обеспечив доступ к серверу для каждого компьютера каждой VLAN.

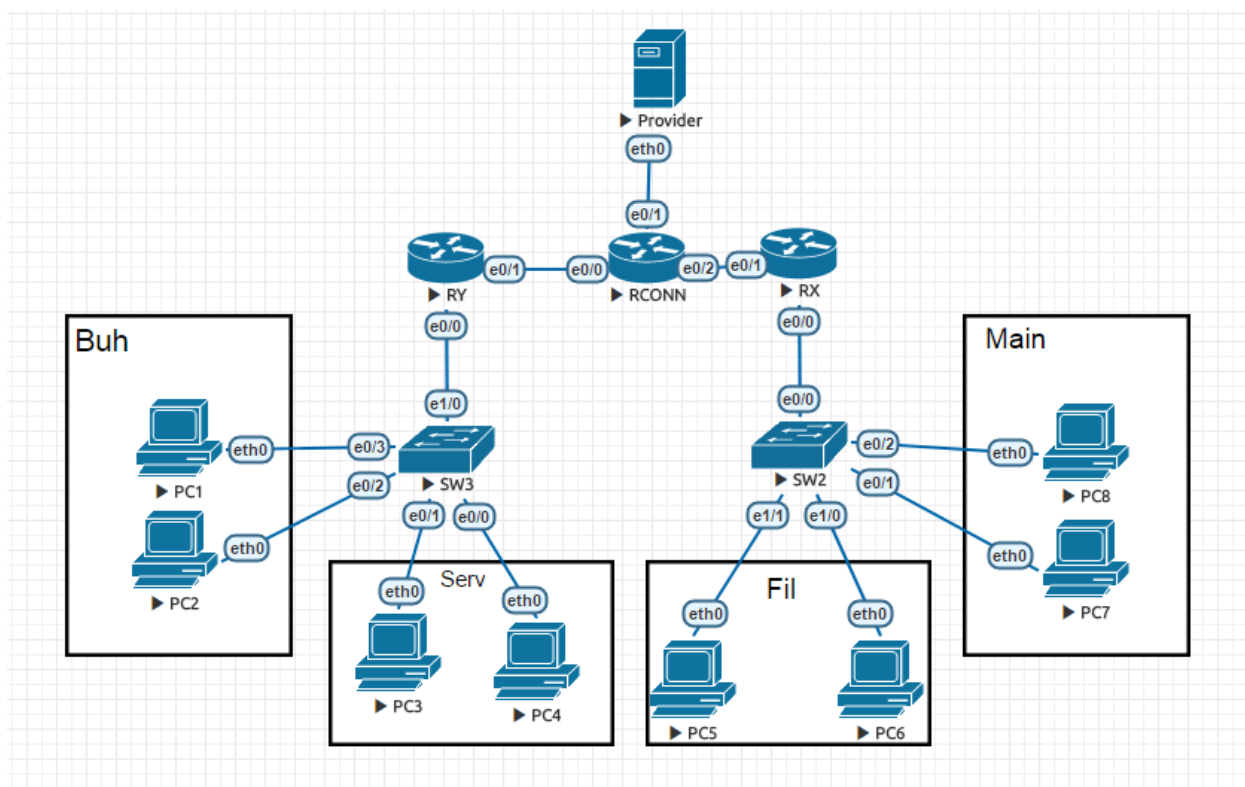


Рисунок 5 – Используемая топология

Лабораторная работа №4. VPN

1. Теоретическая часть

1.1. Основные сведения

VPN (от английского *virtual private network* – *виртуальная частная сеть*) – обобщённое название технологий, позволяющих обеспечить одно или несколько сетевых соединений поверх другой сети. Несмотря на то, что соединение осуществляется по сетям с меньшим уровнем доверия, уровень доверия к построенной логической сети не зависит от уровня доверия к базовой сети благодаря использованию механизмов шифрования. В зависимости от используемых протоколов и назначения, VPN может обеспечивать соединения трёх типов: *узел-узел*, *сеть-сеть* и *узел-сеть*.

VPN состоит из двух частей: «внутренняя» сеть (или сети) и «внешняя» сеть, по которой проходит соединение (обычно используется Интернет). Возможно также подключение к виртуальной сети отдельного компьютера. Подключение удалённого пользователя к VPN производится при помощи сервера доступа, который подключается как к внутренней, так и к внешней сети. При подключении удалённого пользователя сервер доступа требует прохождения процесса идентификации, а затем процесса аутентификации. После успешного прохождения обоих процессов, удалённому пользователю предоставляются определённые полномочия для работы в сети, т.е. происходит процесс авторизации.

1.2 Уровни реализации

Обычно VPN развёртывают на уровнях не выше сетевого, так как применение криптографии на этих уровнях позволяет использовать в неизменном виде транспортные протоколы.

Пользователи Windows обозначают термином VPN одну из реализаций виртуальной сети — PPTP, причём используемую зачастую не для создания частных сетей.

Чаще всего для создания виртуальной сети используется инкапсуляция протокола PPP в какой-либо другой протокол — IP (такой способ использует реализация PPTP — *Point-to-Point Tunneling Protocol*) или Ethernet (PPPoE – *PPP over Ethernet*). Технология VPN в последнее используется не только для создания частных сетей, но и предоставления доступа в Интернет.

При использовании специализированного ПО и необходимом уровне реализации сеть VPN может обеспечить высокую степень защищённости передаваемой информации. При правильной настройке всех компонентов технология VPN может обеспечивать анонимность в сети Интернет.

1.3. Классификация VPN

Классифицировать VPN можно по нескольким параметрам:

1. По степени защищённости:

- **Защищённые** - наиболее распространённый вариант виртуальных частных сетей. С его помощью возможно создать надёжную и защищённую сеть на основе ненадёжной сети, как правило, Интернета. Примером защищённых VPN являются: IPSec, OpenVPN и PPTP.
- **Доверительные** - Используются в случаях, когда передающую среду можно считать надёжной и необходимо решить лишь задачу создания виртуальной подсети в рамках большей сети. Проблемы безопасности становятся неактуальными. Примерами подобных VPN решений являются MPLS и L2TP.

2. По способу реализации:

- **VPN на основе маршрутизаторов** - данный способ построения VPN предполагает применение маршрутизаторов для создания защищенных каналов. Так как вся информация из локальной сети проходит через маршрутизатор, то рационально выполнять на нём её шифрование. Хорошим примером могут служить устройства Cisco.
- **VPN на основе межсетевых экранов** – межсетевые экраны большинства производителей поддерживают функции туннелирования и шифрования данных. Однако, данное решение подходит только для небольших сетей и сильно зависит от технических характеристик оборудования.
- **VPN на основе специализированных аппаратных средств** – обеспечивают высочайшую производительность в силу того, что шифрование выполняется специализированными микросхемами. Стоит отметить, что в данном случае производительность прямо пропорциональна цене оборудования.
- **VPN на основе программного обеспечения** - VPN-решения, реализованные программным способом, уступают специализированным устройствам по производительности, но обладают достаточной мощностью для реализации VPN-сетей. Главным достоинством программных средств является гибкость и удобство в применении, а также относительно невысокая стоимость.

3. По уровню сетевого протокола:

- **VPN канального уровня** - позволяют обеспечить инкапсуляцию различных видов трафика третьего уровня (и выше) и построение виртуальных туннелей типа «точка—точка».
- **VPN сетевого уровня** - выполняют инкапсуляцию IP в IP. Одним из широко известных протоколов на этом уровне является протокол IPSec (*IP Security*), предназначенным для аутентификации, туннелирования и шифрования IP-пакетов.
- **VPN сеансового уровня** - используют другой подход под названием «посредники каналов». Этот метод функционирует над транспортным уровнем и ретранслирует трафик из защищенной сети в Интернет для каждого сокета в отдельности.

4. По назначению:

- **Internet VPN** - используется для предоставления доступа к интернету провайдером, обычно если по одному физическому каналу подключаются несколько пользователей.
- **Intranet VPN** - используют для объединения в единую защищённую сеть нескольких распределённых филиалов одной организации, обменивающихся данными по открытым каналам связи.
- **Remote Access VPN** - используют для создания защищённого канала между сегментом корпоративной сети и одиночным пользователем, который подключается к корпоративным ресурсам с домашнего компьютера, корпоративного ноутбука, смартфона.
- **Extranet VPN** - используют для сетей, к которым подключаются «внешние» пользователи (например, заказчики или клиенты). Уровень доверия к ним ниже, чем к сотрудникам компании, поэтому требуется обеспечение специальных «рубежей» защиты, предотвращающих или ограничивающих доступ последних к конфиденциальной информации.

- **Client/Server VPN** - обеспечивает защиту передаваемых данных между двумя узлами корпоративной сети. Особенность данного варианта в том, что VPN строится между узлами, находящимися, как правило, в одном сегменте сети, например, между рабочей станцией и сервером. Такая необходимость возникает в тех случаях, когда в одной физической сети необходимо создать несколько логических сетей. Этот вариант похож на VLAN, но вместо разделения трафика, используется его шифрование.

1.4 Пример настройки VPN

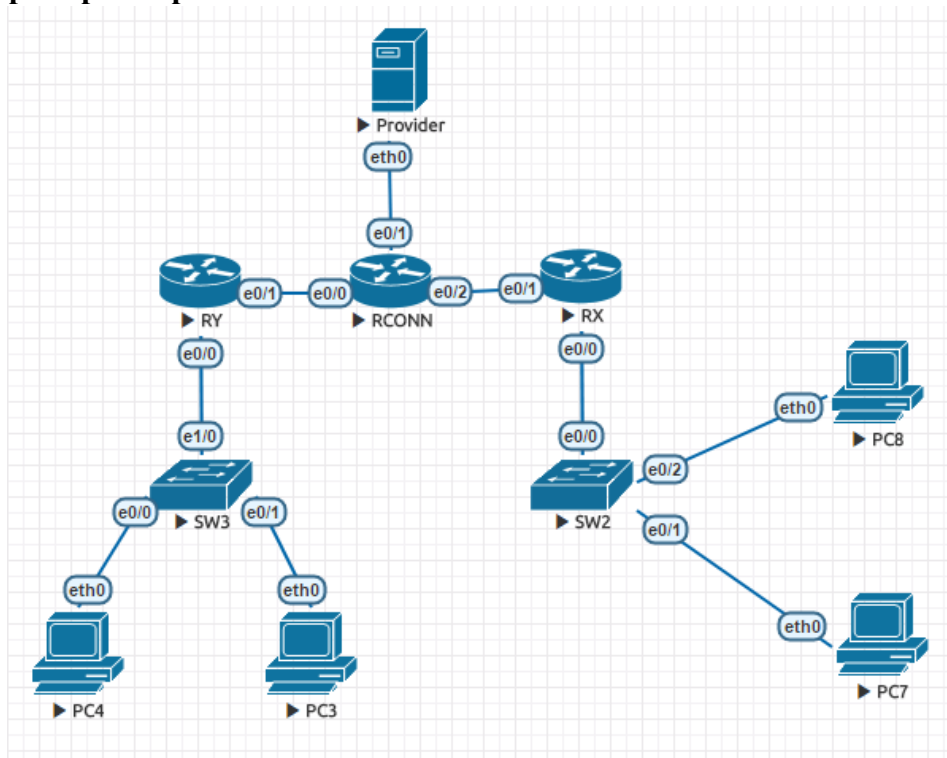


Рисунок 1 – Используемая топология

- Для настройки VPN будем использовать топологию, приведённую на рисунке 1. Оборудование провайдера (RCONN и Provider) уже настроено надлежащим образом, компьютерам PC присвоены IP-адреса, настроены шлюзы по умолчанию. Перед началом основной работы необходимо произвести настройку NAT в обеих локальных сетях, т.е. на маршрутизаторах RX и RY (см. рисунок 2). После того как вы убедились в том, что каждый из компьютеров имеет доступ в Интернет (к серверу Provider), можно приступать к настройке VPN.

```
interface Ethernet0/0
 ip address 192.168.2.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
!
interface Ethernet0/1
 ip address 210.232.10.2 255.255.255.252
 ip nat outside
 ip virtual-reassembly in
ip access-list standard NAT1
 permit 192.168.2.0 0.0.0.255
```

Рисунок 2 – Конфигурация маршрутизатора RY

- Произведем выбор параметров шифрования для туннеля на маршрутизаторе RY:

```
crypto isakmp policy 1
 encr 3des
 authentication pre-share
```

```
group 2
crypto ipsec transform-set ESP_3DES_SHA_HMAC esp-3des esp-sha-hmac
crypto ipsec df-bit clear
crypto ipsec profile VTI_PROF
set transform-set ESP_3DES_SHA_HMAC
set pfs group2
```

На маршрутизаторе RX будут выполнены аналогичные настройки.

- Настроим ключ шифрования на маршрутизаторах. Они должны быть идентичны:

```
crypto isakmp key 0 101010 address 210.232.20.2 - для RY;
```

```
crypto isakmp key 0 101010 address 210.232.10.2 - для RX.
```

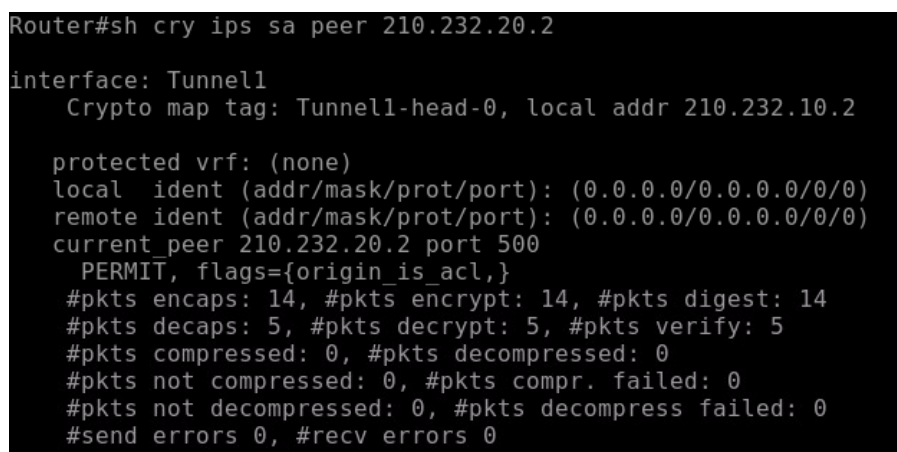
- Произведём создание и настройку тоннельных интерфейсов на маршрутизаторах RX и RY:

```
interface Tunnel1
ip address 10.0.0.1 255.255.255.252
tunnel source e0/1
tunnel destination 210.232.20.2
tunnel mode ipsec ipv4
tunnel protection ipsec profile VTI_PROF - для RY;
```

```
interface Tunnel1
ip address 10.0.0.1 255.255.255.252
tunnel source e0/1
tunnel destination 210.232.10.2
tunnel mode ipsec ipv4
tunnel protection ipsec profile VTI_PROF - для RX.
```

Рассмотрим выполненные команды:

- *ip address 10.0.0.1 255.255.255.252* – собственный адрес туннеля;
- *tunnel source e0/1* – собственный внешний интерфейс маршрутизатора;
- *tunnel destination 210.232.10.2* – внешний интерфейс маршрутизатора назначения;
- *tunnel mode ipsec ipv4* – вид шифрования;
- *tunnel protection ipsec profile VTI_PROF* – способ шифрования.



```
Router#sh cry ips sa peer 210.232.20.2
interface: Tunnel1
  Crypto map tag: Tunnel1-head-0, local addr 210.232.10.2

  protected vrf: (none)
  local  ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
  remote ident (addr/mask/prot/port): (0.0.0.0/0.0.0.0/0/0)
  current_peer 210.232.20.2 port 500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps: 14, #pkts encrypt: 14, #pkts digest: 14
    #pkts decaps: 5, #pkts decrypt: 5, #pkts verify: 5
    #pkts compressed: 0, #pkts decompressed: 0
    #pkts not compressed: 0, #pkts compr. failed: 0
    #pkts not decompressed: 0, #pkts decompress failed: 0
    #send errors 0, #recv errors 0
```

Рисунок 3 – Проверка доступности и зашифрованности созданного туннеля

После произведённых настроек необходимо добавить маршруты для каждого из маршрутизаторов следующим образом:

```
ip route 192.168.3.0 255.255.255.0 10.0.0.2 – для RY;
```

```
ip route 192.168.2.0 255.255.255.0 10.0.0.1 – для RX.
```

После этого все компьютеры обеих сетей должны быть доступны друг другу, а соединение между ними должно быть защищено.

```
C4
VPCS> ping 192.168.3.3

84 bytes from 192.168.3.3 icmp_seq=1 ttl=62 time=101.362 ms
84 bytes from 192.168.3.3 icmp_seq=2 ttl=62 time=3.017 ms
84 bytes from 192.168.3.3 icmp_seq=3 ttl=62 time=3.049 ms
84 bytes from 192.168.3.3 icmp_seq=4 ttl=62 time=3.167 ms
84 bytes from 192.168.3.3 icmp_seq=5 ttl=62 time=3.045 ms
```

Рисунок 4 – Проверка соединения между PC4 и PC8

2. Практическая часть

Данная лабораторная работа выполняется в эмуляторе EVE-NG, первоначально не установленном в качестве виртуальной машины. Все необходимые действия приведены в порядке выполнения.

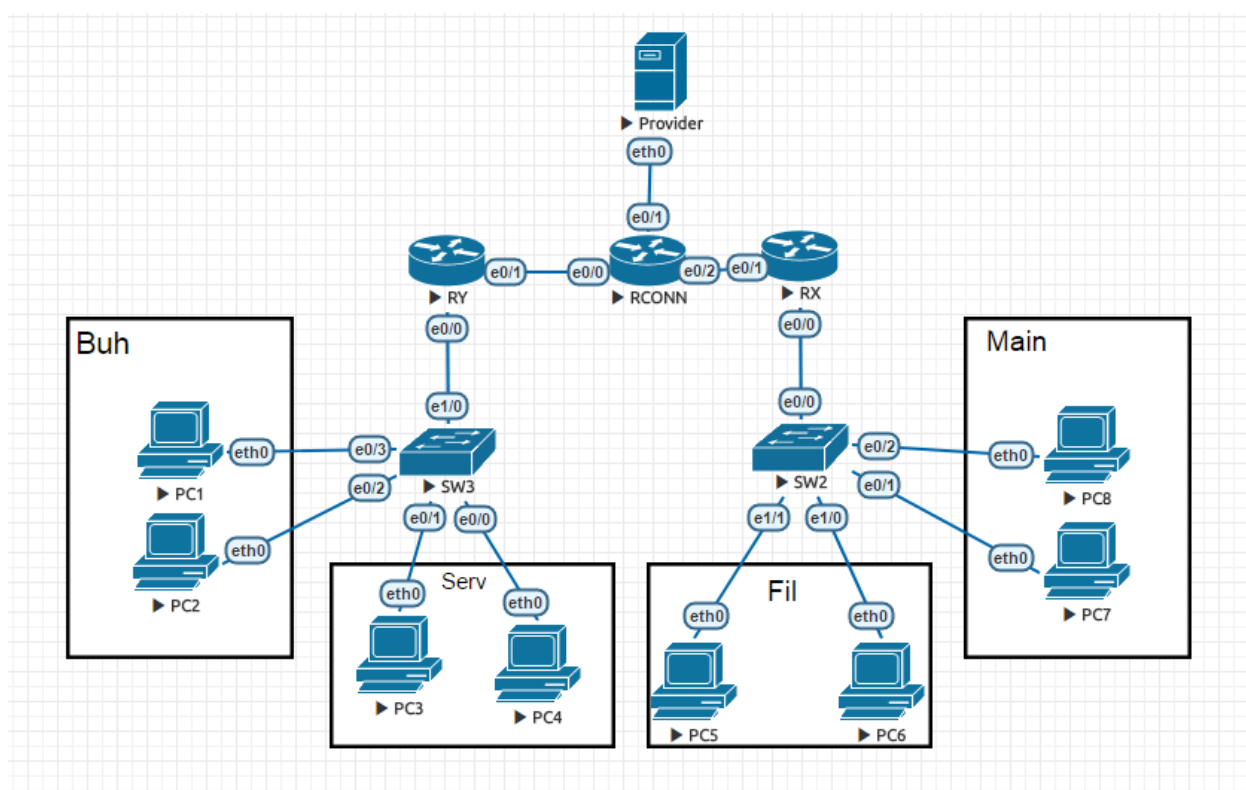


Рисунок 5 – Используемая топология

2.1 Порядок выполнения работы

- 2.1.1 Построить топологию сети в соответствии с рисунком 5.
- 2.1.2 Произвести настройку оборудования, задать IP-адреса устройств, распределить устройства на 4 VLAN.
- 2.1.3 Настроить NAT на маршрутизаторах.
- 2.1.4 Создать VPN-тоннель для объединения всех компьютеров в виртуальную частную сеть. Проверить выполненные настройки *различными способами*.

Лабораторная работа № 5. Протокол BGP.

1. Теоретическая часть

1.1. Общие сведения

Протокол **BGP (Border Gateway Protocol)** – *протокол граничного шлюза*. Это основной протокол динамической маршрутизации, используемый в Интернете. Вместе с DNS, BGP является одним из главных механизмов, обеспечивающих функционирование Интернета. Это протокол прикладного уровня, функционирующий поверх протокола транспортного уровня TCP. Данный протокол поддерживает бесклассовую адресацию, использует суммирование маршрутов для уменьшения таблиц маршрутизации. На данный момент используется четвёртая версия протокола, все остальные считаются устаревшими с 1994 года. Маршрутизаторы, использующие данный протокол, обмениваются информацией о доступности сетей. Наряду с этой информацией передаются и различные атрибуты этих сетей, с помощью которых выбирается лучший маршрут и политика маршрутизации.

Среди атрибутов большую роль играет список автономных систем, через которые прошла информация. Это позволяет определять положение сети относительно автономных систем, использоваться для определения политик, исключения петель маршрутизации.

Сам процесс маршрутизации осуществляется пошагово, т.е. от одной автономной системы к другой. В основном, все политики BGP настраиваются по отношению к внешним (соседним) системам – описываются политики их взаимодействия. В силу того, что протокол BGP оперирует колоссальными таблицами маршрутизации (на данный момент таблица маршрутизации для IPv4 насчитывает более 450 тысяч маршрутов), принципы его настройки отличаются от внутренних протоколов маршрутизации (IGP).

1.2. Описание протокола

В отличие от внутренних протоколов маршрутизации, исходящих из технических характеристик сети при выборе маршрутов, BGP производит выбор наилучших маршрутов на основании политик. Это происходит в силу того, что при выборе между каналами двух провайдеров наибольшее значение имеют не технические характеристики, а ряд внутренних правил компании, к примеру – стоимость использования каналов. Именно по этой причине выбор наиболее выгодного маршрута в BGP производится на основании настраиваемых при помощи изменения атрибутов и применения фильтров политик.

Протокол BGP так же, как и остальные протоколы динамической маршрутизации, может производить передачу трафика только на основании IP-адреса получателя. Из этого следует, что при помощи данного протокола нельзя указать правила маршрутизации, в которых будет учитываться, к примеру, то, из какой сети был отправлен пакет. В том случае, если необходимо принимать решение о том, как должен передаваться трафик по каким-то дополнительным атрибутам, кроме адреса получателя, следует использовать механизм *PBR – policy-based routing*.

В целом, BGP – это *path-vector* протокол, имеющий следующие общие характеристики:

- Для обеспечения надежности доставки обновлений протокола использует TCP;
- Периодические обновления отсутствуют;
- Время от времени отправляет *keepalive-сообщения* для проверки TCP-соединения;

Стоит также привести некоторые теоретические сведения по термину, применявшемуся выше – *автономная система*. *Автономная система* (AS) – система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющие единую и чётко определённую политику соединения с интернетом. Имеют несколько диапазонов номеров (ASN – autonomous system number):

- 0-65535 для ASN 16 бит;
- 65536-4294967295 для ASN 32 бит;

Протокол разделяется на два класса:

- *Внутренний* iBGP (от англ. *internal* – *внутренний*) – BGP, работающий внутри AS. iBGP-соседи не обязательно должны быть соединены непосредственно;
- *Внешний* eBGP (от англ. *External* – *внешний*) – BGP, работающий между AS. По умолчанию, соседи eBGP должны быть непосредственно соединены.

Если iBGP-маршрутизаторы работают в нетранзитной автономной системе, то они должны быть соединены по принципу «каждый с каждым». Это следствие принципов работы протокола — если маршрутизатор, находящийся на границе автономной системы, получил обновление, то он передает его всем соседям; соседи, которые находятся внутри автономной системы, больше это обновление не распространяют, так как считают, что все соседи внутри системы уже его получили.

1.3. Формат сообщения BGP

Сообщение BGP всегда начинается с заголовка, после которого (в зависимости от типа сообщения) могут следовать данные. Максимальная длина сообщения – 4096 октетов, минимальная – 19 октетов. Заголовок сообщения содержит следующие поля:

- Маркер (16 октетов) – заполняется единицами и используется для совместимости;
- Длина (2 октета) – указывает длину сообщения, включая заголовок.
- Тип (1 октет):
 - 1 – открытие;
 - 2 – обновление информации;
 - 3 – оповещение;
 - 4 – сохранение соединения (keepalive);

Первое сообщение после установки соединения должно быть «Открытие». Если сообщение успешно обработано, в ответ будет послано «Сохранение соединения». В дополнение к заголовку BGP сообщение «Открытие» содержит следующие поля:

- Версия (1 октет) — версия протокола, текущее значение 4;
- Моя система (2 октета) — номер автономной системы;
- Интервал времени (2 октета) — максимальный интервал времени в секундах между получением сообщений «Обновление информации» или «Сохранение соединения»;
- Идентификатор отправителя (4 октета) — устанавливается равным IP-адресу;
- Длина дополнительных параметров (1 октет);
- Дополнительные параметры:
 - Тип параметра (1 октет);

- Длина параметра (1 октет);
- Значение параметра.

Сообщение «Обновление информации» предназначено для передачи информации о маршрутах между Сообщением «Обновление информации» предназначено для передачи информации о маршрутах между АС. Сообщение может указывать новые маршруты и удалять неработающие. Структура сообщения:

- Длина удаляемых маршрутов (2 октета);
- Удаляемые маршруты:
 - Длина (1 октет) — длина в битах префикса IP-адреса;
 - Префикс IP-адреса, дополненный минимальным количеством бит до полного октета;
- Длина атрибутов пути (2 октета);
- Атрибуты пути:
 - Тип атрибута:
 - Флаг атрибута;
 - Код атрибута;
 - Длина атрибута (1 или 2 октета, в зависимости от флага);
 - Данные атрибута;
- Информация о достижимости — список префиксов IP-адресов:
 - Длина (1 октет) — длина в битах префикса IP-адреса (нулевая длина — соответствие всем IP-адресам);
 - Префикс IP-адреса, дополненный минимальным количеством бит до полного октета.

Все атрибуты пути соответствуют всем записям в поле «Информация о достижимости».

Сообщение сохранения соединения должно посылаться не реже чем раз в одну третью часть максимального интервала времени между сообщениями, но не чаще чем один раз в секунду. Если интервал времени установлен равным нулю, то сообщение не должно периодически рассылаться. Сообщение не использует дополнительных полей.

Оповещение посылается в случае обнаружения ошибки, при этом соединение закрывается. Сообщение содержит следующие поля:

- Код ошибки (1 октет);
- Субкод (1 октет);
- Данные.

Сам же алгоритм работы протокола приведён на рисунке 1.

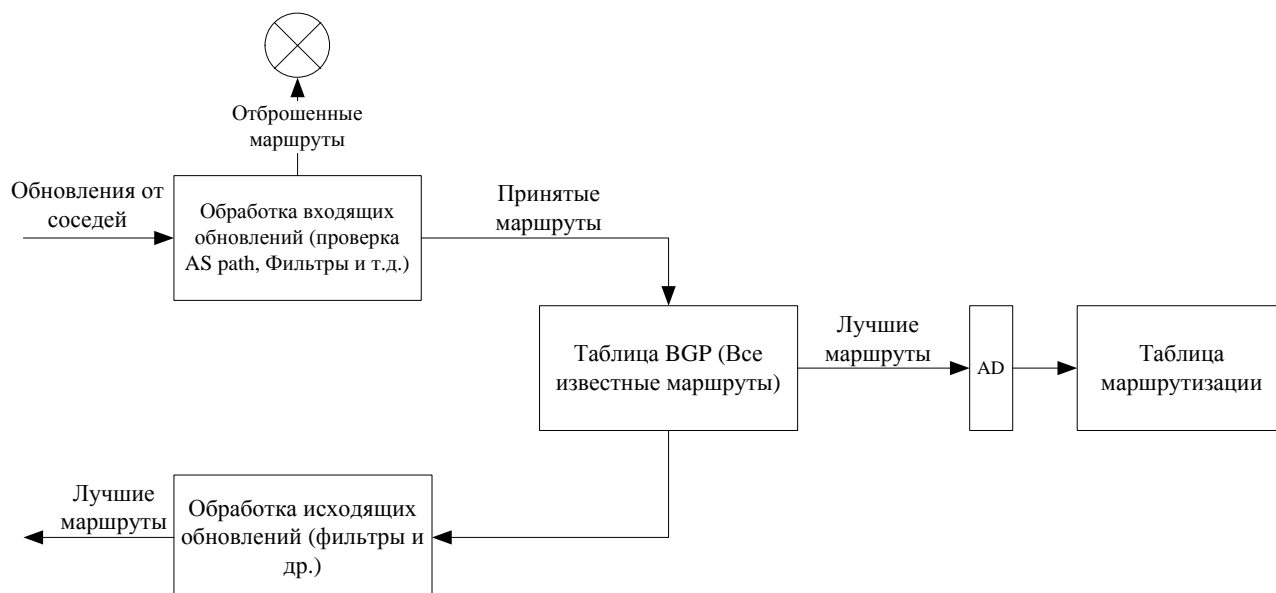


Рисунок 1 – Схема алгоритма работы BGP

1.4. Отношения соседства

Протокол BGP требует ручной настройки соседей для установления отношения соседства. Когда указывается сосед локального маршрутизатора, обязательно указывается AS соседа. Исходя из этой информации определяется тип соседа:

- Внутренний BGP-сосед (iBGP-сосед) — сосед, который находится в той же AS, что и локальный маршрутизатор. iBGP-соседи не обязательно должны быть непосредственно соединены;
- Внешний BGP-сосед (eBGP-сосед) — сосед, который находится в отличной от локального маршрутизатора AS. eBGP-соседи должны быть непосредственно соединены.

При формировании отношений соседства протоколом проводится ряд проверок:

- Маршрутизатор должен получить запрос на TCP-соединение с адресом отправителя, который маршрутизатор найдет указанным в списке соседей (команда *neighbor*);
- Номер AS локального маршрутизатора должен совпадать с номером AS, который указан на соседнем маршрутизаторе командой *neighbor remote-as* (это требование не соблюдается при настройках конфедераций);
- Идентификаторы маршрутизаторов (*Router ID*) не должны совпадать;
- Соседи должны пройти аутентификацию в том случае, если она настроена.

У первого пункта проверки есть важная особенность: только у одного из маршрутизаторов IP-адрес, указанный как адрес отправки обновлений, должен быть указан в команде *neighbor* другого маршрутизатора.

Состояния связи с соседями приведены в таблице 1.

Таблица 1 – Состояния связи с соседями

Состояние	Ожидание TCP?	Инициация TCP?	Установлено TCP?	Отправлено Open?	Получено Open?	Сосед Up?
Idle	Нет					
Connect	Да					
Active	Да	Да				
Open sent	Да	Да	Да	Да		
Open confirm	Да	Да	Да	Да	Да	
Established	Да	Да	Да	Да	Да	Да

1.5. Атрибуты пути

Атрибуты пути подразделяются на 4 основных категории:

- *Well-known mandatory* — все маршрутизаторы, работающие по протоколу BGP, должны распознавать эти атрибуты. Такие атрибуты обязательно присутствуют во всех обновлениях;
- *Well-known discretionary* — все маршрутизаторы, работающие по протоколу BGP, должны распознавать эти атрибуты. Такие атрибуты не обязательно присутствуют в обновлениях;
- *Optional transitive* — могут не распознаваться всеми реализациями BGP. Если маршрутизатор не распознал атрибут, он помечает обновление как частичное (*partial*) и отправляет его дальше соседям, сохраняя не распознанный атрибут;
- *Optional non-transitive* — могут не распознаваться всеми реализациями BGP. Если маршрутизатор не распознал атрибут, то атрибут игнорируется и при передаче соседям отбрасывается.

Рассмотрим некоторые основные атрибуты из категории *well-known mandatory*:

- *Autonomous system path (AS Path)* - атрибут описывает, через какие автономные системы необходимо пройти для достижения сети назначения. При этом, номер AS добавляется при передаче обновления из одной AS eBGP-соседу в другой AS. Атрибут используется для обнаружения петель и применения политик.
- *Next-hop* – атрибут представляет собой IP-адрес следующей AS для достижения сети назначения, IP-адрес eBGP-маршрутизатора, через который пролегает путь к сети назначения.
- *Origin* – указывает на то, каким образом был получен маршрут в обновлении:
 - 0 – IGP – внутри исходной автономной системы;
 - 1 – EGP – получен по протоколу EGP, в настоящее время не используется;
 - 2 – Incomplete (каким-то иным образом).

В категории *well-known discretionary* можно отметить атрибут *Local Preference*, который:

- Указывает путь выхода за пределы системы маршрутизаторам внутри AS;
- Передается в пределах одной AS;
- На маршрутизаторах Cisco его значение по умолчанию равно 100;
- Выбирается такая точка входа, у которой значение данного атрибута наибольшее;
- При получении eBGP-соседом обновления с выставленным значением этого атрибута, он игнорируется.

Атрибут *Aggregator* из категории optional transitive указывает список RID и ASN маршрутизаторов, создавших суммарную NLRI – Network Layer Reachability Information – информацию о маршрутах.

Из категории optional non-transitive в качестве примера приведём два атрибута:

- *Multi-Exit Discriminator* (MED):
 - Используется для информирования eBGP-соседей о том, какой путь в автономную систему более предпочтителен;
 - Атрибут передается между автономными системами;
 - Маршрутизаторы внутри соседней автономной системы используют этот атрибут, но, как только обновление выходит за пределы AS, атрибут MED отбрасывается;
 - Чем меньше значение атрибута, тем более предпочтительна точка входа в автономную систему.
- *Weight* (проприетарный атрибут Cisco):
 - Позволяет назначить "вес" различным путям локально на маршрутизаторе;
 - Используется в тех случаях, когда у одного маршрутизатора есть несколько выходов из автономной системы (сам маршрутизатор является точкой выхода);
 - Имеет значение только локально, в пределах маршрутизатора;
 - Не передается в обновлениях;
 - Чем больше значение атрибута, тем более предпочтителен путь выхода.

1.6. Схема выбора пути

Характеристики процедуры выбора пути протоколом BGP:

- В таблице BGP хранятся все известные пути, а в таблице маршрутизации — лучшие.
- Пути выбираются на основании политик.
- Пути не выбираются на основании пропускной способности.

Сначала проверяется, доступен ли Next-hop.

В случае с маршрутизатором Cisco, выбор пути происходит следующим образом (на каждый следующий шаг маршрутизатор переходит только при совпадении значений на предыдущем):

- Максимальное значение weight (локально для маршрутизатора);
- Максимальное значение local preference (для всей AS);
- Предпочесть локальный маршрут маршрутизатора (next hop = 0.0.0.0);
- Кратчайший путь через автономные системы. (самый короткий AS_PATH);
- Минимальное значение origin code (IGP < EGP < Incomplete);
- Минимальное значение MED (распространяется между автономными системами);
- Путь eBGP лучше чем путь iBGP;
- Выбрать путь через ближайшего IGP-соседа;
- Выбрать самый старый маршрут для eBGP-пути;
- Выбрать путь через соседа с наименьшим BGP router ID;
- Выбрать путь через соседа с наименьшим IP-адресом.

1.7. Конфедерации

Использование конфедераций позволяет избежать необходимости полной связности внутренних соседей BGP. При использовании конфедераций исходная AS разбивается на

подавтономные системы (sub-AS), внутри которых соседи должны быть соединены друг с другом в полносвязной топологии. В целом, использование конфедераций позволяет следующее:

- Избежать необходимости создания полносвязной топологии между всеми iBGP-соседями;
- Всем iBGP-соседям выучить все iBGP-маршруты в автономной системе;
- Предотвратить образование петель.

Существует ряд правил, по которым работают маршрутизаторы в конфедерации:

- iBGP-соседи в конфедерации должны быть соединены в полносвязную топологию. Они, как и обычные iBGP-соседи, не передают iBGP-маршруты друг другу;
- eBGP-соседи в конфедерации:
 - анонсируют iBGP-маршруты, выученные внутри sub-AS конфедерации, в другую sub-AS;
 - по умолчанию используют для пакетов TTL = 1;
 - во всех остальных случаях работают как обычные iBGP-соседи (например, next-hop по умолчанию не изменяется);
- Внутри конфедераций для предотвращения петель используется атрибут AS Path. Маршрутизаторы, которые находятся в конфедерации добавляют в атрибут сегменты AS_CONFED_SEQ и AS_CONFED_SET;
- Когда маршрутизатор выбирает лучший маршрут на основании атрибута AS Path, номера автономных систем конфедераций не учитываются;
- Когда обновление отправляется маршрутизатору, который не находится в конфедерации, номера конфедераций удаляются.

Рассмотрим пример создания конфедерации. Используемая топология приведена на рисунке 2.

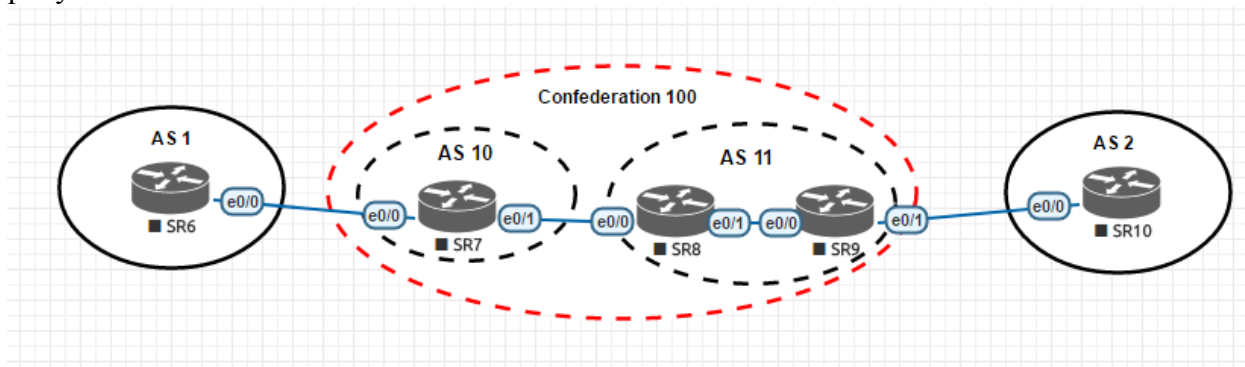


Рисунок 2 – Используемая топология

Требуется произвести настройку всех маршрутизаторов, при этом необходимо учесть, что автономные системы AS 10 и AS 11 образуют конфедерацию. Рассмотрим настройку маршрутизатора SR7:

```
router bgp 10
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 100
bgp confederation peers 11
network 192.168.1.0
network 192.168.2.0
neighbor 192.168.1.2 remote-as 1
```

```
neighbor 192.168.2.3 remote-as 11
no auto-summary
```

После настройки данного маршрутизатора произведём настройку его внешнего соседа SR8:

```
router bgp 11
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 100
bgp confederation peers 10
network 192.168.2.0
network 192.168.4.0
neighbor 192.168.2.1 remote-as 10
neighbor 192.168.4.5 remote-as 11
no auto-summary
```

Конфигурация BGP на маршрутизаторе SR9 выглядит следующим образом:

```
router bgp 11
no synchronization
bgp log-neighbor-changes
bgp confederation identifier 100
bgp confederation peers 10
network 192.168.4.0
network 192.168.5.0
network 192.168.100.0
neighbor 192.168.4.3 remote-as 11
neighbor 192.168.5.6 remote-as 2
no auto-summary
```

Для маршрутизатора SR10 используется конфигурация:

```
router bgp 2
bgp router-id 192.168.5.6
network 192.168.5.0/24
neighbor 192.168.5.5 remote-as 100
```

Маршрутизатор SR6 имеет практически аналогичные приведённым выше настройки BGP.

В процессе настройки оборудования использовались команды:

- `no synchronization` - отключает механизм синхронизации. Как правило, синхронизация не включается. При включенном правиле синхронизации все маршруты BGP должны быть получены и по протоколу IGP. То есть, должно быть настроено перераспределение маршрутов BGP в протокол IGP;
- `bgp log-neighbor-changes` - включает протоколирование изменения состояния BGP-соседей;
- `bgp confederation identifier` - назначает идентификатор конфедерации;
- `bgp confederation peers` - назначает номера AS, принадлежащих к конфедерации;
- `neighbor <IP маршрутизатора> remote-as <ASN>` - устанавливает соединение с другой AS, добавляя запись в таблицу маршрутизации.

2. Практическая часть

Данная лабораторная работа выполняется в эмуляторе EVE-NG, первоначально не установленном в качестве виртуальной машины. Все необходимые действия приведены в порядке выполнения.

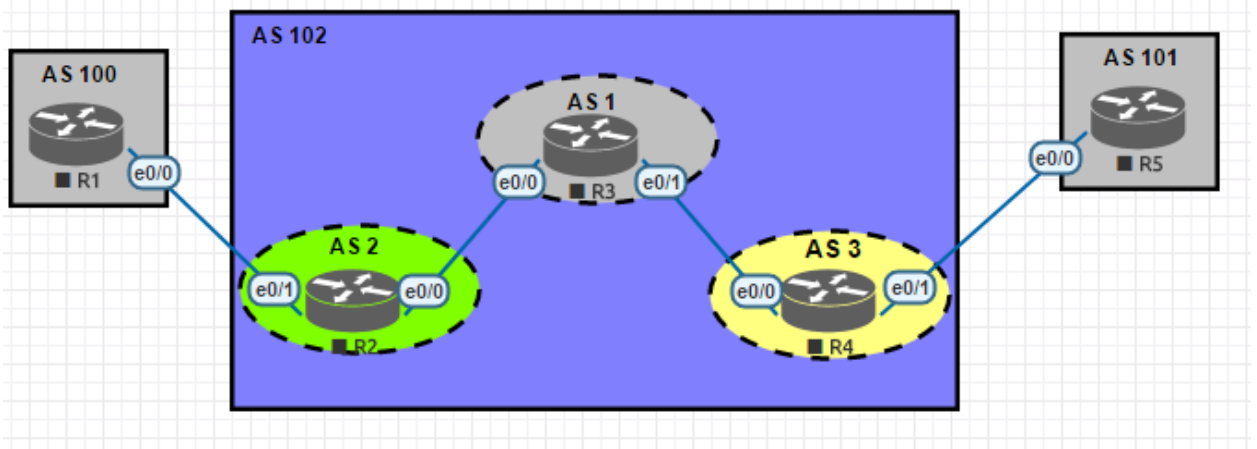


Рисунок 3 – Используемая топология

2.1. Порядок выполнения работы

- 2.1.1. Построить приведённую на рисунке 3 топологию, используя новый стенд эмулятора EVE-NG.
- 2.1.2. Произвести конфигурирование оборудования согласно приведённой на рисунке 3 схеме:
 - Создать три автономные системы: AS 100-102;
 - Создать три подавтономные системы AS 1-3 в составе AS 102;
 - Произвести настройку оборудования для корректной работы топологии;
- 2.1.3. Проверить полученную топологию на работоспособность. Для просмотра маршрутной информации и состояния соединений BGP использовать команды `show ip bgp paths` и `show ip bgp summary`.

ЛАБОРАТОРНАЯ РАБОТА №6. МЕЖСЕТЕВЫЕ ЭКРАНЫ

1.1. Общие сведения

Межсетевой экран, сетевой экран — программный или программно-аппаратный узел сети, который выполняет функцию контроля и фильтрации проходящего через него трафика исходя из заданных правил. Также часто встречаются другие названия:

- *Брандмауэр (от нем. Brandmauer – противопожарная стена)*
- *Файрвол (от англ. Firewall – имеет аналогичное значение)*

Межсетевые экраны решают задачи защиты сегментов сети или отдельных устройств от неправомерного доступа, который может производиться через уязвимости в протоколах модели OSI или в программном обеспечении, установленном на самих устройствах.

Межсетевые экраны запрещают или пропускают трафик, определяя разрешенный путём сравнения его характеристик с установленными шаблонами.

Чаще всего установка межсетевых экранов производится на границе периметра локальных сетей – преследуется цель защиты внутренних узлов от атак извне. С другой стороны, атаки могут начинаться и с внутренних устройств, но в таком случае, если атакуемый узел находится в той же сети, трафик не будет пересекать границу периметра, на которой установлен сетевой экран, и он, соответственно, не будет задействован. Исходя из возможности возникновения подобных ситуаций, межсетевые экраны размещают не только на границе, но и между различными сегментами сети, тем самым обеспечивая дополнительный уровень безопасности.

1.2. История создания и развития

Первые устройства, которые фильтровали сетевой трафик, появились в конце 80-х. Интернет был ещё очень молодым и не использовался в таких грандиозных масштабах, как сейчас. Это были маршрутизаторы, инспектирующие трафик на основании данных, содержащихся в заголовках протоколов сетевого уровня. Впоследствии, с развитием сетевых технологий, данные устройства получили возможность выполнять фильтрацию трафика, используя данные протоколов более высокого, транспортного уровня. Маршрутизаторы можно по праву считать первой программно-аппаратной реализацией межсетевого экрана.

Программные межсетевые экраны появились гораздо позже и были существенно моложе, чем антивирусы. К примеру, проект *Netfilter/iptables* (программный файрвол, встраиваемый в ядро *Linux* с версии 2.4) был основан в 1998 году. Такое позднее появление вполне объяснимо, так как долгое время антивирус решал проблему защиты персональных компьютеров от вредоносных программ. Однако в конце 1990-х вирусы стали активно использовать отсутствие межсетевых экранов на компьютерах, что привело к повышению интереса пользователей к данному классу устройств. В данной работе мы будем рассматривать устройства *Cisco ASA*, в основе которых лежат следующие разработки:

- **Cisco PIX** (*Private Internet Exchange*) — межсетевой экран с преобразованием сетевых адресов (NAT), выпускавшийся американской компанией Cisco Systems. Один из первых продуктов в этом сегменте рынка. Устройство было разработано в 1994 году инженерами компании *Network Translation* как функциональный аналог телефонной станции *PBX* (*Private Branch Exchange*, производным от чего и стало название PIX) с целью решения проблемы нехватки публичных IP-адресов.

Разработчики хотели скрыть блок частных адресов за одним или несколькими публичными — так же, как это делается в АТС для внутренних телефонов.

- **Cisco IPS 4200** - система обнаружения вторжений (COB). Система обнаружения вторжений программное или аппаратное средство, предназначенное для выявления фактов неавторизованного доступа в компьютерную систему или сеть либо несанкционированного управления ими в основном через Интернет. Соответствующий английский термин — *Intrusion Detection System (IDS)*. Системы обнаружения вторжений обеспечивают дополнительный уровень защиты компьютерных систем.
- **Cisco VPN 3000 Concentrator** – серия специализированных платформ и клиентского ПО для построения сетей удаленного доступа на основе технологии виртуальных частных сетей (Virtual Private Networks). Объединяя в себе высокую доступность, производительность, масштабируемость и поддержку прогрессивных алгоритмов аутентификации и шифрования, серия Cisco VPN 3000 Concentrator позволяет существенно снизить затраты компании на удаленный доступ. Благодаря модульной архитектуре платформа позволяет наращивать производительность и емкость.

Если углубиться в конкретику, в данной работе мы будем использовать *Cisco ASA 5500*. Линейка модульных многофункциональных устройств защиты *ASA (Adaptive Security Appliance) 5500* была создана корпорацией Cisco Systems как логическое продолжение развития линейки Cisco PIX – очень популярного решения для защиты периметра сетевой инфраструктуры предприятия. Основной идеей при создании линейки ASA 5500 была новая стратегия Cisco Systems по созданию *SDN (Self Defending Networks)*, обеспечивающая комплексную защиту сети, позволяющую активно реагировать на сетевые атаки и предотвращать их в самом начале. В инфраструктуре предприятия ASA 5500 занимает место классического межсетевого экрана (*firewall*), позволяющего защитить сеть предприятия или отдельные сегменты, но функционал данного решения намного шире. ASA 5500 объединяет в себе множество технологий и решений по обеспечению информационной безопасности.

1.3. Фильтрация трафика

Фильтрация трафика осуществляется на основе набора предварительно сконфигурированных правил, которые называются *ruleset*. Удобно представлять межсетевой экран как последовательность фильтров, обрабатывающих информационный поток. Каждый из фильтров предназначен для интерпретации отдельного правила. Последовательность правил в наборе существенно влияет на производительность межсетевого экрана. Например, многие межсетевые экраны последовательно сравнивают трафик с правилами до тех пор, пока не будет найдено соответствие. Для таких межсетевых экранов, правила, которые соответствуют наибольшему количеству трафика, следует располагать как можно выше в списке, увеличивая тем самым производительность.

Существует два принципа обработки поступающего трафика. Первый принцип гласит: «*Что явно не запрещено, то разрешено*». В данном случае, если межсетевой экран получил пакет, не попадающий ни под одно правило, то он передаётся далее.

Противоположный принцип — «*Что явно не разрешено, то запрещено*» — гарантирует гораздо большую защищённость, так как он запрещает весь трафик, который явно не разрешён правилами. Однако, этот принцип оборачивается дополнительной нагрузкой на администратора.

В конечном счёте, межсетевые экраны выполняют над поступающим трафиком одну из двух операций: пропустить пакет далее (*allow*) или отбросить пакет (*deny*). Некоторые межсетевые экраны имеют ещё одну операцию — *reject*, при которой пакет отбрасывается, но отправителю сообщается о недоступности сервиса, доступ к которому он пытался получить. В противовес этому, при операции *deny* отправитель не информируется о недоступности сервиса, что является более безопасным.

1.4. Классификация межсетевых экранов

До сих пор не существует единой и общепризнанной классификации межсетевых экранов. Однако в большинстве случаев поддерживаемый уровень сетевой модели OSI является основной характеристикой при их классификации. Учитывая данную модель, различают следующие типы межсетевых экранов:

- *Управляемые коммутаторы.*
- *Пакетные фильтры.*
- *Шлюзы сеансового уровня.*
- *Посредники прикладного уровня.*
- *Инспекторы состояния.*

1.4.1. Управляемые коммутаторы

Управляемые коммутаторы иногда причисляют к классу межсетевых экранов, так как они осуществляют фильтрацию трафика между сетями или узлами сети. Однако они работают на канальном уровне и разделяют трафик в рамках локальной сети, а значит не могут быть использованы для обработки трафика из внешних сетей (например, из Интернета).

Многие производители сетевого оборудования, такие как Cisco, Nortel, 3Com, ZyXEL, предоставляют в своих коммутаторах возможность фильтрации трафика на основе MAC-адресов, содержащихся в заголовках фреймов. Например, в коммутаторах семейства Cisco Catalyst эта возможность реализована при помощи механизма Port Security. Однако данный метод фильтрации не является эффективным, так как аппаратно установленный в сетевой карте MAC-адрес легко меняется программным путем, поскольку значение, указанное через драйвер, имеет более высокий приоритет, чем зашитое в плату. Поэтому многие современные коммутаторы позволяют использовать другие параметры в качестве признака фильтрации — например, VLAN ID. Технология виртуальных локальных сетей позволяет создавать группы хостов, трафик которых полностью изолирован от других узлов сети.

При реализации политики безопасности в рамках корпоративной сети, основу которых составляют управляемые коммутаторы, они могут быть мощным и достаточно дешёвым решением. Взаимодействуя только с протоколами канального уровня, такие межсетевые экраны фильтруют трафик с очень высокой скоростью. Основным недостатком такого решения является невозможность анализа протоколов более высоких уровней.

1.4.2. Пакетные фильтры

Пакетные фильтры функционируют на сетевом уровне и контролируют прохождение трафика на основе информации, содержащейся в заголовке пакетов. Многие межсетевые экраны данного типа могут оперировать заголовками протоколов и более высокого, транспортного, уровня (например, TCP или UDP). Пакетные фильтры одними из первых появились на рынке межсетевых экранов и по сей день остаются самым распространённым их типом. Данная технология реализована в подавляющем большинстве маршрутизаторов и даже в некоторых коммутаторах.

При анализе заголовка сетевого пакета могут использоваться следующие параметры:

- IP-адреса источника и получателя;
- тип транспортного протокола;
- поля служебных заголовков протоколов сетевого и транспортного уровней;
- порт источника и получателя.

Достаточно часто приходится фильтровать фрагментированные пакеты, что затрудняет определение некоторых атак. Многие сетевые атаки используют данную уязвимость межсетевых экранов, выдавая пакеты, содержащие запрещённые данные, за фрагменты другого, доверенного пакета. Одним из способов борьбы с данным типом атак является конфигурирование межсетевого экрана таким образом, чтобы блокировать фрагментированные пакеты. Некоторые межсетевые экраны могут дефрагментировать пакеты перед пересылкой во внутреннюю сеть, но это требует дополнительных ресурсов самого межсетевого экрана, особенно памяти. Дефрагментация должна использоваться очень обоснованно, иначе такой межсетевой экран легко может сам стать жертвой DoS-атаки.

Пакетные фильтры могут быть реализованы в следующих компонентах сетевой инфраструктуры:

- пограничные маршрутизаторы;
- операционные системы;
- персональные межсетевые экраны.

Так как пакетные фильтры обычно проверяют данные только в заголовках сетевого и транспортного уровней, они могут выполнять это достаточно быстро. Поэтому пакетные фильтры, встроенные в пограничные маршрутизаторы, идеальны для размещения на границе с сетью с низкой степенью доверия. Однако в пакетных фильтрах отсутствует возможность анализа протоколов более высоких уровней сетевой модели OSI. Кроме того, пакетные фильтры обычно уязвимы для атак, которые используют подделку сетевого адреса. Такие атаки обычно выполняются для обхода управления доступом, осуществляемого межсетевым экраном.

1.4.3. Шлюзы сеансового уровня

Межсетевой экран сеансового уровня исключает прямое взаимодействие внешних хостов с узлом, расположенным в локальной сети, выступая в качестве посредника (*proxy*), который реагирует на все входящие пакеты и проверяет их допустимость на основании текущей фазы соединения. Шлюз сеансового уровня гарантирует, что ни один сетевой пакет не будет пропущен, если он не принадлежит ранее установленному соединению. Как только приходит запрос на установление соединения, в специальную таблицу помещается соответствующая информация (адреса отправителя и получателя, используемые протоколы сетевого и транспортного уровня, состояние соединения и т. д.).

В случае, если соединение установлено, пакеты, передаваемые в рамках данной сессии, будут просто копироваться в локальную сеть без дополнительной фильтрации. Когда сеанс связи завершается, сведения о нём удаляются из данной таблицы. Поэтому все последующие пакеты, «притворяющиеся» пакетами уже завершённого соединения, отбрасываются.

Так как межсетевой экран данного типа исключает прямое взаимодействие между двумя узлами, шлюз сеансового уровня является единственным связующим элементом между внешней сетью и внутренними ресурсами. Это создаёт видимость того, что на все запросы из внешней сети отвечает шлюз, и делает практически невозможным определение топологии защищаемой сети. Кроме того, так как контакт между узлами устанавливается только при условии его допустимости, шлюз сеансового уровня предотвращает возможность реализации DoS-атаки, присущей пакетным фильтрам.

Несмотря на эффективность этой технологии, она обладает серьёзным недостатком: как и у всех вышеперечисленных классов межсетевых экранов, у шлюзов сеансового уровня отсутствует возможность проверки содержания поля данных, что позволяет злоумышленнику передавать «тройных коней» в защищаемую сеть.

1.4.4. Посредники прикладного уровня

Межсетевые экраны прикладного уровня, также, как и шлюзы сеансового уровня, исключают прямое взаимодействие двух узлов. Однако, функционируя на прикладном уровне, они способны «понимать» контекст передаваемого трафика. Межсетевые экраны, реализующие эту технологию, содержат несколько приложений-посредников (*application proxy*), каждое из которых обслуживает свой прикладной протокол. Такой межсетевой экран способен выявлять в передаваемых сообщениях и блокировать несуществующие или нежелательные последовательности команд, что зачастую означает DoS-атаку, либо запрещать использование некоторых команд (например, FTP PUT, которая даёт возможность пользователю записывать информацию на FTP сервер).

Посредник прикладного уровня может определять тип передаваемой информации. Например, это позволяет заблокировать почтовое сообщение, содержащее исполняемый файл. Другой возможностью меж сетевого экрана данного типа является проверка аргументов входных данных. Например, аргумент имени пользователя длиной в 100 символов либо содержащий бинарные данные является, по крайней мере, подозрительным.

Посредники прикладного уровня способны выполнять аутентификацию пользователя, а также проверять, что SSL-сертификаты подписаны конкретным центром. Межсетевые экраны прикладного уровня доступны для многих протоколов, включая HTTP, FTP, почтовые (SMTP, POP, IMAP), Telnet и другие.

Недостатками данного типа межсетевых экранов являются большие затраты времени и ресурсов на анализ каждого пакета. По этой причине они обычно не подходят для приложений реального времени. Другим недостатком является невозможность автоматического подключения поддержки новых сетевых приложений и протоколов, так как для каждого из них необходим свой агент.

1.4.5. Инспекторы состояния

Каждый из вышеперечисленных типов межсетевых экранов используется для защиты корпоративных сетей и обладает рядом преимуществ. Однако, куда эффективней было бы

собрать все эти преимущества в одном устройстве и получить межсетевой экран, осуществляющий фильтрацию трафика с сетевого по прикладной уровень. Данная идея была реализована в инспекторах состояний, совмещающих в себе высокую производительность и защищённость. Данный класс межсетевых экранов позволяет контролировать:

- каждый передаваемый пакет — на основе таблицы правил;
- каждую сессию — на основе таблицы состояний;
- каждое приложение — на основе разработанных посредников.

Осуществляя фильтрацию трафика по принципу шлюза сеансового уровня, данный класс межсетевых экранов не вмешивается в процесс установления соединения между узлами. Поэтому производительность инспектора состояний заметно выше, чем у посредника прикладного уровня и шлюза сеансового уровня, и сравнима с производительностью пакетных фильтров. Ещё одно достоинство инспекторов состояний — прозрачность для пользователя: для клиентского программного обеспечения не потребуется дополнительная настройка. Данные межсетевые экраны имеют большие возможности расширения. При появлении новой службы или нового протокола прикладного уровня для его поддержки достаточно добавить несколько шаблонов. Однако инспекторам состояний по сравнению с посредниками прикладного уровня свойственна более низкая защищённость.

Термин инспектор состояния (*stateful inspection*), внедрённый компанией *Check Point Software*, полюбился производителям сетевого оборудования настолько, что сейчас практически каждый межсетевой экран причисляют к этой технологии, даже если он и не реализует её полностью.

1.5. Реализация межсетевых экранов

Существует два варианта исполнения межсетевых экранов — программный и программно-аппаратный. В свою очередь программно-аппаратный вариант имеет две разновидности — в виде отдельного модуля в коммутаторе или маршрутизаторе и в виде специализированного устройства.

В настоящее время чаще используется программное решение, которое на первый взгляд выглядит более привлекательным. Это вызвано тем, что для его применения достаточно, казалось бы, всего лишь приобрести программное обеспечение межсетевого экрана и установить на любой имеющийся в организации компьютер. Однако, как показывает практика, в организации далеко не всегда находится свободный компьютер, да ещё и удовлетворяющий достаточно высоким требованиям по системным ресурсам. После того, как компьютер всё-таки найден, следует процесс установки и настройки операционной системы, а также программного обеспечения межсетевого экрана. Нетрудно заметить, что использование обычного персонального компьютера далеко не так просто, как может показаться. Именно поэтому всё большее распространение стали получать специализированные программно-аппаратные комплексы, называемые *security appliance*, на основе, как правило, FreeBSD или Linux, «урезанные» для выполнения только необходимых функций. Достоинствами данных решений являются:

- Простота внедрения: данные устройства имеют предустановленную и настроенную операционную систему и требуют минимум настроек после внедрения в сеть.
- Простота управления: данными устройствами можно управлять откуда угодно по стандартным протоколам, таким как SNMP или Telnet, либо посредством защищённых протоколов, таких как SSH или SSL.

- Производительность: данные устройства работают более эффективно, так как из их операционной системы исключены все неиспользуемые сервисы.
- Отказоустойчивость и высокая доступность: данные устройства созданы выполнять конкретные задачи с высокой доступностью.

1.6. Пример настройки конфигурации

Рассмотрим приведённую на рис.1 топологию сети. В данной схеме мы имеем несколько компьютеров в офисе, коммутатор, предназначенный для организации локальной сети, канал доступа в Интернет со статическим IP-адресом и межсетевой экран Cisco ASA 5500. Требуется обеспечить доступ в Интернет компьютерам локальной сети.

Работа с оборудованием производится через *VNC-клиент*, однако существует возможность использования Telnet-клиентов. Информацию об использовании данного способа можно легко найти в Интернете.

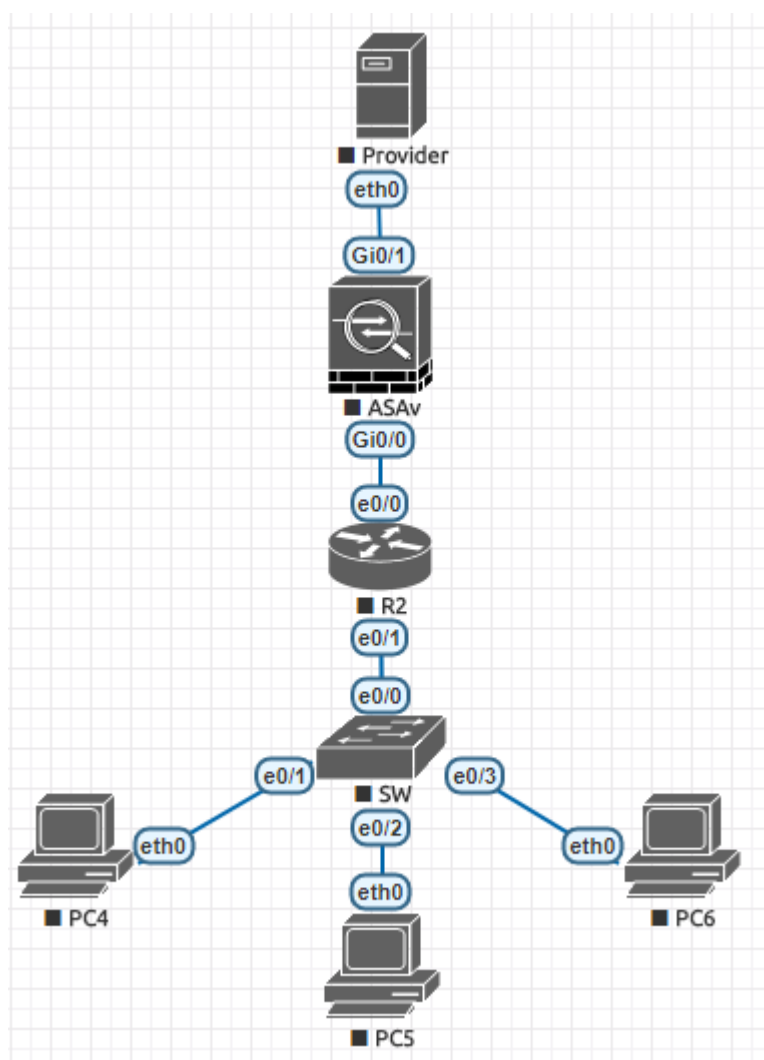


Рисунок 1 – Топология сети.

- Для начала очистим конфигурацию оборудования и зададим имя устройства:

```

ciscoasa> enable
Password: /Enter/
ciscoasa#
ciscoasa(config)# clear configure all

```

```
ciscoasa# hostname FIREWALL
FIREWALL#
```

- Далее нам необходимо настроить внешний и внутренний интерфейсы: через внешний интерфейс будет производится связь с Интернетом, а внутренний настроим для локальной сети. После настройки привяжем VLAN к соответствующим интерфейсам Ethernet.

```
FIREWALL(config)#
interface Vlan1
nameif outside
security-level 0
ip address 200.150.100.2 255.255.255.252
no sh
```

```
FIREWALL(config)#
interface Vlan2
nameif inside
security-level 100
ip address 192.168.10.1 255.255.255.0
no sh
```

```
FIREWALL(config)#
interface Ethernet0/0
switchport access vlan 1
no sh
```

```
FIREWALL(config)#
interface Ethernet0/1
switchport access vlan 2
no sh
```

- Настроим удалённый доступ сетевого администратора, как будто взаимодействуя с реальным оборудованием. Администратор должен иметь доступ к ASA для его настройки.

```
FIREWALL(config)#
enable password /пароль/
FIREWALL(config)#
username admin password /логин/ privilege 15
FIREWALL(config)#
aaa authentication ssh console LOCAL
crypto key generate rsa modulus 1024
FIREWALL(config)#
ssh 192.168.10.100 255.255.255.255 inside /ip-адрес компьютера администратора/
```

- Зададим шлюз по умолчанию (адрес провайдера доступа в Интернет).

```
FIREWALL(config)#
route outside 0.0.0.0 0.0.0.0 200.150.100.1
```

- Настроим NAT и PAT. Для доступа из Интернета на внутренний сервер организации настроим Static NAT.

```
FIREWALL(config)#
object network NATLAN
subnet 192.168.10.0 255.255.255.0
nat (inside,outside) dynamic interface
FIREWALL(config)#
host object network NATSERVER
host 192.168.10.200
```



```
nat (inside,outside) static interface
```

Таким образом, запросы на внешний интерфейс ASA будут направляться по адресу 192.168.10.200.

- Теперь необходимо настроить правила доступа. Пользователи должны иметь доступ в Интернет для просмотра сайтов, сетевой администратор имеет доступ без ограничений, а один из сотрудников (например, бухгалтер) должен иметь доступ к некоторому ресурсу (продолжая тему – к закрытому ресурсу ФНС для предоставления отчётности организаций). Для удобства разделим access-list на группы.

LAN – все пользователи и устройства локальной сети.

ADMIN – компьютер сетевого администратора.

BUH – компьютер бухгалтера.

FNS — адрес ресурса ФНС.

SERVICE – группа порты для веб-доступа.

HDNS – внешний DNS-сервер

SDNS – порты для доступа к службам DNS.

```
FIREWALL(config)#  
object-group network LAN  
network-object 192.168.10.0 255.255.255.0
```

```
object-group network ADMIN  
network-object host 192.168.10.100
```

```
object-group network FNS  
network-object host 1.1.1.1  
object-group network HDNS  
network-object host 8.8.8.8
```

```
object-group service SERVICE  
service-object tcp eq http  
service-object tcp eq https
```

```
object-group service SDNS  
service-object tcp eq 53  
service-object udp eq 53
```

- Настроим правила доступа для созданных групп:

```
FIREWALL(config)#  
access-list LISTIN extended permit ip object-group ADMIN any  
/полный доступ администратору/
```

```
FIREWALL(config)#  
access-list LISTIN extended permit tcp object-group BUH object-group FNS eq  
9443  
/доступ бухгалтера к ресурсу ФНС/
```

```
FIREWALL(config)#  
access-list LISTIN extended permit object-group SERVICE object-group LAN any  
/доступ в Интернет для устройств в локальной сети по портам TCP 80 (http) и  
TCP 443 (https)/
```