Принципы коммутации и маршрутизации

Описание

Базовые принципы коммутации и маршрутизации, базовый подход к траблшутингу и обслуживанию сетей

Оглавление

1. Нужно знать: про маршрутизацию и коммутацию

Сетевая индустрия использует множество терминов и понятий для описания коммутации и маршрутизации, потому что многие термины пересекаются в определениях этих понятий. Это может сбить с толку. Работает ли маршрутизатор маршрутизатором или коммутатором? В чем разница между коммутацией на 3 уровне (L3) и маршрутизацией?

Что бы найти ответы на эти вопросы необходимо разобраться, что происходит с пакетом, когда он проходит через сеть.

ПОНИМАНИЕ ШИРОКОВЕЩАТЕЛЬНЫХ И КОЛЛИЗИОННЫХ ДОМЕНОВ

Два основных понятия, которые вы должны понять.

КОММУТАЦИЯ. ПОНЯТИЕ ШИРОКОВЕЩАТЕЛЬНОГО ДОМЕНА И ДОМЕНА КОЛЛИЗИЙ

На рисунке изображена простая сеть, иллюстрирующая эти два понятия.

Домен коллизий определяется как набор хостов, подключенных к сети. В некоторых случаях хосты одновременно не будут передавать пакеты из-за возможного столкновения последних.

Например, если Хост А и хост Б соединены прямым проводом, то они не смогут передавать пакеты одновременно. Однако, если между хостами установлено какое-то физическое устройств, то одновременная передача данных возможна, так как они находятся в отдельных доменах коллизий.

Широковещательный домен-это набор хостов, которые могут обмениваться данными, просто отправляя данные на 2 уровне(L2). Если узел А посылает широковещательный пакет для всех хостов, по локальной сети, и хост В получает его, эти два хоста находятся в одном широковещательном домене.

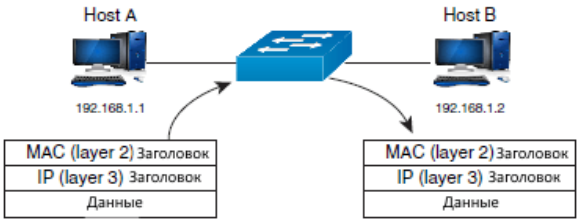
ШИРОКОВЕЩАТЕЛЬНЫЙ ДОМЕН И ДОМЕН КОЛЛИЗИЙ

Мостовое соединение создает домен коллизий, но не широковещательный домен.

Традиционная коммутация пакетов и мостовое соединение- технически- это одно и то же. Основное различие заключается в том, что в большинстве коммутируемых сред каждое устройство, подключенное к сети, находится в отдельном домене коллизий.

Что же изменяется в формате типичного пакета, когда он проходит через коммутатор?

Рисунок не показывает измения в формате пакетов данных прошедших через коммутатор



Вообще, устройства по обе стороны от коммутатора не "видят", что между ними есть коммутатор, они также не знают назначения своих пакетов; коммутаторы прозрачны для устройств подключенных к сети.

Если узел А хочет отправить пакет на ip-адресс 192.168.1.2 (узел В), он отправляет в эфир широковещательный запрос для всех узлов, подключенных к тому же сегменту сети, запрашивает MAC-адрес хоста с IP-адресом 192.168.1.2 (это называется *Address Resolution Protocol* (ARP)). Так как узел В находится в том же широковещательном домене, что и узел А, узел А может быть уверен, что узел В получит этот широковещательный запрос и отправит ответный пакет с верным MAC-адресом для обмена пакетами.

ШИРОКОВЕЩАТЕЛЬНЫЕ ДОМЕНЫ И ДОМЕНЫ КОЛЛИЗИИ В МАРШРУТИЗАЦИИ

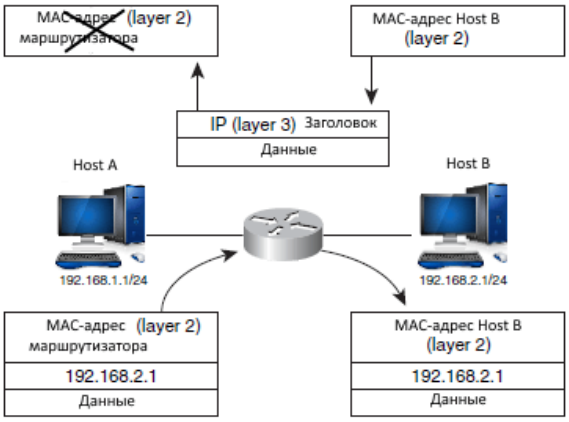
Сеть построена на основе маршрутизатора не создает широковещательный домен и домен коллизий данная схема приведена на рисунке:

Возникает вопрос, как пакет отправленный с хоста А достигнет хост Б с ip-адресом 192.168.2.1? Хост А не может отправить широковещательный пакет для обнаружения адреса узла В, поэтому он должен использовать какой-то другой метод чтобы выяснить, как добраться до этого пункта назначения. Откуда узел А знает об этом? Обратите внимание, что после каждого IP-адреса на рисунке выше, есть значение */ 24*. Это число указывает длину префикса, или количество битов, установленных в маске подсети. Хост А может использовать эту информацию для определения что

хост B не находится в том же широковещательном домене (не в том же сегменте), и хост A должен использовать определенный метод маршрутизации для достижения цели, как показано на рисунке ниже.

Теперь, когда хост A знает, что хост B не находится в том же широковещательном домене, что и он, он не может отправить широковещательный запрос для получения адреса хоста B. Как, тогда, пакету, отправленному с узла A, добраться до узла B?

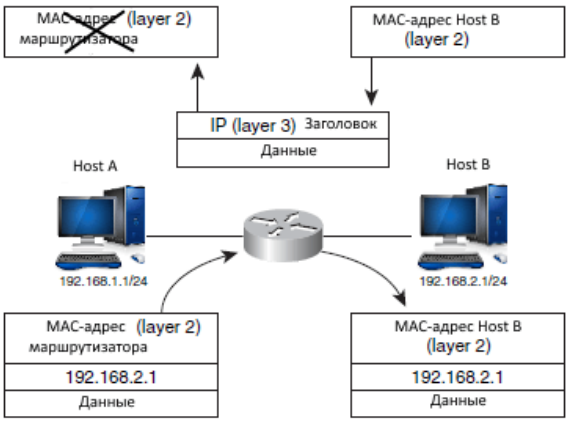
Отправляя свои пакеты к промежуточному маршрутизатору, Хост A помещает в заголовок пакета IP-адрес хоста B, а также еще MAC-адрес промежуточного маршрутизатора, как показано на рисунке.



Узел A помещает MAC-адрес маршрутизатора в заголовок пакета. Маршрутизатор принимает этот пакет, приходящий из сети. Далее маршрутизатор проверяет IP-адрес назначения и определяет, какой наиболее короткий маршрут построить и сравнивает данные из пакета с таблицей маршрутизации (в данном случае сравниваются данные хоста B), и заменяет MAC-адрес правильным MAC-адресом для следующего перехода. Затем маршрутизатор пересылает пакет в другой сегмент, который находится в другом широковещательном домене.

КОММУТАЦИЯ L3

Коммутация 3 уровня очень похожа на маршрутизацию, как показано на рисунке ниже (обратите внимание, что это то же самое, что изображено на рисунке выше). Это связано с тем, что коммутация 3 уровня является маршрутизируемой; Нет никакой функциональной разницы между коммутацией 3 уровня и маршрутизацией.


MAC-адрес (layer 2)
маршрутизатора
MAC-адрес Host B
(layer 2)
IP (layer 3) Заголовок
Данные
Host A
Host B
192.168.1.1/24
192.168.2.1/24
MAC-адрес (layer 2)
маршрутизатора
192.168.2.1
Данные
MAC-адрес Host B
(layer 2)
192.168.2.1
Данные

2. Базовая настройка коммутатора Cisco

Сетевые устройства могут работать в режимах, которые подразделяются на три большие категории.

Первая и основная категория- это передача данных (плоскость данных, data plane). Это режим работы коммутатора по передаче кадров, генерируемых устройствами, подключенными к коммутатору. Другими словами, передача данных является основным режимом работы коммутатора.

Во-вторых, управление передачей данных относится к настройкам и процессам, которые управляют и изменяют выбор, сделанный передающим уровнем коммутатора. Системный администратор может контролировать, какие интерфейсы включены и отключены, какие порты работают с какой скоростью, как связующее дерево блокирует некоторые порты, чтобы предотвратить циклы, и так далее. Так же важной частью этой статьи является управление устройством, осуществляемое через плоскость наблюдения (management plane). Плоскость наблюдения - это управление самим устройством, а не управление тем, что делает устройство. В частности, в этой статье рассматриваются самые основные функции управления, которые могут быть настроены в коммутаторе Cisco. В первом разделе статьи рассматривается настройки различных видов безопасности входа в систему. Во втором разделе показано, как настроить параметры IPv4 на коммутаторе, чтобы им можно было управлять удаленно. В последнем разделе рассматриваются практические вопросов, которые могут немного облегчить жизнь системного администратора.

Защита коммутатора через CLI

По умолчанию коммутатор Cisco Catalyst позволяет любому пользователю подключиться к консольному порту, получить доступ к пользовательскому режиму, а затем перейти в привилегированный режим без какой-либо защиты. Эти настройки заданы в сетевых устройствах Cisco по умолчанию и, если у вас есть физический доступ к устройству, то вы спокойно можете подключиться к устройству через консольный порт или USB, используя соответствующий кабель и соответственно производить различные настройки.

Однако не всегда имеется физический доступ к коммутатору и тогда необходимо иметь доступ к устройствам для удаленного управления, и первым шагом в этом процессе является обеспечение безопасности коммутатора так, чтобы только соответствующие пользователи могли получить доступ к интерфейсу командной строки коммутатора (CLI).

Настройка парольного доступа к коммутатору Cisco

В данной части рассматривается настройка безопасности входа для коммутатора Cisco Catalyst.

Защита CLI включает защиту доступа в привилегированный режим, поскольку из этого режима злоумышленник может перезагрузить коммутатор или изменить конфигурацию.

Защита пользовательского режима также важна, поскольку злоумышленники могут видеть настройки коммутатора, получить настройки сети и находить новые способы атаки на сеть.

Особенно важно, что бы все протоколы удаленного доступа и управления, чтобы IP-настройки коммутатора были настроены и работали.

Для того чтобы получить удаленный доступ по протоколам Telnet и Secure Shell (SSH) к коммутатору, необходимо на коммутаторе настроить IP-адресацию.

Чуть позже будет показано, как настроить IPv4-адресацию на коммутаторе.

В первой части статьи будут рассмотрены следующие вопросы защиты входа:

- Защита пользовательского режима и привилегированного режима с помощью простых паролей;
- Защита доступа в пользовательский режим с использованием локальной базы данных;
- Защита доступа в пользовательский режим с помощью внешних серверов аутентификации;
- Защита удаленного доступа с помощью Secure Shell (SSH);

Защита пользовательского и привилегированного режима с помощью простых паролей.

Получить полный доступ к коммутатору Cisco можно только через консольный порт.

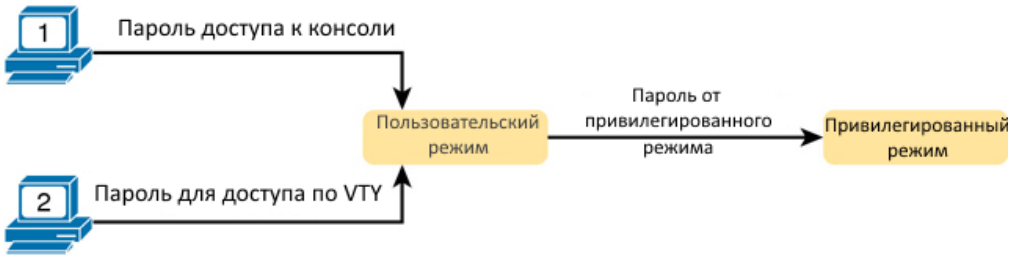
В этом случае, настройки по умолчанию, позволяют получить доступ сначала к режиму пользователя, а затем можно перейти в привилегированный режим без использования паролей.

А вот по протоколам удаленного доступа Telnet или SSH получить доступ даже к режиму пользователя невозможно.

Настройки по умолчанию идут у совершенно нового коммутатора, но в производственной среде необходимо обеспечить безопасный доступ через консоль, а также включить удаленный вход через Telnet и/или SSH, чтобы была возможность подключаться ко всем коммутаторам в локальной сети.

Можно организовать доступ к сетевому оборудованию с использованием одного общего пароля.

Этот метод позволяет подключиться к оборудованию, используя только пароль - без ввода имени пользователя - с одним паролем для входа через консольный порт и другим паролем для входа по протоколу Telnet. Пользователи, подключающиеся через консольный порт, должны ввести пароль консоли, который был предварительно настроен в режиме конфигурации. Пользователи, подключающиеся через протокол Telnet, должны ввести пароль от Telnet, также называемый паролем vty, так называемый, потому что это режим конфигурации терминальных линий (vty). На рисунке 1 представлены варианты использования паролей с точки зрения пользователя, подключающегося к коммутатору.



Как видно из рисунка 1, на коммутаторах Cisco стоит защита привилегированного режима (enable) с помощью еще одного общего пароля, задаваемый командой enable password. Системный администратор, подключающийся к CLI коммутатора попадает в режим пользователя и далее, вводит команду enable.

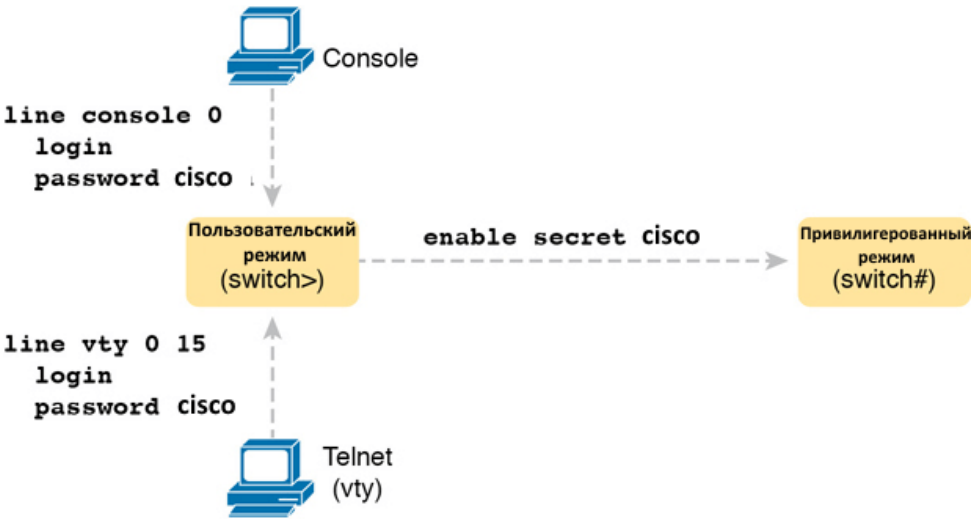
Эта команда запрашивает у пользователя пароль входа в привилегированный режим; если пользователь вводит правильный пароль, IOS перемещает пользователя в привилегированный режим.

Пример 1. Пример входа в коммутатор из консоли, когда пароль консоли и пароль привилегированного режима были заранее установлены. Предварительно пользователь запустил эмулятор терминала, физически подключил ноутбук к консольному кабелю, а затем нажал клавишу Enter, чтобы войти в коммутатор.

```
(User now presses enter now to start the process. This line of text does not appear.)
User Access Verification
Password: cisco
Switch> enable
Password: cisco
Switch#
```

В примере показаны пароли в открытом виде, как если бы они были набраны в обычном текстовом редакторе (cisco), а также команда enable, которая перемещает пользователя из пользовательского режима в привилегированный режим (enable). В реальности же IOS скрывает пароли при вводе, чтобы никто не смог увидеть их.

Чтобы настроить общие пароли для консоли, Telnet и привилегированного режима (enable), необходимо ввести несколько команд. На рис. 2 показан порядок задания всех трех паролей.



На рисунке 2 показаны два ПК, пытающиеся получить доступ к режиму управления устройством. Один из ПК подключен посредством консольного кабеля, соединяющейся через линию console 0, а другой посредством Telnet, соединяющейся через терминальную линию vty 0 15. Оба компьютера не имеют Логинов, пароль для консоли и Telnet -cisco. Пользовательский режим получает доступ к привилегированному режиму (enable) с помощью ввода команды "enable secret cisco". Для настройки этих паролей не надо прилагать много усилий. Все делается легко. Во-первых, конфигурация консоли и пароля vty устанавливает пароль на основе контекста: для консоли (строка con 0) и для линий vty для пароля Telnet (строка vty 0 15). Затем в режиме консоли и режиме vty, соответственно вводим команды:

```
login
password <пароль задаваемый пользователем>
```

Настроенный пароль привилегированного режима, показанный в правой части рисунка, применяется ко всем пользователям, независимо от того, подключаются ли они к пользовательскому режиму через консоль, Telnet или иным образом. Команда для настройки *enable password* является командой глобальной конфигурации: *enable secret <пароль пользователя>*.

В старых версиях, для задания пароля на привилегированный режим, использовалась команда *password*. В современных IOS применяется два режима задания пароля: *password* и *secret*.

Рекомендуется использовать команду *secret*, так как она наиболее безопасна по сравнению с *password*.

Для правильной настройки защиты коммутатора Cisco паролями необходимо следовать по шагам, указанным ниже:

Шаг 1. Задайте пароль на привилегированный режим командой *enable secret password-value*

Шаг 2. Задайте пароль на доступ по консоли

1. Используйте команду *line con 0* для входа режим конфигурирования консоли;
2. Используйте команду *password password-value* для задания пароля на консольный режим;
3. Используйте команду *login* для запроса пароля при входе по консоли;

Шаг 3. Задайте пароль на терминальные подключения vty (Telnet)

1. Используйте команду *line vty 0 15* для входа режим конфигурирования терминальных линий. В данном примере настройки будут применены ко всем 16 терминальным линиям;
2. Используйте команду *password password-value* для задания пароля на режим vty;
3. Используйте команду *login* для запроса пароля при входе по Telnet

В Примере 2 показан процесс настройки, согласно вышеописанным шагам, а также установка пароля *enable secret*. Строки, которые начинаются с ! - это строки комментариев. Они предназначены для комментирования назначения команд.

```
! Enter global configuration mode, set the enable password, and also set the hostname (just
because it makes sense to do so)
Switch# configure terminal
Switch(config)# enable secret cisco
Switch#(config)# line console 0
Switch#(config-line)# password cisco
Switch#(config-line)# login
Switch#(config-line)# exit
Switch#(config)# line vty 0 15
Switch#(config-line)# password cisco
Switch#(config-line)# login
Switch#(config-line)# end
Switch#
```

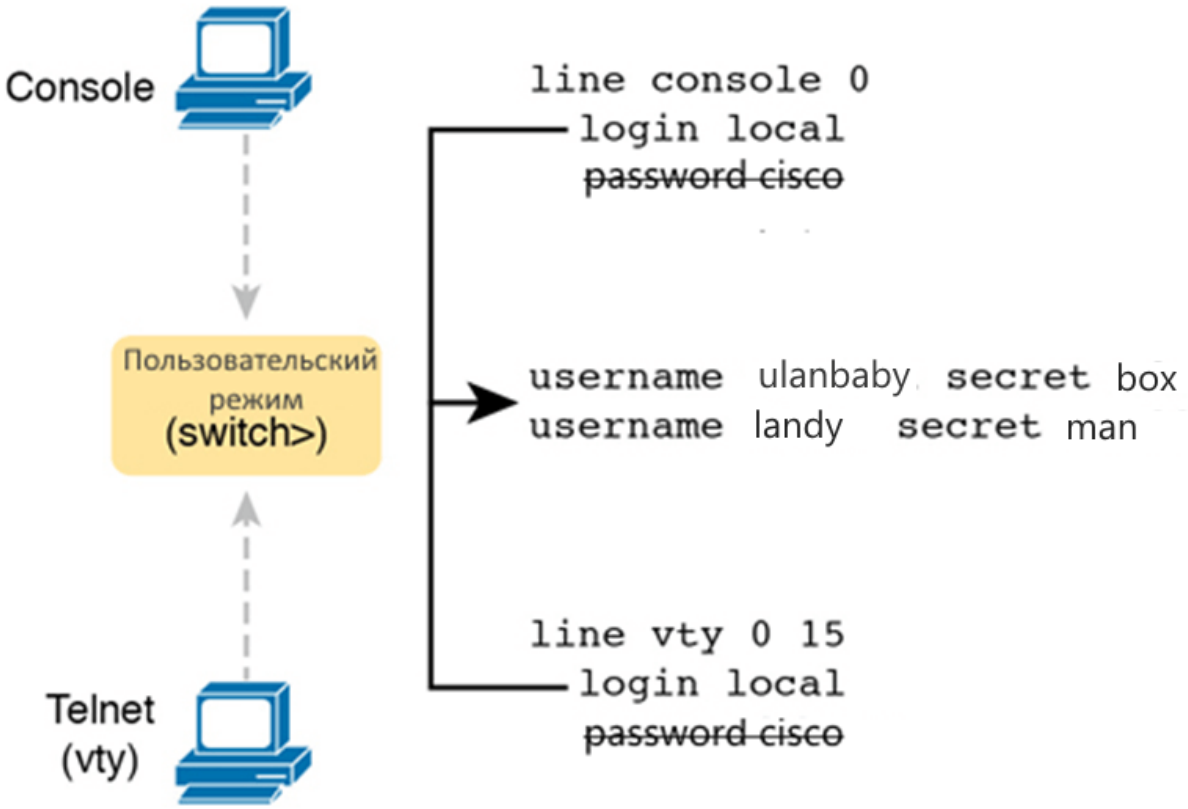
Пример 3 показывает результирующую конфигурацию в коммутаторе, выводимой командой *show running-config*. Выделенный текст показывает новую конфигурацию. Часть листинга было удалено, что бы сконцентрировать ваше внимание на настройке пароля.

```
Switch# show running-config
!
Building configuration...
Current configuration: 1333 bytes
!
version 12.2
!
enable secret 5 $1$OwtI$A58c2XgqWyDNeDnv51mNR.
!
interface FastEthernet0/1
!
interface FastEthernet0/2
!
! Several lines have been omitted here - in particular, lines for
! FastEthernet interfaces 0/3 through 0/23.
!
interface FastEthernet0/24
!
interface GigabitEthernet0/1
!
interface GigabitEthernet0/2
!
line con 0
password cisco
login
!
line vty 0 4
password cisco
login
!
line vty 5 15
password cisco
login
```

Защита доступа в пользовательском режиме с помощью локальных имен пользователей и паролей

Коммутаторы Cisco поддерживают два других метода безопасного входа, которые используют пары имя пользователя / пароль вместо общего пароля без ввода имени пользователя. Первый метод, использует ввод локального имени пользователя и пароля. Происходит настройка пары имя пользователя / пароль локально-то есть в конфигурации коммутатора. Коммутаторы поддерживают режим локального имени пользователя / пароля для входа по консоли, по Telnet и даже по SSH, но не изменяют пароль от привилегированного режима (enable), используемый для входа в режим enable.

Настройки для перехода от использования простых общих паролей к использованию локальных имен пользователей/паролей требует лишь небольших изменений конфигурации, как показано на рис.3.



На рисунке показаны два ПК, пытающиеся получить доступ к пользовательскому режиму. Один из ПК подключен по консольному кабелю в пользовательский режим через линию console 0, а другой ПК по Telnet, соединяющийся через терминальные линии vty 0 15. Оба ПК не имеют паролей для входа, и задано имя пользователя для обоих ПК - " local."

На рисунке в Пользовательском режиме используется две команды:

- 1- username ulanbaby secret box
- 2- username landy secret box

Глядя на настройки на рисунке, видно, во-первых, коммутатору, необходимо задать *пару имя пользователя/пароль*. Для их создания, в режиме глобальной конфигурации, введите команду создания имени пользователя и зашифрованного пароля -username <имя пользователя> secret

<пароль>. Затем, чтобы включить тип безопасности входа с проверкой логина (имени пользователя) по консоли или Telnet, просто добавьте команду *login local*. По сути, эта команда означает " использовать локальный список имен пользователей для входа в систему."

Вы также можете использовать команду no password, чтобы очистить все оставшиеся команды паролей из консоли или режима vty, потому что эти команды не нужны при использовании локальных имен пользователей и паролей.

Ниже подробно описаны шаги для настройки доступа к к коммутатору с использованием логина и пароля:

Шаг 1. В режиме глобальной конфигурации используйте команду username <имя пользователя > secret <пароль>, чтобы создать одну или несколько пар имя пользователя/пароль в локальной базе коммутатора.

Шаг 2. Настройте консоль на использование пар имя пользователя / пароль из локальной базы коммутатора:

1. используйте команду line con 0 для входа в режим конфигурации консоли.
2. используйте подкоманду login local, чтобы разрешить коммутатору запрашивать имя пользователя и пароль, совпадающие со списком локальных имен пользователей/паролей.
3. (необязательно) используйте подкоманду no password для удаления всех существующих простых общих паролей, просто для оптимизации конфигурации.

Шаг 3. Настройте Telnet (vty) для использования пар имя пользователя / пароль из локальной базы коммутатора:

1. 1. используйте команду line vty 0 15 для входа в режим конфигурации vty для всех 16 терминальных линий vty (пронумерованных от 0 до 15).
2. 2. используйте подкоманду login local, чтобы разрешить коммутатору запрашивать имя пользователя и пароль для всех входящих пользователей Telnet, со списком локальных имен пользователей/паролей.
3. 3. (необязательно) используйте подкоманду no password для удаления всех существующих простых общих паролей, просто для оптимизации конфигурации.

При попытке подключиться по Telnet к коммутатору, настроенному как показано на рисунке, пользователю будет предложено сначала ввести имя пользователя, а затем пароль, как показано в Примере 4. Пара имя пользователя / пароль должна быть в локальной базе коммутатора.В противном случае вход в систему будет отклонен.

```
SW2# telnet 172.18.2.19

Trying 172.18.2.19 ... Open

User Access Verification

Username: ulanbaby
Password:
SW1> enable
Password:
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW1(config)#^Z
SW1#
*Mar 1 02:00:56.229: %SYS-5-CONFIG_I: Configured from console
by ulanbaby on vty0 (172.18.2.19)
```

В примере 4 коммутаторы Cisco не отображает символы при вводе пароля по соображениям безопасности.

Защита доступа в пользовательском режиме с помощью внешних серверов аутентификации

В конце примера 4 показано одно из многочисленных улучшений безопасности, когда требуется, чтобы каждый пользователь входил под своим собственным именем пользователя. Также в конце примера показано, как пользователь входит в режим конфигурации (configure terminal), а затем сразу же покидает его (end). Обратите внимание, что при выходе пользователя из режима конфигурации коммутатор генерирует сообщение журнала (log). Если пользователь вошел в систему с именем пользователя, сообщение журнала (log) идентифицирует это имя пользователя; В примере сгенерировано сообщение журнала по имени "ulanbaby".

Однако использование имени пользователя / пароля, настроенного непосредственно на коммутаторе, не всегда удобно при администрировании. Например, каждому коммутатору и маршрутизатору требуется настройка для всех пользователей, которым может потребоваться войти на устройства. Затем, когда возникнет необходимость внесения изменений в настройки, например, изменение паролей для усиления безопасности, настройки всех устройств должны быть изменены.

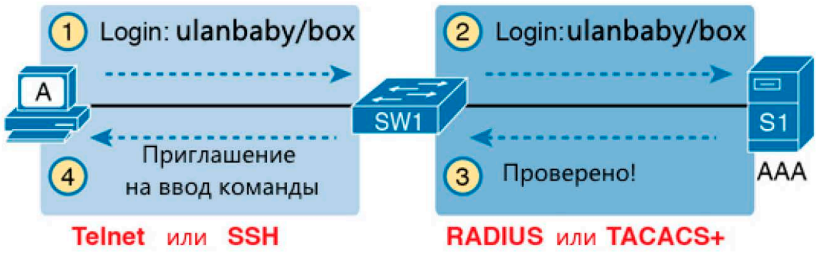
Лучшим вариантом было бы использовать инструменты, подобные тем, которые используются для многих других функций входа в ИТ. Эти инструменты обеспечивают центральное место для безопасного хранения всех пар имя пользователя / пароль, с инструментами, чтобы заставить пользователей регулярно менять свои пароли, инструменты, чтобы отключать пользователей, когда они завершают сеанс работы, и так далее.

Коммутаторы Cisco позволяют использовать внешний сервер, называемый сервером аутентификации, авторизации и учета (authentication, authorization, and accounting)(AAA). Эти серверы содержат имена пользователей / пароли. Сегодня многие существующие сети используют AAA-серверы для входа на коммутаторы и маршрутизаторы.

Для настройки данного входа по паре имя пользователя / пароль необходимо произвести дополнительные настройки коммутатора.

При использовании AAA-сервера для аутентификации коммутатор (или маршрутизатор) просто отправляет сообщение на AAA-сервер, спрашивая, разрешены ли имя пользователя и пароль, и AAA-сервер отвечает.

На рисунке показано, что пользователь сначала вводит имя пользователя / пароль, коммутатор запрашивает AAA-сервер, а сервер отвечает коммутатору, заявляя, что имя пользователя/пароль действительны.



На рисунке процесс начинается с того, что ПК " A " отправляет регистрационную информацию через Telnet или SSH на коммутатор SW1. Коммутатор передает полученную информацию на сервер "AAA" через RADIUS или TACACS+. Сервер отправляет подтверждение коммутатору, который, в свою очередь, отправляет приглашение (разрешение) на ввод команды в пользовательскую систему.

Хотя на рисунке показана общая идея, обратите внимание, что информация поступает с помощью нескольких различных протоколов. Слева, соединение между Пользователем и коммутатором или маршрутизатором использует Telnet или SSH. Справа коммутатор и AAA-сервер обычно используют протокол RADIUS или TACACS+, оба из которых шифруют пароли, при передаче данных по сети.

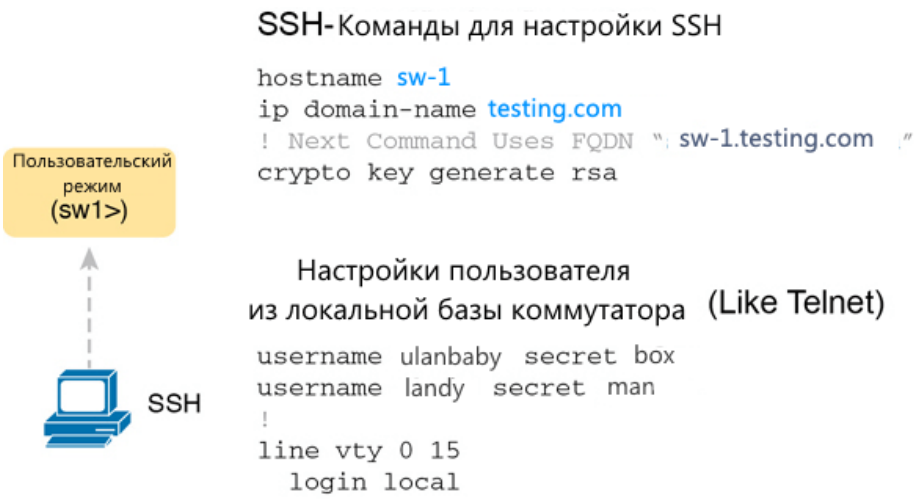
Настройка защищенного удаленного доступа по SSHl

До сих пор мы рассматривали доступ к коммутатору по консоли и Telnet, в основном игнорируя SSH. У Telnet есть один серьезный недостаток: все данные в сеансе Telnet передаются в открытом виде, включая обмен паролями. Таким образом, любой, кто может перехватывать сообщения между Пользователем и коммутатором (man-in-the-middle attack), может видеть пароли. SSH шифрует все данные, передаваемые между SSH-клиентом и сервером, защищая данные и пароли.

SSH может использовать тот же метод аутентификации локального входа, что и Telnet, с настроенными именем пользователя и паролем в локальной базе коммутатора. (SSH не работает с методами аутентификации, которые не используют имя пользователя, например только общие пароли.)

Итак, в настройке доступа для локальных пользователей по Telnet, как показано ранее на рисунке, также включена локальная аутентификация по имени пользователя для входящих соединений SSH.

На рисунке показан один пример настройки того, что требуется для поддержки SSH. Рисунок повторяет конфигурацию создания локального пользователя, (см. рисунок) для подключения по Telnet. На скриншоте показаны три дополнительные команды, необходимые для завершения настройки SSH на коммутаторе.



На рисунке показаны три дополнительные команды, необходимые для завершения настройки SSH на коммутаторе. На рисунке показан листинг настройки SSH. Для настройки SSH на рисунке, отображаются команды:

1. hostname sw-1 (задает имя коммутатору)
2. ip domain-name testing.com (команда использует полное доменное имя sw-1.testing.com)
3. crypto key generate rsa.

Для локальной конфигурации имени пользователя (например, Telnet) отображаются следующие команд:

```
username ulanbaby secret box
username landy secret man
line vty 0 15
login local
```

IOS использует три команды: две для конфигурации SSH, а также одну команду для создания ключей шифрования SSH. Сервер SSH использует полное доменное имя коммутатора в качестве входных данных для создания этого ключа. Коммутатор создает полное доменное имя из имени хоста и доменного имени коммутатора. Рисунок 5 начинается с установки обоих значений (на тот случай, если они еще не настроены). Затем третья команда, команда crypto key generate rsa, генерирует ключи шифрования SSH. IOS по умолчанию использует SSH-сервер. Кроме того, IOS по умолчанию разрешает SSH-соединения по vty.

Просмотр настроек в режиме конфигурации, шаг за шагом, может быть особенно полезен при настройке SSH. Обратите внимание, в частности, что в этом примере команда crypto key запрашивает у пользователя модуль ключа; вы также можете добавить параметр modulus modulus-

value в конец команды crypto key, чтобы добавить этот параметр в команду. В примере 5 показан порядок настройки ssh (такие же команды, что и на рис. 5) Ключ шифрования является последним шагом.

```
SW1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
! Step 1 next. The hostname is already set, but it is repeated
just
! to be obvious about the steps.

SW1(config)# hostname SW-1
SW1(config)# ip domain-name testing.com
SW1(config)# crypto key generate rsa
The name for the keys will be: SW-1.testing.com
Choose the size of the key modulus in the range of 360 to 2048
for your
  General Purpose Keys. Choosing a key modulus greater than 512
may take
  a few minutes.
How many bits in the modulus [512]: 1024
% Generating 1024 bit RSA keys, keys will be non-exportable...
[OK] (elapsed time was 4 seconds)
SW1(config)#
!! Optionally, set the SSH version to version 2 (only) -
preferred
SW1(config)# ip ssh version 2
! Next, configure the vty lines for local username support,
!just like with Telnet
!
SW1(config)# line vty 0 15
SW1(config-line)# login local
SW1(config-line)# exit
!
! Define the local usernames, just like with Telnet
!
SW1(config)# username ulanbaby password box
SW1(config)# username landy password man
SW1(config)# ^Z
SW1#
```

Ранее упоминалось, что одним полезным значением по умолчанию было то, что коммутатор по умолчанию поддерживает как SSH, так и Telnet на линиях vty. Однако, поскольку Telnet не безопасный протокол передачи данных, то вы можете отключить Telnet, чтобы обеспечить более жесткую политику безопасности.

Для управления тем, какие протоколы коммутатор поддерживает на своих линиях vty, используйте подкоманду transport input {all | none / telnet / ssh} vty в режиме vty со следующими опциями:

- **transport input all** or **transport input telnet ssh** поддержка как Telnet, так и SSH
- **transport input none:** не поддерживается ни один протокол
- **transport input telnet:** поддержка только Telnet
- **transport input ssh:** поддержка только SSH

В завершении этой части статьи о SSH, расписана пошаговая инструкция настройки коммутатора Cisco для поддержки SSH с использованием локальных имен пользователей. (Поддержка SSH в IOS может быть настроена несколькими способами; эта пошаговая инструкция показывает один простой способ ее настройки.)

Процесс, показанный здесь, заканчивается инструкцией настройки локального имени пользователя на линиях vty, как было обсуждено ранее в первой части данной серии статей.

Шаг 1. Настройте коммутатор так, чтобы он генерировал совпадающую пару открытых и закрытых ключей для шифрования:

- если еще не настроено, задайте командой **hostname***name* имя для этого коммутатора в режиме глобальной конфигурации.
- Если еще не настроено, задайте командой **ip domain-name***name* доменное имя для коммутатора в режиме глобальной конфигурации.
- Используйте команду crypto key generate rsa в режиме глобальной конфигурации (или команду crypto key generate RSA modulus modulus-value, чтобы избежать запроса модуля ключа) для генерации ключей. (Используйте по крайней мере 768-битный ключ для поддержки SSH версии 2.)

Шаг 2. (Необязательно) используйте команду ip ssh version 2 в режиме глобальной конфигурации, чтобы переопределить значение по умолчанию для поддержки обеих версий протокола удаленного доступа SSH 1 и 2, так что бы разрешены были только соединения SSHv2.

Шаг 3. (Необязательно) если вы еще не настроили нужный параметр, задайте на линии vty для работы по SSH и Telnet.:

- используйте команду transport input ssh в режиме конфигурации линий vty, чтобы разрешить только SSH.
- используйте команду transport input all (по умолчанию) или команду transport input telnet ssh в режиме конфигурации линий vty, чтобы разрешить как SSH, так и Telnet.

Шаг 4. Используйте различные команды в режиме конфигурации линий vty для настройки локальной аутентификации имени пользователя, как описано ранее в этой статье.

На маршрутизаторах Cisco часто по умолчанию настроен параметр transport input none. Поэтому необходимо добавить подкоманду transport input line для включения Telnet и / или SSH в маршрутизаторе.

Для просмотра информации о состояния SSH на коммутаторе используются две команды. Во-первых, команда show ip ssh выводит информацию о состоянии самого SSH-сервера. Затем команда show ssh выводит информацию о каждом клиенте SSH, подключенном в данный момент к коммутатору. В пример 6 показаны примеры работы каждой из команд, причем пользователь ULANBABY в данный момент подключен к коммутатору.

```
SW1# show ip ssh

SSH Enabled - version 2.0
Authentication timeout: 120 secs; Authentication retries: 3

SW1# show ssh
Connection Version Mode  Encryption Hmac             State
Username
0          2.0   IN    aes126-cbc    hmac-sha1       Session started    ulanbaby
0          2.0   OUT   aes126-cbc    hmac-sha1       Session started    ulanbaby
%No SSHv1 server connections running.
```

3. Настройка интерфейсов коммутатора Cisco

IOS использует термин интерфейс для обозначения физических портов, используемых для передачи и приема данных на другие устройства в сети. Каждый интерфейс может иметь несколько различных настроек, каждая из которых может отличаться от интерфейса к интерфейсу. В IOS для настройки этих параметров используются подкоманды (subcommands) в режиме пользовательского интерфейса. Для каждого интерфейса настраиваются свои параметры. Соответственно, сначала необходимо определить интерфейс, на котором будут настраиваться параметры, а затем выполнить настройки этих параметров.

В этой статье рассмотрим три параметра интерфейса: скорость порта, дуплекс и текстовое описания.

Настройка скорости, дуплекса и описания

Интерфейсы коммутатора, поддерживающие несколько скоростей (10/100 и 10/100/1000), по умолчанию будут автоматически определять, какую скорость использовать. Однако вы можете указать параметры скорости и дуплекса с помощью подкоманд *duplex {auto / full / half}* и *speed {auto| 10 | 100 | 1000}*.

В большинстве случаев лучше использовать режим автосогласования (auto).Но существуют такие моменты, когда необходимо вручную изменить скорость и дуплекс. Например, необходимо установить максимально возможную скорость на соединениях между коммутаторами, чтобы избежать вероятности того, что автосогласование выберет более низкую скорость.

Подкоманда *description<текстовое описание>*позволяет добавить текстовое описание к интерфейсу (комментарий). Например, после изменения скорости и дуплекса на порту, можно добавить описание, объясняющее, почему вы это сделали. В примере 1 показан листинг команд для настройки дуплекса, скорости и описание.

```
SW-1(config)# interface FastEthernet 0/1
SW-1(config-if)# duplex full
SW-1(config-if)# speed 100
SW-1(config-if)# description Server on 3rd PC, Preset to 100/full
SW-1 (config-if)# exit
SW-1 (config)# interface range FastEthernet 0/11 - 20
SW-1 (config-if-range)# description user's from BUH
SW-1 (config-if-range)# ^Z
SW-1 #
```

Для начала настройки трех параметров необходимо вспомнить команды позволяющие перейти из пользовательского режима в режим глобальной конфигурации, а так же команды перехода в режим конфигурации и настройки интерфейса. Выше, показан пример использования команд *duplex, speed* и *description*. Данные команды вводятся сразу после команды `interface FastEthernet 0/1`, что означает, что настройки этих трех параметров применяются к интерфейсу Fa0/1, а не к другим интерфейсам.

Команда show interfaces status отображает детальную информацию, настроек произведенных в примере 1:

```
SW-1# show interfaces status

Port      Name                  Status       Vlan       Duplex  Speed Type

Fa0/1     Server on 3rd PC      notconnect   1            full    100 10/100BaseTX
Fa0/2                           notconnect   1            auto   auto 10/100BaseTX
Fa0/3                           notconnect   1            auto   auto 10/100BaseTX
Fa0/4                           connected    1          a-full  a-100 10/100BaseTX
Fa0/5                           notconnect   1            auto   auto 10/100BaseTX
Fa0/6                           connected    1          a-full  a-100 10/100BaseTX
Fa0/7                           notconnect   1            auto   auto 10/100BaseTX
Fa0/8                           notconnect   1            auto   auto 10/100BaseTX
Fa0/9                           notconnect   1            auto   auto 10/100BaseTX
Fa0/10                          notconnect   1            auto   auto 10/100BaseTX
Fa0/11    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/12    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/13    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/14    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/15    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/16    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/17    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/18    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/19    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/20    user from BUH connect notconnect   1            auto   auto 10/100BaseTX
Fa0/21                          notconnect   1            auto   auto 10/100BaseTX
Fa0/22                          notconnect   1            auto   auto 10/100BaseTX
Fa0/23                          notconnect   1            auto   auto 10/100BaseTX
Fa0/24                          notconnect   1            auto   auto 10/100BaseTX
Gi0/1                           notconnect   1            auto   auto 10/100/1000BaseTX
Gi0/2                           notconnect   1            auto   auto 10/100/1000BaseTX
```

Разберем выходные данные из примера:

- **FastEthernet 0/1 (Fa0 / 1):** выведено описание интерфейса (задается командой `description`). Также представлена информация о настройке скорости в 100Mb/s и выставлен режим интерфейса full duplex. В представленной в примере информации есть статус `notconnect` это означает, что интерфейс Fa0 / 1 в настоящее время не подключен (не подключен кабель) и не работает.
- **FastEthernet 0/2 (Fa0 / 2):** данный интерфейс не настраивался. Отображаются настройки по умолчанию. Обратите внимание, на слова "auto" под заголовком speed и duplex это означает, что данный порт автоматически согласовывает обе настройки с портами других устройств. Этот порт также не подключен (не подключен кабель).

- **FastEthernet 0/4 (Fa0 / 4):** Как и Fa0/2 порт имеет настройки по умолчанию. Данный порт завершил процесс автосогласования, поэтому вместо надписи "auto" под заголовками speed и duplex выводится информация a-full и a-100 (согласованные параметры speed и duplex). Символ "A" перед параметрами full и 100, означает, что указанные значения скорости и дуплекса были согласованы автоматически.

Одновременная настройка интерфейсов с помощью команды interface range

Далее в примере 2 показан способ, облегчающий настройку одних и тех же параметров на нескольких интерфейсах. Для этого используйте команду `interface range`. В примере 2 команда `interface range FastEthernet 0/11-20` сообщает IOS, что следующая подкоманда(ы) применяется к интерфейсам в диапазоне от Fa0/11 до Fa0/20.

```
SW-1(config)# interface range FastEthernet 0/11 - 20
SW-1 (config-if-range)# description end-users connect here
SW-1 (config-if-range)# ^Z
SW-1#
```

IOS действует так, как если бы вы ввели подкоманду под каждым отдельным интерфейсом в указанном диапазоне. Ниже показан фрагмент из вывода команды show running-config, который показывает настройки портов F0 / 11-12 . Из примера видно, что применяются одни и те же настройки на всем диапазоне портов. Для облегчения понимания часть листинга, удалено.

```
SW-1# show running-config

! Lines omitted for brevity

interface FastEthernet0/11
 description buh-end connect here
!
interface FastEthernet0/12
 description buh-end connect here
! Lines omitted for brevity
```

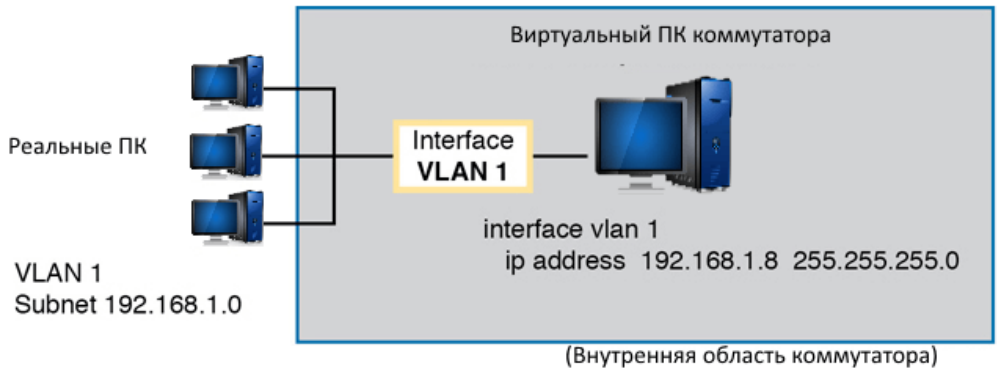
4. Настройка IPv4-адресации для удаленного доступа к устройствам Cisco

Чтобы иметь возможность подключения к коммутатору по Telnet или SSH, а также использовать другие протоколы управления на основе IP (например, Simple Network Management Protocol или SNMP) функционировать должным образом, коммутатору требуется IP-адрес, а также настройки других сопутствующих параметров. IP-адрес не влияет на функциональную работу коммутатора.

В этой части будут рассмотрены основные параметры IPv4-адресации, необходимые для настройки коммутатора, а затем будут приведены команды и примеры настроек. Коммутаторы могут быть настроены с параметрами IPv6-адресации. Настройки IPv4 и IPv6 аналогичны. Далее уделим основное внимание исключительно IPv4.

Настройки IP-адресации узла и коммутатора

Коммутатор нуждается в тех же настройках IP-адресации, что и компьютер с сетевым интерфейсом Ethernet (FastEthernet). Напомню, что каждый ПК имеет процессор. Этот процессор управляется специальной операционной системой для обработки сигналов и отправки их на сетевую карту. Компьютер имеет минимум оду сетевую карту Ethernet (NIC). Настройки сетевой карты ПК включают в себя: настройка статического или получаемого по DHCP IP-адреса сетевой карты. Коммутатор использует те же принципы, что и ПК, но за исключением того, что коммутатор использует виртуальную сетевую карту внутри устройства. Как и ПК, коммутатор имеет реальный процессор, работающий под управлением ОС (IOS). Коммутатор обладает множеством портов Ethernet (FastEthernet, GigEthernet), но в отличие от ПК, коммутатор не назначает IP-адрес управления какому-то конкретному порту или всем сразу. Коммутатор использует NIC концепцию (NIC-like), называемую коммутируемым виртуальным интерфейсом (SVI), или, чаще всего, именуемым интерфейсом VLAN, который действует как отдельная сетевая карта (NIC) коммутатора. Тогда настройки на коммутаторе сводятся к настройке IP-адресации VLAN. Пример настройки показан на рисунке:



На рисунке изображен виртуальный ПК, подключенный к другим реальным узлам в сети через виртуальный интерфейс VLAN 1. IP-адрес интерфейса VLAN1-192.168.1.8; маска подсети 225.255.255.0 и подсеть VLAN 1 - 192.168.1.0. Виртуальный ПК и интерфейс VLAN1 являются частью коммутатора. Остальные узлы находятся за пределами коммутатора. Используя интерфейс VLAN 1 с

настроенной IP-адресацией, коммутатор может отправлять и получать кадры на любом из портов VLAN 1. В коммутаторе Cisco, по умолчанию, все порты назначены во VLAN 1. В коммутаторах можно настроить большое количество VLAN, поэтому у системного администратора есть выбор, какой VLAN использовать. Таки образом IP-адрес управления **не** обязательно должен быть настроен именно на VLAN1

Коммутатору Cisco второго уровня (L2) задается только один IP-адрес для управления. Однако можно использовать любой VLAN, через который подключается коммутатор. Настройка включает: присвоения интерфейса VLAN с указанием его номера (например VLAN11) и присвоением соответствующего IP-адреса с маской подсети.

Например, на рисунке показан коммутатор 2 уровня с несколькими физическими портами в двух различных VLAN (VLAN 1 и 2). На рисунке также показаны подсети, используемые в этих VLAN. Системный администратор может выбрать для использования передачи данных либо то, либо другое.

На рисунке виртуальный ПК коммутатора соединен с другими системами вне устройства с помощью двух интерфейсов VLAN. Подсети виртуальных локальных сетей 192.168.1.0 и 192.168.2.0.

- Интерфейсу VLAN 1 присвоен Ili-адрес из подсети 192.168.1.0
- Интерфейсу VLAN 2, присвоен Ili-адрес из подсети 192.168.2.0

Обратите внимание, что VLAN должен быть привязан к физическому порту коммутатора.

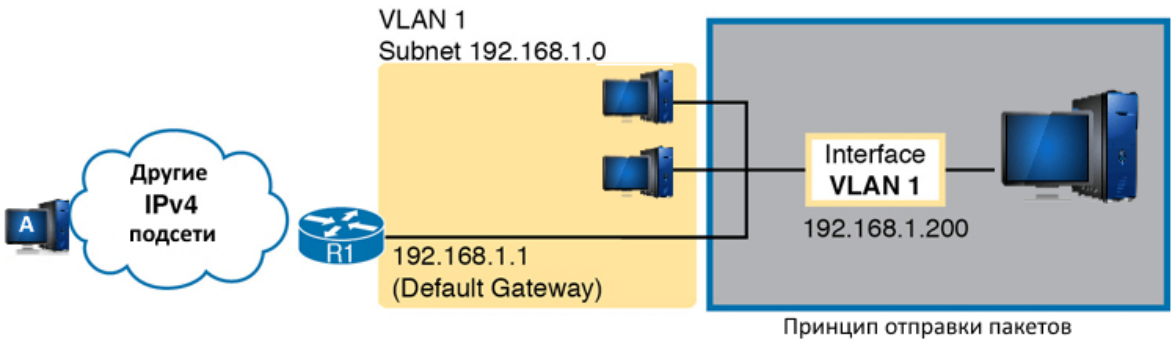
Если этого не сделать, то интерфейс VLAN не включится (то есть он будет в состоянии down), и соответственно коммутатор не сможет обмениваться пакетами с другими устройствами в сети.

> **Примечание**: Некоторые коммутаторы Cisco могут быть настроены для работы в качестве коммутатора 2 уровня или коммутатора 3 уровня. Действуя в качестве коммутатора 2 уровня, коммутатор обрабатывает, пересылает и управляет пакетами Ethernet. В другом случае, коммутатор может работать как коммутатор 3 уровня. Это означает, что коммутатор может выполнять как коммутацию 2 уровня, так и маршрутизацию IP-пакетов уровня 3, используя логику третьего уровня, обычно используемую маршрутизаторами. В данной статье рассматриваются коммутаторы второго уровня (L2)

Настройка IP-адреса (и маски) на одном интерфейсе VLAN позволяет коммутатору обмениваться пакетами с другими узлами в подсети, принадлежащей этой VLAN. Однако коммутатор не может взаимодействовать за пределами локальной подсети без другого параметра конфигурации, называемого шлюзом по умолчанию (default gateway).

Причина настройки шлюза по умолчанию на коммутаторе такая же, как и на обычном компьютере. То есть при отправке пакета сетевая карта компьютера думает, как и кому отправить пакет А именно: отправить IP-пакеты узлам, находящимся в той же подсети, напрямую или отправить IP-пакеты узлам, находящимся в другой подсети, через ближайший маршрутизатор, то есть через шлюз по умолчанию.

На рисунке изображена данная концепция:



Принцип отправки пакетов

На коммутаторе (справа) на VLAN1 настроен IP-адрес 192.168.1.200. Через этот интерфейс (VLAN1) коммутатор может обмениваться пакетами с ПК, входящими в подсеть 192.168.1.0 (желтый сектор) Однако для связи с узлом А, расположенным в левой части рисунка, коммутатор должен использовать маршрутизатор R1 (шлюз по умолчанию) для пересылки IP-пакетов на узел А.

Чтобы пакеты дошли до узла А на коммутаторе необходимо произвести настройку шлюза по умолчанию, указав IP-адрес маршрутизатора R1 (в данном случае 192.168.1.1).

Обратите внимание, что коммутатор и маршрутизатор используют одну и ту же маску, 255.255.255.0, которая помещает адреса в одну подсеть.

Настройка IPv4-адресации на коммутаторе

Настройка IP-адресации на коммутаторе осуществляется настройкой на VLAN.

Следующие этапы показывают команды, используемые для настройки IPv4 на коммутаторе (настройка IP-адресации на VLAN 1).

1. Введите команду `interface vlan 1` в режиме глобальной конфигурации для входа в режим настройки интерфейса VLAN 1.
2. Введите команду `ip address <ip-address> <mask>` для назначения ip-адреса и маски подсети в режиме конфигурации интерфейса.
3. Введите команду `no shutdown` в режиме конфигурации интерфейса, чтобы включить интерфейс VLAN 1, если он еще не включен.
4. Введите команду `ip default-gateway<ip-address>` для назначения ip-адреса шлюза по умолчанию в режиме глобальной конфигурации, чтобы настроить шлюз по умолчанию.
5. (Необязательно) Введите команду `ip name-server ip-address1 ip-address2` ... в режиме глобальной конфигурации, чтобы настроить коммутатор на использование DNS для преобразования имен в соответствующие IP-адреса.

Пример настройки статической IP-адресации

```
SW-1# configure terminal
SW-1 (config)# interface vlan 1
SW-1 (config-if)# ip address 192.168.1.200 255.255.255.0
SW-1 (config-if)# no shutdown
00:25:07: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:25:08: %LINEPROTO-5-UPDOWN: Line protocol on Interface
Vlan1, changed
 state to up
SW-1 (config-if)# exit
SW-1 (config)# ip default-gateway 192.168.1.1
```

В этом примере показана особенно важная и распространенная команда: команда *[no] shutdown*. Что бы включить интерфейс ("поднять") на коммутаторе, используйте команду *no shutdown* в режиме конфигурации интерфейса . Что бы отключить интерфейс используйте в этом же режиме команду *shutdown* . Эта команда может использоваться на физических интерфейсах Ethernet, которые коммутатор использует для пересылки пакетов Ethernet, а также на интерфейсах VLAN. Кроме того, обратите внимание на сообщения, которые появляются непосредственно под командой *no shutdown* в примере выше. Эти сообщения являются сообщениями системного журнала, генерируемыми коммутатором, говорящий о том, что коммутатор действительно включил интерфейс. Коммутаторы (и маршрутизаторы) генерируют сообщения системного журнала в ответ на различные события, и эти сообщения появляются на консоли.

Настройка коммутатора для получения IP-адреса по DHCP

Коммутатор также может использовать протокол Dynamic Host Configuration Protocol (DHCP)для динамического назначения параметров IPv4-адресации. В принципе, все, что вам нужно сделать, это сказать коммутатору использовать DHCP на интерфейсе и включить интерфейс. Предполагая, что DHCP работает в этой сети, коммутатор автоматически получит все его настройки. Следующие этапы показывают команды для настройки коммутатора, используя в качестве примера интерфейс VLAN 1.

1. Войдите в режим конфигурации VLAN 1 с помощью команды глобальной конфигурации `interface vlan` 1 и включите интерфейс с помощью команды `no shutdown` по мере необходимости.
2. Назначьте IP-адрес и маску с помощью подкоманды `ip address dhcp`.

Пример настройки IP-адресации коммутатора по DHCP

```
SW-1# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
SW-1(config)# interface vlan 1
SW-1(config-if)# ip address dhcp
SW-1(config-if)# no shutdown
SW-1(config-if)# ^Z
SW-1#
00:38:20: %LINK-3-UPDOWN: Interface Vlan1, changed state to up
00:38:21: %LINEPROTO-5-UPDOWN: Line protocol on Interface Vlan1, changed state to up
```

Проверка настроек IPv4 - адресации на коммутаторе

Настройку IPv4 адресацию коммутатора можно проверить несколькими способами.

Во-первых, вы всегда можете посмотреть текущую конфигурацию с помощью команды `show running-config`. Во-вторых, вы можете посмотреть информацию об IP-адресе и маске с помощью команды `show interfaces vlan x`, которая показывает подробную информацию о состоянии интерфейса VLAN в VLAN x. Наконец, если используется DHCP, используйте команду `show dhcp lease`, чтобы увидеть (временно) арендованный IP-адрес и другие параметры.

(Обратите внимание, что коммутатор не хранит полученные настройки IP-адресации по DHCP в файле running-config.) Ниже показан пример выходных данных вышеприведенных команд.

```
SW-1# show dhcp lease
Temp IP addr: 192.168.1.101   for peer on Interface: Vlan1
Temp sub net mask: 255.255.255.0
  DHCP Lease server: 192.168.1.1, state: 3 Bound
  DHCP transaction id: 1966
  Lease: 86400 secs,  Renewal: 43200 secs,  Rebind: 75600 secs
Temp default-gateway addr: 192.168.1.1
  Next timer fires after: 11:59:45
  Retry count: 0   Client-ID: cisco-0019.e86a.6fc0-Vl1
  Hostname: SW-1
SW-1# show interfaces vlan 1
Vlan1 is up, line protocol is up
 Hardware is EtherSVI, address is 0019.e86a.6fc0 (bia 0019.e86a.6fc0)
 Internet address is 192.168.1.101/24
 MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec,
    reliability 255/255, txload 1/255, rxload 1/255
! lines omitted for brevity
SW-1# show ip default-gateway
192.168.1.1
```

Выходные данные команды `show interfaces vlan 1` отображают две очень важные детали, связанные с IP-адресацией коммутатора. Во-первых, команда `show` выводит список состояния интерфейса VLAN 1-в данном случае "up/up." Если интерфейс VLAN 1 выключен, тогда коммутатор не сможет отправлять пакеты через этот интерфейс. Примечательно, что если вы забудете выполнить команду `no shutdown`, интерфейс VLAN 1 останется в состоянии выключен и будет указан как " administratively down " в выводе команды show.

Во-вторых, обратите внимание, что выходные данные содержат IP-адрес интерфейса в третьей строке. Если вы вручную настроите IP-адрес, то он всегда будет отображаться; однако, если вы используете DHCP и DHCP не работает, то команда `show interfaces vlan x` не будет выводить IP-адрес на экран. Если же DHCP работает, то вы увидите IP-адрес после использования команды `show interfaces vlan 1`.

5. Протокол ICMP - что это и для чего нужен?

ICMP, который расшифровывается как **Internet Control Message Protocol** это протокол третьего уровня модели OSI, который используется для диагностики проблем со связностью в сети. Говоря простым языком, ICMP помогает определить может ли достичь пакет адреса назначения в установленные временные рамки. Обычно, ICMP “юзают" маршрутизаторы и устройства третьего уровня.

Для чего используется ICMP?

Основная цель ICMP это отчетность об ошибках. При соединении двух девайсов в сети, если часть данных не доходит до адреса назначения, теряется или превышает допустимые таймауты - ICMP генерирует ошибки.

Второе, и, пожалуй, одно из самых популярных применений ICMP это утилиты **ping** и **traceroute**. Термин “пинговать" как - раз связан с протоколом ICMP и “пинговать" хост - означает отправлять ICMP пакеты с целью понять, отвечает ли на них целевое устройство.

Про трассировку

Так и с “трассировкой". Когда говорят “сделайте трассировку маршрута" это означает, что мы хотим увидеть полный маршрут между хостом, на котором выполняется трассировка до хоста назначения. Трассировка покажет каждый из маршрутизаторов на пути до цели и время обработки и прохождения каждого из участков маршрута. Кстати, такой маршрут называется “хопом". Часто говорят: если от узла отправления до узла назначения на пути встретиться 7 маршрутизаторов, то говорят на пути будет 7 хопов. А если на 6 маршрутизаторе пакет обрабатывается дольше обычного, то в среде инженеров говорят “на 6 хопе повышенная задержка". Это один из базовых инструментов того, как можно понять, какой из сетевых узлов на маршруте пакет “сбоит". Именно в этом нам помогает протокол ICMP.

Про пинг

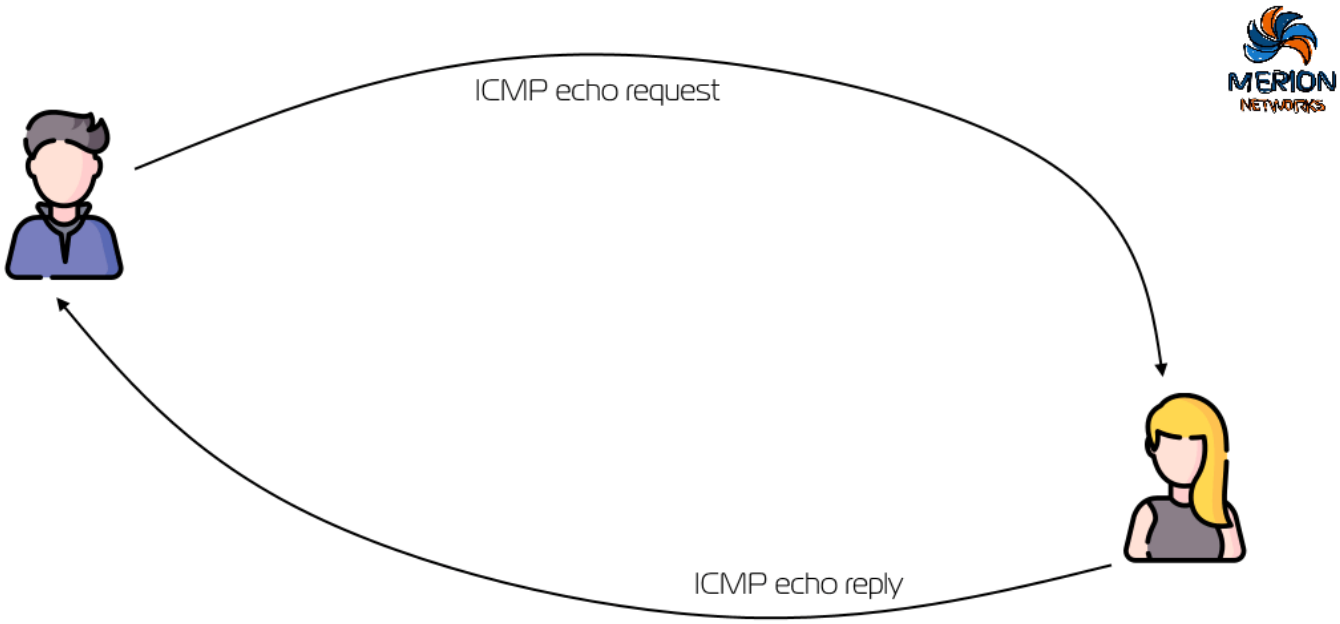
Теперь про **ping**. Можно сказать, это самый базовый инструмент инженера, который позволяет понять “"А жив ли хост?"

Помимо прочего, пинг поможет понять как долго пакет доходит до адреса назначения и, соответственно, поможет измерить задержку.

Работает ping предельно просто:

1. Источник отправляет запрос вида **ICMP echo request**. Это выглядит как вопрос “бро, ты живой?"
2. Получатель отправляет ответ источнику **ICMP echo reply**. Это звучит как ответ вида “да, бро, я жив, спасибо!"

3. Время с момента отправки вопроса до получения ответа суммируется и считается за время пинга



Темная сторона ICMP

На самом деле, с помощью ICMP можно провести атаки на сеть. Эти атаки связаны с отказом устройства в обслуживании (denial-of-service, DoS). Например “флуд - атака", суть которой заключается в отправке огромного количества пинг (ICMP) - запросов на хоста назначения с разных источников. В итоге устройство отвечает кучей пакетов на разные адреса и перегружает собственные мощности и сетевой адаптер.

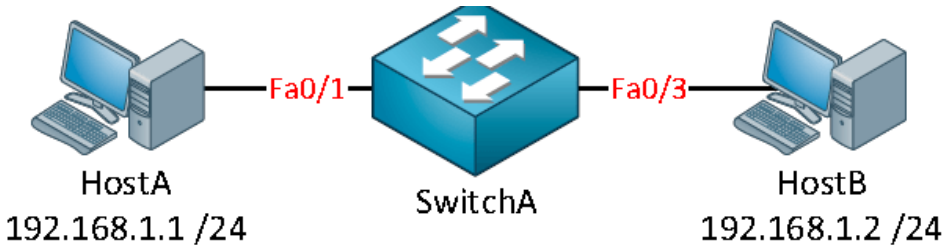
Так же, раньше была популярна атака **Ping of Death**. Если кратко, ее суть заключалась в следующем: злоумышленник намеренно отправляет пакет больше максимального размера. Такой пакет фрагментируется на сети на несколько частей, прилетает в буфер устройства и попадает в очередь на сборка пакета “воедино". Переполнение этой очереди приводило к подвисанию хоста и полному отказу в работе.

Что же, теперь вы знаете, что такое ICMP, почему и как он используется в утилитах ping и трассировке, а так же, какие виды атак можно выполнить с помощью ICMP.

6. Устранение неполадок коммутации Cisco

Для устранения неполадок мы должны пройти путь от нижней части модели OSI к верхней. Для этого нам придется начать с протоколов, которые используются для коммутации. Будем думать о VLAN, транкинге, об агрегировании каналов и связующем дерева. Мы рассмотрим различные протоколы и различные сценарии, где "что-то работает" не так. Мы решим эти проблемы с помощью комбинации команд **show** и **debug**. Первая остановка ... проблемы с интерфейсом!

CASE #1



В этом примере мы имеем коммутатор в центре и два компьютера, которые подключены к нему. Каждый компьютер имеет свой IP-адрес, и они должны иметь возможность пинговать друг друга. Мы будем считать, что компьютеры настроены правильно и там нет никаких проблем.

```
C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

SwitchA#show interfaces fa0/1
FastEthernet0/1 is down, line protocol is down (notconnect)
Hardware is Fast Ethernet, address is 0011.bb0b.3603 (bia 0011.bb0b.3603)
MTU 1900 bytes, BW 100000 Kbit, DLY 100 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Half-duplex, Auto-speed, media type is 10/100BaseTX
input flow-control is off, output flow-control is unsupported
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:26:47, output 00:19:17, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
3457 packets input, 309301 bytes, 0 no buffer
Received 2407 broadcasts (1702 multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 1702 multicast, 0 pause input
0 input packets with dribble condition detected
42700 packets output, 8267872 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 PAUSE output
0 output buffer failures, 0 output buffers swapped out
```

Интерфейс **FastEthernet 0/1** находится в состоянии **down**. Это может указывать на проблему уровня 1, такую как неисправный кабель, неправильный кабель (кроссовер вместо прямого) или, возможно, нерабочая сетевая карта. Обратите внимание, что этот интерфейс работает в полудуплексном режиме. Если повезет, вы можете получить дуплексное сообщение через CDP, которое сообщит вам, что существует дуплексное несоответствие. Если вам не повезло, возможно, из-за этого ваш интерфейс переходит в состояние **down**. Имейте в виду, что гигабитный интерфейс не поддерживает halfduplex.

```
SwitchA(config)#interface fa0/1
SwitchA(config-if)#duplex auto
```

Изменим настройки интерфейса на duplex auto, чтобы коммутатор мог само настроиться.

```
SwitchA#
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up

C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Может быть, нам повезет...но не в этот раз, пинг не работает.

```
SwitchA#show interfaces fa0/3
FastEthernet0/3 is down, line protocol is down (notconnect)
Hardware is Fast Ethernet, address is 0011.bb0b.3605 (bia 0011.bb0b.3605)
MTU 1900 bytes, BW 10000 Kbit, DLY 1000 usec,
reliability 255/255, txload 1/255, rxload 1/255
Encapsulation ARPA, loopback not set
Keepalive set (10 sec)
Auto-duplex, 10Mb/s, media type is 10/100BaseTX
input flow-control is off, output flow-control is unsupported
ARP type: ARPA, ARP Timeout 04:00:00
Last input 00:38:09, output 00:01:42, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0/0 (size/max/drops/flushes); Total output drops: 0
Queueing strategy: fifo
Output queue: 0/40 (size/max)
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
1908 packets input, 181819 bytes, 0 no buffer
Received 858 broadcasts (826 multicasts)
0 runts, 0 giants, 0 throttles
0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored
0 watchdog, 826 multicast, 0 pause input
0 input packets with dribble condition detected
46861 packets output, 9365341 bytes, 0 underruns
0 output errors, 0 collisions, 1 interface resets
0 babbles, 0 late collision, 0 deferred
0 lost carrier, 0 no carrier, 0 PAUSE output
0 output buffer failures, 0 output buffers swapped out
```

Интерфейс **fa0 / 3**, подключенный к хосту B, также не работает. После проверки кабелей и разъемов мы можем проверить ошибки дуплекса и скорости. Дуплекс включен в режим auto, так что это не является проблемой. Скорость была установлена на 10 Мбит, в то время как этот интерфейс является каналом Fast Ethernet (100 Мбит).

```
SwitchA(config)#interface fa0/3
SwitchA(config-if)#speed auto
```

Давайте переключим скорость на авто и посмотрим, что произойдет.

```
SwitchA#
%LINK-3-UPDOWN: Interface FastEthernet0/3, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/3, changed
state to up
```

Похоже, что несоответствие скорости привело к тому, что интерфейс перешел в состояние down. Изменение его на **auto-speed** возвращает интерфейс в состояние up.

```
SwitchA#show ip interface brief
Interface IP-Address OK? Method Status
Protocol
FastEthernet0/1 unassigned YES unset up up
FastEthernet0/3 unassigned YES unset up up
```

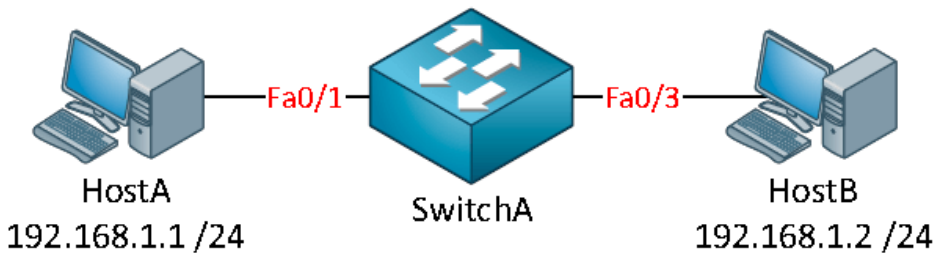
Это то, что мы искали. Интерфейсы, с которыми мы работаем, оба показывают состояние up/up. По крайней мере, теперь мы знаем, что нет никаких ошибок в кабеле, скорости или дуплексе.

```
C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Теперь наш пинг проходит.

Первый урок усвоен: Проверьте свои интерфейсы и посмотрите, отображаются ли они как up/up.

CASE #2



Та же топология, но здесь другая проблема.

```
C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:
Request timed out.
Request timed out.
Ping statistics for 192.168.1.2:
Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Хост A не может пропинговать хост B. Мы начнем с проверки интерфейсов:

```
SwitchA#show ip interface brief
Interface         IP-Address        OK?         Method      Status
Protocol
FastEthernet0/1   unassigned        YES          unset      down     down
FastEthernet0/3   unassigned        YES          unset      up       up
```

Состояние интерфейса FastEthernet0/3 выглядит нормально, но что-то не так с интерфейсом FastEthernet 0/1.

Давайте изучим его подробнее:

```
SwitchA#show interfaces fa0/1
FastEthernet0/1 is down, line protocol is down (err-disabled)
```

Так так, мы видим сообщение **err-disabled**. Это уже дает нам понять, что проблема, где здесь (по крайней мере, это означает, что мы на что-то наткнулись).

```
SwitchA#show interfaces status err-disabled
Port  Name        Status            Reason      Err-disabled      Vlans
Fa0/1             err-disabled      psecure-violation
```

Используйте команду **show interfaces status err-disabled**, чтобы узнать, почему интерфейс перешел в режим **error-disabled**. Это сообщит нам, что причина-безопасность порта.

```
SwitchA#show port-security interface fa0/1
Port Security                :Disabled
Port Status                  :Secure-shutdown
Violation Mode               :Shutdown
Aging Time                   :0 mins
Aging Type                   :Absolute
SecureStatic Address Aging   :Disabled
Maximum MAC Addresses        :1
Total MAC Addresses          :1
Configured MAC Addresses     :1
Sticky MAC Addresses         :0
Last Source Address:Vlan     :000c.2928.5c6c:1
Security Violation Count     :1
```

Мы можем посмотреть на конфигурацию безопасности порта, и мы видим, что только 1 MAC-адрес разрешен. Последний MAC-адрес, который виден на интерфейсе - 000c.2928.5c6c.

```
SwitchA#show port-security interface fa0/1 address
                  Secure Mac Address Table
------------------------------------------------------------------------
Vlan        Mac Address       Type              Ports  Remaining Age(mins)
----        -----------       ----              -----  -------------
    1       0019.569d.5742    SecureConfigured  Fa0/1  -
------------------------------------------------------------------------
Total Addresses: 1
```

Выше мы видим, что интерфейс был настроен для обеспечения безопасности на другой MAC-адрес. Именно по этой причине порт перешел в режим err-disabled.

```
SwitchA(config)#interface fa0/1
SwitchA(config-if)#no switchport port-security
```

Давайте уберем `port security`, чтобы решить эту проблему.

```
SwitchA(config)#interface fa0/1
SwitchA(config-if)#shutdown
SwitchA(config-if)#no shutdown
```

Главное, что вы не должны забыть сделать - это после очистки настройки от port security ваш интерфейс все еще находится в режиме err-disabled. Вам нужно выполнить команды отключения и включения порта (shutdown и no shutdown), чтобы он снова заработал!

```
SwitchA#
%LINK-3-UPDOWN: Interface FastEthernet0/1, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/1, changed
state to up
```

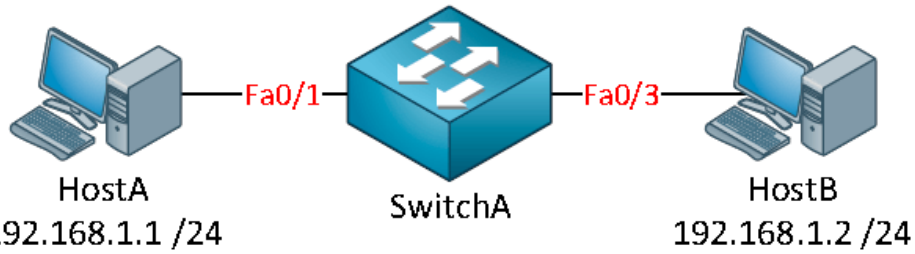
Консоль сообщает нам, что интерфейс теперь включен.

```
C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Как мы видим эхо-запрос проходит между компьютерами. Проблема решена!

Урок 2 усвоен: проверьте, находится ли интерфейс в состоянии **err-disabled**, и если да, то:
а) проверьте, почему это произошло
б) устраните проблему.

CASE #3



Давайте продолжим с другой проблемой. Та же топология, но опять проблема.

```
C:\Documents and Settings\HostA>ping 192.168.1.2
Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Эти два компьютера не "видят" друг друга (нет ответа на ping - Request timed out).
Проверим статус интерфейсов:

```
SwitchA#show ip int brief
Interface         IP-Address        OK?         Method       Status
Protocol
FastEthernet0/1   unassigned        YES         unset        up         up
FastEthernet0/3   unassigned        YES         unset        up         up
```

Интерфейсы выглядят хорошо, никаких ошибок здесь нет.
Проверим - не мешает ли механизм **Port Security**?

```
SwitchA#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)      (Count)         (Count)
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)     : 0
Max Addresses limit in System (excluding one mac per port) : 6144
```

Мы видим, что port security отключена на этом коммутаторе. На данный момент мы, по крайней мере, знаем, что нет никаких проблем с интерфейсом и port security не фильтрует никакие MAC-адреса.

```
SwitchA#show vlan
VLAN Name                             Status    Ports
---- -------------------------------- --------- -------------------------------
----
1 default                            active    Fa0/1, Fa0/2, Fa0/4, Fa0/5
                                               Fa0/6, Fa0/7, Fa0/8, Fa0/9
                                               Fa0/10, Fa0/11, Fa0/12,Fa0/13
                                               Fa0/14, Fa0/15, Fa0/16,Fa0/17
                                               Fa0/18, Fa0/19, Fa0/20,Fa0/21
                                               Fa0/22, Fa0/23, Fa0/24, Gi0/1
                                               Gi0/2
2 VLAN0002                           active    Fa0/3
```

В данный момент это хорошая идея, чтобы проверить информацию о VLAN.
Вы можете использовать команду **show vlan**, чтобы быстро проверить, к какой VLAN принадлежат интерфейсы.

Как вы можете видеть, наши интерфейсы находятся не в одной и той же VLAN.

```
SwitchA(config)#interface fa0/3
SwitchA(config-if)#switchport access vlan 1
```

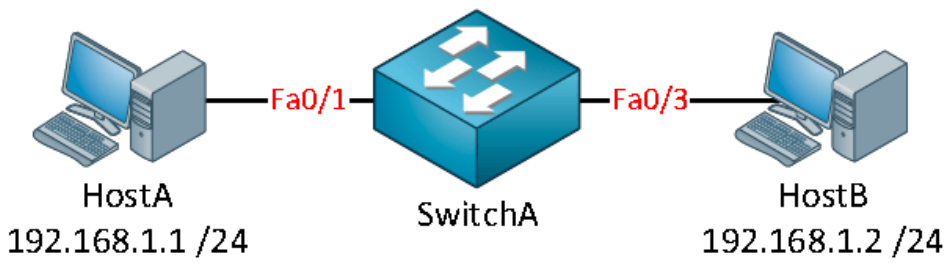
Мы переместим интерфейс fa0/3 обратно в VLAN 1.

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Теперь оба компьютера находятся в одной VLAN. Проблема решена!

Урок 3 усвоен: убедитесь, что интерфейсы находится в нужной VLAN.

CASE #4



Пришло время для другой проблемы!
Наши два компьютера снова не "пингуюся" между собой. Вы теперь знаете, как выглядит неудачный пинг, поэтому скрин не будет публиковаться снова.

```
SwitchA#show ip interface brief
Interface          IP-Address        OK?         Method      Status      Protocol
FastEthernet0/1    unassigned         YES          unset       up            up
FastEthernet0/3    unassigned         YES          unset       up            up
```

Интерфейсы не показывают никаких ошибок.

```
SwitchA#show vlan
VLAN Name                Status      Ports
---- ------------------------------ --------- ----------------------------
----
1  default               active      Fa0/2, Fa0/4, Fa0/5, Fa0/6
                                     Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                     Fa0/11, Fa0/12,
Fa0/13,Fa0/14
                                     Fa0/15, Fa0/16,
Fa0/17,Fa0/18
                                     Fa0/19, Fa0/20,
Fa0/21,Fa0/22
                                     Fa0/23, Fa0/24, Gi0/1,
Gi0/2
10  VLAN0010           active       Fa0/1
```

Мы изучим настройку VLAN. Вы видите, что FastEthernet 0/1 находится в VLAN 10, но мы нигде не видим FastEthernet 0/3. Вот возможные причины:

- Что-то не так с интерфейсом. Мы проверили и убедились, что это не так, потому что он показывает состояние up/up, поэтому он активен.
- Интерфейс находится в другом режиме (не в режиме access port, а в режиме trunk)

```
SwitchA#show interfaces fa0/3 switchport
Name: Fa0/3
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 10 (VLAN0010)
Trunking Native Mode VLAN: 1 (default)
```

Быстрый взгляд на информацию о коммутаторе показывает нам, что нужно знать. Мы убедились, что интерфейс fa0/3 находится в режиме trunk, а native VLAN - 1. Это означает, что всякий раз, когда хост B отправляет трафик и не использует маркировку 802.1 Q, наш трафик заканчивается в VLAN 1.

```
SwitchA(config)#interface fa0/3
SwitchA(config-if)#switchport mode access
SwitchA(config-if)#switchport access vlan 10
```

Мы включим fa0/3 в режим доступа и убедимся, что он находится в VLAN 10.

```
SwitchA#show vlan id 10
VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
----
10   VLAN0010                          active    Fa0/1, Fa0/3
```

Оба интерфейса теперь активны в VLAN 10.

```
SwitchA#show interfaces fa0/3 switchport | include Operational Mode
Operational Mode: static access
```

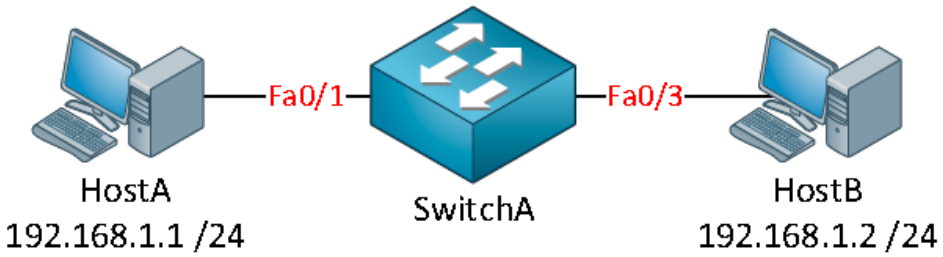
Возможно, лучше проверить информацию на коммутаторе.

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Теперь я могу отправить пинг с хоста а на хост Б...проблема решена!

Урок 4 усвоен: убедитесь, что интерфейс находится в нужном режиме (доступ или магистральный режим).

CASE #5



Те же два компьютера, тот же коммутатор. Однако этот сценарий немного интереснее. Компьютеры не могут пинговать друг друга, поэтому давайте пройдемся по нашему списку "возможных" ошибок:

```
SwitchA#show ip interface brief
Interface         IP-Address        OK?   Method      Status      Protocol
FastEthernet0/1   unassigned        YES   unset       up          up
FastEthernet0/3   unassigned        YES   unset       up          up
```

Интерфейсы выглядят хорошо, up/up-это очень хорошо.

```
SwitchA#show vlan
VLAN Name                             Status Ports
---- -------------------------------- --------- -------------------------------
----
1    default                          active Fa0/2, Fa0/4, Fa0/5, Fa0/6
                                             Fa0/7, Fa0/8, Fa0/9, Fa0/10
                                             Fa0/11, Fa0/12, Fa0/13,Fa0/14
                                             Fa0/15, Fa0/16, Fa0/17,Fa0/18
                                             Fa0/19, Fa0/20, Fa0/21,Fa0/22
                                             Fa0/23, Fa0/24, Gi0/1, Gi0/2
10   VLAN0010                         active Fa0/1, Fa0/3
```

Оба интерфейса находятся в VLAN 10, так что это тоже хорошо.

```
SwitchA#show port-security
Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action
               (Count)       (Count)       (Count)
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Total Addresses in System (excluding one mac per port)      : 0
Max Addresses limit in System (excluding one mac per port)  : 6144
```

Просто чтобы быть уверенным...там нет port security. Это очень интересная ситуация. Интерфейсы работают (в состоянии up/up), мы находимся в одной VLAN, и нет никакой защиты портов. Что еще может быть причиной "перекрытия" трафика?

```
SwitchA#show vlan filter
VLAN Map BLOCKSTUFF is filtering VLANs: 10
```

Ага! Это может быть не то, о чем нам может прийти в голову, но мы же можем использовать VACLs (VLAN access-list), чтобы разрешить или запретить трафик в пределах VLAN. Если вы устраняете неполадки коммутаторов, то необходимо проверить эту настройку, если все остальное кажется вам нормальным. В этом случае есть VACL, подключенный к VLAN 10, давайте проверим его.

```
SwitchA#show vlan access-map
Vlan access-map "BLOCKSTUFF" 10
      Match clauses:
         ip address: 1
      Action:
         drop
Vlan access-map "BLOCKSTUFF" 20
      Match clauses:
      Action:
         Forward
```

Есть два порядковых номера ... 10 и 20. Порядковый номер 10 соответствует access-list 1, и его задача состоит в том, чтобы отбросить трафик. Давайте посмотрим, что это за access-list 1:

```
SwitchA#show access-lists
Standard IP access list 1
      10 permit 192.168.1.0, wildcard bits 0.0.0.255
```

Не смущайтесь из-за заявления о разрешении здесь. Использование оператора permit в access-list означает, что он будет "соответствовать" подсети 192.168.1.0/24. Наши два компьютера используют IP-адреса из этого диапазона. Если он соответствует этому access-list, то VLAN access-map отбросит трафик.

```
SwitchA(config)# vlan access-map BLOCKSTUFF 10
SwitchA(config-access-map)# action forward
```

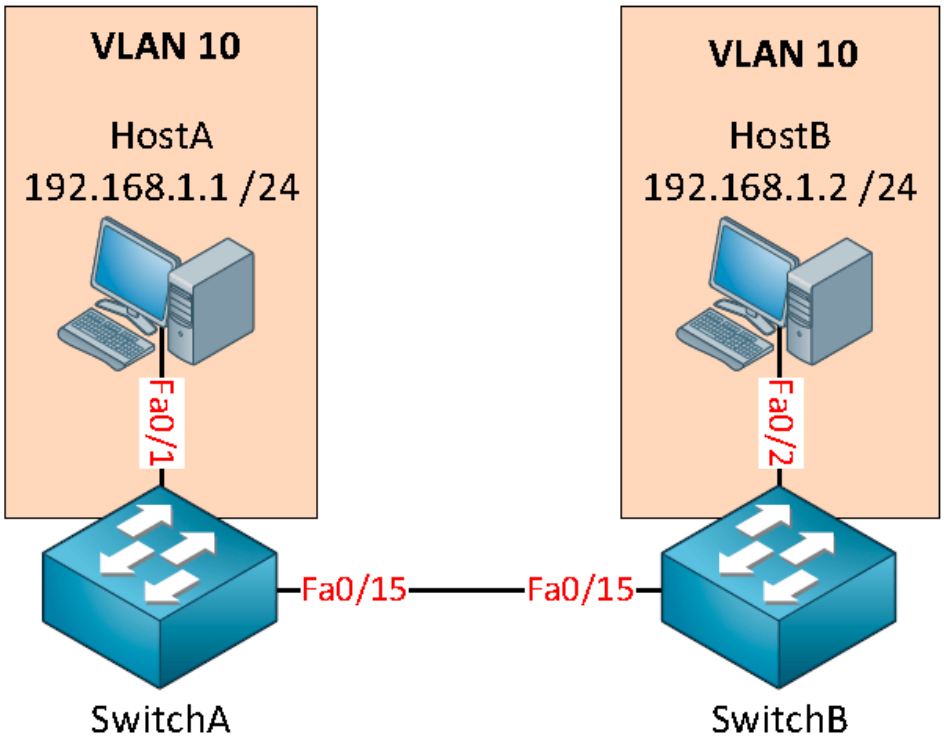
Давайте изменим действие на "forward" и посмотрим, решит ли оно нашу проблему.

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Ну вот, все работает.

Урок 5 усвоен: если все остальное кажется нормальным, убедитесь, что нет никакого VACL!

CASE #6



Давайте продолжим урок 6 с другой топологией. Теперь вы знаете, что нам нужно сначала проверить интерфейсы, а затем VLAN. В этом примере у нас есть те же два компьютера, но теперь у нас есть два коммутатора.
Пинг от Хост А к Хосту Б не работает, так с чего начнем поиск?

```
SwitchA#show interfaces fa0/1
FastEthernet0/1 is up, line protocol is up (connected)

SwitchA#show interfaces fa0/1 switchport
Name: Fa0/1
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 10 (VLAN0010)

SwitchA#show port-security interface fa0/1
      Port Security : Disabled
```

Сначала мы проверим интерфейс fa0/1 на коммутаторе 1. Интерфейс запущен и работает, это switchport, назначенный для VLAN 10. Пока все выглядит неплохо. Port security не включен, так что нам не нужно беспокоиться об этом.

```
SwitchB#show interfaces fa0/2
      FastEthernet0/2 is up, line protocol is up (connected)

SwitchB#show interfaces fa0/2 switchport
Name: Fa0/2
Switchport: Enabled
Administrative Mode: dynamic auto
Operational Mode: static access
Administrative Trunking Encapsulation: negotiate
Operational Trunking Encapsulation: native
Negotiation of Trunking: On
Access Mode VLAN: 10 (VLAN0010)


SwitchB#show port-security interface fa0/2
      Port Security : Disabled
```

Давайте проверим то же самое на коммутаторе 2. Интерфейс работает, и он был назначен на VLAN 10.

В данный момент мы видим, что интерфейсы, "**смотрящие**" к компьютерам выглядят хорошо. В этот момент Вы могли бы сделать две вещи:

- Подключите другой компьютер к коммутатору 1 и назначьте его во VLAN 10. Посмотрите, можно ли общаться между компьютерами во VLAN 10, когда они подключены к одному коммутатору. Сделайте то же самое на коммутаторе 2.
- Проверьте интерфейсы между коммутатором 1 и коммутатором 2.

Мы сконцентрируем свое внимание на интерфейсах между коммутатором 1 и коммутатором 2, потому что там много чего может пойти не так!

```
SwitchA#show interfaces fa0/15
FastEthernet0/15 is up, line protocol is up (connected)

SwitchB#show interfaces fa0/15
FastEthernet0/15 is up, line protocol is up (connected)
```

Интерфейсы не показывают никаких проблем, время проверить информацию о switchport.

```
SwitchA#show interfaces fa0/15 switchport
Name: Fa0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: isl
Operational Trunking Encapsulation: isl
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)
      Trunking Native Mode VLAN: 1 (default)
```

Коммутатор A находится в магистральном режиме и использует инкапсуляцию ISL.

```
SwitchB#show interfaces fa0/15 switchport
Name: Fa0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q
Negotiation of Trunking: On
Access Mode VLAN: 1 (default)Trunking Native Mode VLAN: 1 (default)
```

Коммутатор B также находится в магистральном (**trunk**) режиме, но использует инкапсуляцию 802.1Q (**dot1q**). Имейте в виду, что (в зависимости от модели коммутатора) административный режим по умолчанию может быть **dynamic auto**. Два интерфейса, которые оба работают в dynamic auto режиме, станут портом доступа (**access**). Лучше всего самостоятельно переключить интерфейс в магистральный режим. В нашем случае оба интерфейса магистральные, так что это хорошо, но у нас есть несоответствие протокола инкапсуляции.

```
SwitchA(config)#interface fa0/15
SwitchA(config-if)#switchport trunk encapsulation dot1q
```

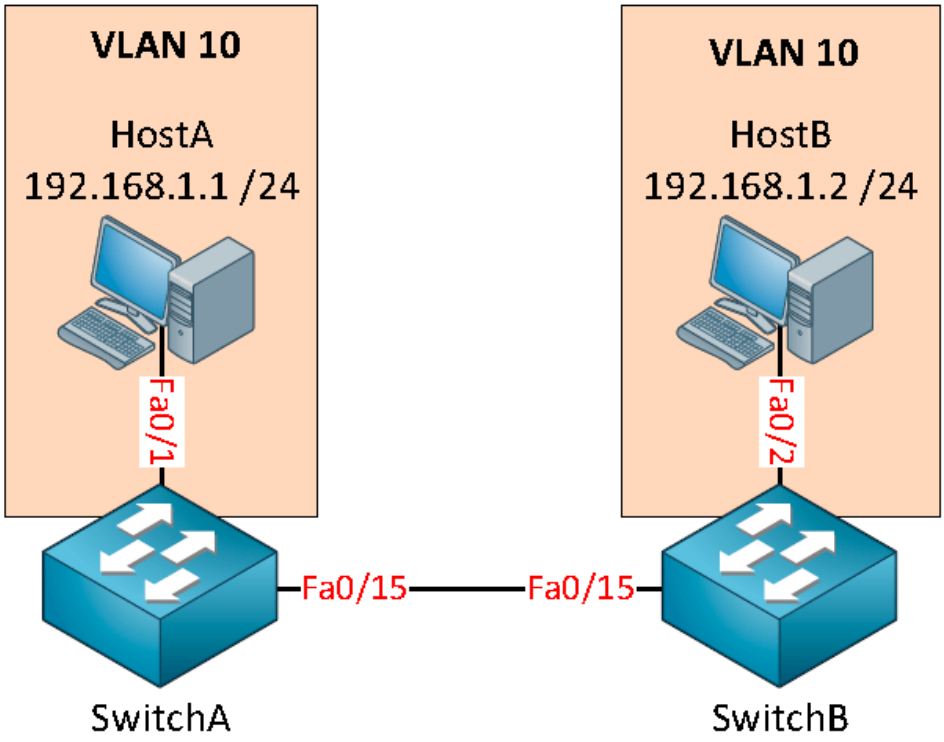
Мы изменим тип инкапсуляции, чтобы оба коммутатора использовали протокол 802.1Q.

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Проблема решена! И опять все работает.

> **Урок 6 усвоен**: убедитесь, что при настройке магистралей используется один и тот же протокол инкапсуляции.

CASE #7



Вот опять тот же сценарий. Сейчас рассмотрим еще кое-что, что важно проверить при решении проблем trunk. Предположим, мы проверили и убедились, что следующие элементы не вызывают никаких проблем:

- Интерфейсы (скорость/дуплекс).
- Безопасность портов.
- Конфигурация Switchport (назначение VLAN, интерфейс, настроенный в режиме доступа).

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

К сожалению, эхо-запрос между компьютерами все еще не проходит. Давайте взглянем на интерфейсы fa0/15 на коммутаторах:

```
SwitchA#show interfaces fa0/15 switchport
Name: Fa0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
Operational Trunking Encapsulation: dot1q


SwitchB#show interfaces fa0/15 switchport
Name: Fa0/15
Switchport: Enabled
Administrative Mode: trunk
Operational Mode: trunk
Administrative Trunking Encapsulation: dot1q
      Operational Trunking Encapsulation: dot1q
```

Проверим, что оба интерфейса находятся в магистральном режиме и что мы используем один и тот же протокол инкапсуляции (802.1 Q). Здесь нет никаких проблем. Что-нибудь еще, что может пойти не так с этой магистральной связью? Да!

```
SwitchA#show interfaces fa0/15 trunk
Port        Mode         Encapsulation           Status            Native vlan
Fa0/15      on           802.1q                  trunking          1
Port        Vlans allowed on trunk
Fa0/15      20


SwitchB#show interfaces fa0/15 trunk
Port        Mode         Encapsulation           Status            Native vlan
Fa0/15      on           802.1q                  trunking          1
Port        Vlans allowed on trunk
Fa0/15      20
```

Магистраль может быть работоспособной, но это не означает, что все VLAN разрешены по магистральному каналу связи. В приведенном выше примере вы видите, что разрешена только VLAN 20.

```
SwitchA(config)#interface fa0/15
SwitchA(config-if)#switchport trunk allowed vlan all

SwitchB(config)#interface fa0/15
SwitchB(config-if)#switchport trunk allowed vlan all
```

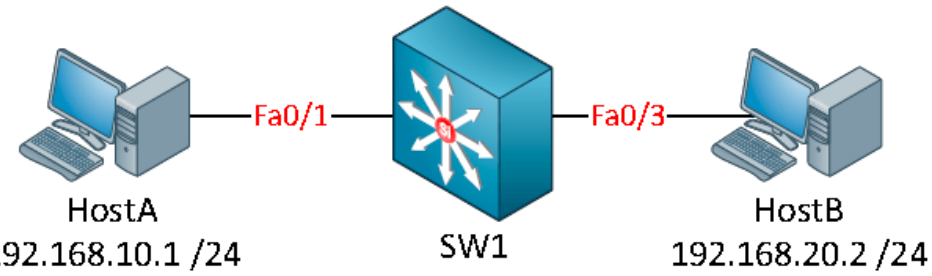
Давайте позволим всем VLAN пройти магистраль.

```
C:\Documents and Settings\HostA>ping 192.168.1.2

Pinging 192.168.1.2 with 32 bytes of data:

Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128
Reply from 192.168.1.2: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

По магистральной линии может передаваться трафик VLAN 10 между двумя коммутаторами. В результате пинг идет между компьютерами....еще одна проблема решена!

Урок 7 усвоен: всегда проверяйте, разрешает ли магистраль все VLAN или нет.

CASE #8



Вот вам новый сценарий. Два компьютера, имеют разные IP-адреса. Коммутатор - это многоуровневый коммутатор. Поскольку компьютеры находятся в разных подсетях, нам приходится беспокоиться о маршрутизации.

```
C:\Documents and Settings\HostA>ping 192.168.20.2

Pinging 192.168.1.2 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.2:
      Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

Мы видим, что два компьютера не могут связаться друг с другом. С чего мы должны начать устранение неполадок?

```
C:\Documents and Settings\HostA>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

            Connection-specific DNS Suffix .  :
            IP Address. . . . . . . . . . . . : 192.168.10.1
            Subnet Mask . . . . . . . . . . . : 255.255.255.0
            Default Gateway . . . . . . . . . : 192.168.10.254
```

Это статья не о настройке windows, но нам нужно обратить внимание на наши хосты. Поскольку компьютеры должны "выйти из своей собственной подсети", мы должны проверить, что IP-адрес шлюза по умолчанию в порядке и доступен.

```
C:\Documents and Settings\VMWare>ping 192.168.10.254

Pinging 192.168.10.254 with 32 bytes of data:

Reply from 192.168.10.254: bytes=32 time=3ms TTL=255
Reply from 192.168.10.254: bytes=32 time=1ms TTL=255
Reply from 192.168.10.254: bytes=32 time=2ms TTL=255
Reply from 192.168.10.254: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.10.254:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 1ms, Maximum = 3ms, Average = 1ms
```

Хост A может достичь шлюза по умолчанию, поэтому мы, по крайней мере, знаем, что хост A работает нормально.

```
C:\Documents and Settings\HostB>ipconfig

Windows IP Configuration

Ethernet adapter Local Area Connection:

            Connection-specific DNS Suffix .  :
            IP Address. . . . . . . . . . . . : 192.168.20.2
            Subnet Mask . . . . . . . . . . . : 255.255.255.0
            Default Gateway . . . . . . . . . : 192.168.20.254
```

Вот IP-конфигурация хоста B. Давайте проверим доступность шлюза по умолчанию!

```
C:\Documents and Settings\HostB>ping 192.168.20.254

Pinging 192.168.20.254 with 32 bytes of data:

Reply from 192.168.20.254: bytes=32 time=4ms TTL=255
Reply from 192.168.20.254: bytes=32 time=2ms TTL=255
Reply from 192.168.20.254: bytes=32 time=2ms TTL=255
Reply from 192.168.20.254: bytes=32 time=1ms TTL=255

Ping statistics for 192.168.20.254:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 1ms, Maximum = 4ms, Average = 2ms
```

Здесь тоже все работает. Мы знаем, что компьютеры рабочие, потому что они знают, как выйти из своей собственной подсети, и шлюз по умолчанию доступен. Пора проверить коммутатор.

```
SwitchA#show interfaces fa0/1 switchport | include VLAN
Access Mode VLAN: 10 (VLAN0010)

SwitchA#show interfaces fa0/3 switchport | include VLAN
Access Mode VLAN: 20 (VLAN0020)
```

Как мы видим, что хост A находится в VLAN 10 и хост B находится в VLAN 20. Мы не проверяли, включены ли интерфейсы, потому что мы можем пинговать IP-адреса шлюза по умолчанию. Это говорит о том, что fa0/1 и fa0/3 работают, но мы не знаем, к какой VLAN они принадлежат.

```
SwitchA#show ip int brief | include Vlan
Vlan1              unassigned        YES   TFTP        up    down
Vlan10             192.168.10.254    YES   manual      up    up
Vlan20             192.168.20.254    YES   manual      up    up
```

Были сконфигурированы два интерфейса SVI. Это IP-адреса, которые компьютеры используют в качестве шлюза по умолчанию. Так почему же наш коммутатор не маршрутизирует трафик?

```
SwitchA#show ip route
Default gateway is not set

Host              Gateway           Last Use          Total Uses Interface
ICMP redirect cache is empty
```

Наличие IP-адресов на интерфейсах не означает автоматическую маршрутизацию трафика. Для этого нам потребуется таблица маршрутизации. Этот коммутатор не имеет

```
SwitchA(config)#ip routing
```

Давайте включим маршрутизацию на этом коммутаторе.

```
SwitchA#show ip route connected
C     192.168.10.0/24 is directly connected, Vlan10
C     192.168.20.0/24 is directly connected, Vlan20
```

Давайте сделаем так, чтобы это выглядело получше. Теперь коммутатор знает, куда перенаправлять IP-пакеты на этом коммутаторе.

```
C:\Documents and Settings\HostA>ping 192.168.20.2

Pinging 192.168.20.2 with 32 bytes of data:

Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127
Reply from 192.168.20.2: bytes=32 time<1ms TTL=127

Ping statistics for 192.168.20.2:
      Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
      Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Вот так...теперь два компьютера могут достучаться друг до друга! Проблема решена!

Урок 8 усвоен: если вы используете многоуровневый коммутатор для маршрутизации interVLAN, убедитесь, что интерфейсы SVI настроены правильно и что маршрутизация включена.

Мы рассмотрели наиболее распространенные ошибки, которые могут произойти с нашими интерфейсами, VLAN, транками и проблемами маршрутизации при использовании многоуровневых коммутаторов.

В следующей статье мы рассмотрим связующее дерево. Spanning-tree-довольно надежный протокол, но есть ряд вещей, которые могут пойти не так, как, вы ожидаете. Кроме того, из-за неправильной настройки могут произойти некоторые странные вещи...

7. Обслуживание и траблшутинг сетей

В первой части этой статьи мы сначала рассмотрим некоторые методы обслуживания сетей. Существуют различные модели, которые помогут вам поддерживать ваши сети и сделать вашу жизнь проще. Во второй части статьи мы рассмотрим некоторые теоретические модели, которые помогут вам в устранении неполадок.

Начнем с рассмотрения технического обслуживания сети! Обслуживание сети в основном означает, что вы должны делать все необходимое для поддержания сети в рабочем состоянии, и это включает в себя ряд задач:

- Устранение неполадок в сети;
- Установка и настройка аппаратного и программного обеспечения;
- Мониторинг и повышение производительности сети;
- Планирование будущего расширения сети;
- Создание сетевой документации и поддержание ее в актуальном состоянии;
- Обеспечение соблюдения политики компании;
- Обеспечение соблюдения правовых норм;
- Обеспечение безопасности сети от всех видов угроз.

Конечно, этот список может отличаться для каждой сети, в которой вы работаете. Все эти задачи можно разделить на два вида:

- Структурированные (проактивные) задачи;
- Interrupt-driven (управляемые прерываниями) задачи.

"Структурированный" означает, что у вас есть заранее определенный план обслуживания сети, который гарантирует, что проблемы будут решены до того, как они возникнут. Как системному администратору, это сделает жизнь намного проще. Управляемый прерыванием означает, что вы просто ждете возникновения проблемы, а затем исправляете ее так быстро, как только можете. Управляемый прерыванием подход больше похож на подход "**пожарного**" ...вы ждете, когда случится беда, а затем пытаетесь решить проблему. Структурированный подход, при котором у вас есть стратегия и план обслуживания сети, сокращает время простоя и является более экономичным.

Конечно, вы никогда не сможете полностью избавиться от Interrupt-driven, потому что иногда все "**просто идет не так**", но с хорошим планом мы можем точно сократить количество задач, управляемых прерываниями.

Вам не нужно думать о модели обслуживания сети самостоятельно. Есть ряд хорошо известных моделей обслуживания сети, которые используются сетевыми администраторами. Лучше всего использовать одну из моделей, которая лучше всего подходит для вашей организации и подкорректировать, если это необходимо.

Вот некоторые из известных моделей обслуживания сети:

- **FCAPS**:

- Управление неисправностями.
- Управление конфигурацией.
- Управление аккаунтингом.
- Управление производительностью.
- Управление безопасностью.

Модель обслуживания сети FCAPS была создана ISO (Международной организацией стандартизации).

- **ITIL**: библиотека ИТ-инфраструктуры - это набор практик для управления ИТ-услугами, который фокусируется на приведении ИТ-услуг в соответствие с потребностями бизнеса.
- **TMN**: сеть управления телекоммуникациями - это еще одна модель технического обслуживания, созданная ITU-T (сектор стандартизации телекоммуникаций) и являющаяся вариацией модели FCAPS. TMN нацелена на управление телекоммуникационными сетями.
- **Cisco Lifecycle Services**: конечно, Cisco имеет свою собственную модель обслуживания сети, которая определяет различные фазы в жизни сети Cisco:
  - **Подготовка**
  - **Планирование**
  - **Проектирование**
  - **Внедрение**
  - **Запуск**
  - **Оптимизация**

Выбор модели обслуживания сети, которую вы будете использовать, зависит от вашей сети и бизнеса. Вы также можете использовать их в качестве шаблона для создания собственной модели обслуживания сети.

Чтобы дать вам представление о том, что такое модель обслуживания сети и как она выглядит, ниже приведен пример для **FCAPS**:

- **Управление неисправностями**: мы будем настраивать наши сетевые устройства (маршрутизаторы, коммутаторы, брандмауэры, серверы и т. д.) для захвата сообщений журнала и отправки их на внешний сервер. Всякий раз, когда интерфейс выходит из строя или нагрузка процессора превышает 80%, мы хотим получить сообщение о том, чтобы узнать, что происходит.
- **Управление конфигурацией**: любые изменения, внесенные в сеть, должны регистрироваться в журнале. Чаще всего используют управление изменениями, чтобы соответствующий персонал был уведомлен о планируемых изменениях в сети. Изменения в сетевых устройствах должны быть зарегистрированы и утверждены до того, как они будут реализованы.
- **Управление аккаунтингом**: Мы будем взимать плату с (гостевых) пользователей за использование беспроводной сети, чтобы они платили за каждые 100 МБ данных или что-то в этом роде. Он также обычно используется для взимания платы с людей за междугородние VoIP-звонки.
- **Управление производительностью**: производительность сети будет контролироваться на всех каналах LAN и WAN, чтобы мы знали, когда что-то пойдет не так. QoS (качество обслуживания)

будет настроено на соответствующих интерфейсах.
- **Управление безопасностью**: мы создадим политику безопасности и реализуем ее с помощью брандмауэров, VPN, систем предотвращения вторжений и используем AAA (Authorization, Authentication and Accounting) для проверки учетных данных пользователей. Сетевые нарушения должны регистрироваться, и должны быть приняты соответствующие мероприятия.

Как вы видите, что FCAPS - это не просто "теоретический" метод, но он действительно описывает "**что**", "**как**" и "**когда**" мы будем делать.

Какую бы модель обслуживания сети вы ни решили использовать, всегда есть ряд рутинных задач обслуживания, которые должны иметь перечисленные процедуры, вот несколько примеров:

- **Изменения конфигурации**: бизнес никогда не стоит на месте, он постоянно меняется. Иногда вам нужно внести изменения в сеть, чтобы разрешить доступ для гостевых пользователей, обычные пользователи могут перемещаться из одного офиса в другой, и для облегчения этой процедуры вам придется вносить изменения в сеть.
- **Замена оборудования**: старое оборудование должно быть заменено более современным оборудованием, и также возможна ситуация, когда производственное оборудование выйдет из строя, и нам придется немедленно заменить его.
- **Резервные копии**: если мы хотим восстановиться после сетевых проблем, таких как отказавшие коммутаторы или маршрутизаторы, то нам нужно убедиться, что у нас есть последние резервные копии конфигураций. Обычно вы используете запланированные резервные копии, поэтому вы будете сохранять текущую конфигурацию каждый день, неделю, месяц или в другое удобное для вас время.
- **Обновления программного обеспечения**: мы должны поддерживать ваши сетевые устройства и операционные системы в актуальном состоянии. Обновления позволяют исправлять ошибки ПО. Также обновление проводится для того, чтобы убедиться, что у нас нет устройств, на которых работает старое программное обеспечение, имеющее уязвимости в системе безопасности.
- **Мониторинг**: нам необходимо собирать и понимать статистику трафика и использования полосы пропускания, чтобы мы могли определить (будущие) проблемы сети, но также и планировать будущее расширение сети.

Обычно вы создаете список задач, которые должны быть выполнены для вашей сети. Этим задачам можно присвоить определенный приоритет. Если определенный коммутатор уровня доступа выходит из строя, то вы, вероятно, захотите заменить его так быстро, как только сможете, но нерабочее устройство распределения или основного уровня будет иметь гораздо более высокий приоритет, поскольку оно влияет на большее число пользователей Сети.

Другие задачи, такие как резервное копирование и обновление программного обеспечения, могут быть запланированы. Вы, вероятно, захотите установить обновления программного обеспечения вне рабочего времени, а резервное копирование можно запланировать на каждый день после полуночи. Преимущество планирования определенных задач заключается в том, что сетевые инженеры с меньше всего забудут их выполнить.

Внесение изменений в вашу сеть иногда влияет на производительность пользователей, которые полагаются на доступность сети. Некоторые изменения будут очень важны, изменения в брандмауэрах или правилах списка доступа могут повлиять на большее количество пользователей,

чем вы бы хотели. Например, вы можете установить новый брандмауэр и запланировать определенный результат защиты сети. Случайно вы забыли об определенном приложении, использующем случайные номера портов, и в конечном итоге устраняете эту проблему. Между тем некоторые пользователи не получат доступ к этому приложению (и возмущаются, пока вы пытаетесь его исправить...).

Более крупные компании могут иметь более одного ИТ-отдела, и каждый отдел отвечает за различные сетевые услуги. Если вы планируете заменить определенный маршрутизатор завтра в 2 часа ночи, то вы можете предупредить парней из отдела "**ИТ-отдел-2**", о том, что их серверы будут недоступны. Для этого можно использовать управление изменениями. Когда вы планируете внести определенные изменения в сеть, то другие отделы будут проинформированы, и они могут возразить, если возникнет конфликт с их планированием.

Перед внедрением управления изменениями необходимо подумать о следующем:

- **Кто** будет отвечать за авторизацию изменений в сети?
- **Какие** задачи будут выполняться во время планового технического обслуживания windows, linux, unix?
- **Какие** процедуры необходимо соблюдать, прежде чем вносить изменения? (например: выполнение "copy run start" перед внесением изменений в коммутатор).
- **Как** вы будете измерять успех или неудачу сетевых изменений? (например: если вы планируете изменить несколько IP-адресов, вы запланируете время, необходимое для этого изменения. Если для перенастройки IP-адресов требуется 5 минут, а вы в конечном итоге устраняете неполадки за 2 часа, так как еще не настроили. Из-за этого вы можете "откатиться" к предыдущей конфигурации. Сколько времени вы отводите на устранение неполадок? 5 минут? 10 минут? 1 час?
- **Как**, **когда** и **кто** добавит сетевое изменение в сетевую документацию?
- **Каким** образом вы создадите план отката, чтобы в случае непредвиденных проблем восстановить конфигурацию к предыдущей конфигурации?
- **Какие** обстоятельства позволят отменить политику управления изменениями?

Еще одна задача, которую мы должны сделать - это создать и обновить вашу сетевую документацию. Всякий раз, когда разрабатывается и создается новая сеть, она должна быть задокументирована. Более сложная часть состоит в том, чтобы поддерживать ее в актуальном состоянии. Существует ряд элементов, которые вы должны найти в любой сетевой документации:

- **Физическая топологическая схема (физическая карта сети)**: здесь должны быть показаны все сетевые устройства и то, как они физически связаны друг с другом.
- **Логическая топологическая схема (логическая карта сети)**: здесь необходимо отобразить логические связи между устройствами, то есть как все связано друг с другом. Показать используемые протоколы, информация о VLAN и т. д.
- **Подключения**: полезно иметь диаграмму, которая показывает, какие интерфейсы одного сетевого устройства подключены к интерфейсу другого сетевого устройства.
- **Инвентаризация**: вы должны провести инвентаризацию всего сетевого оборудования, списков поставщиков, номера продуктов, версии программного обеспечения, информацию о лицензии на программное обеспечение, а также каждое сетевое устройство должно иметь инвентарный номер.

- **IP-адреса**: у вас должна быть схема, которая охватывает все IP-адреса, используемые в сети, и на каких интерфейсах они настроены.
- **Управление конфигурацией**: перед изменением конфигурации мы должны сохранить текущую запущенную конфигурацию, чтобы ее можно было легко восстановить в предыдущую (рабочую) версию. Еще лучше хранить архив старых конфигураций для дальнейшего использования.
- **Проектная документация**: документы, которые были созданы во время первоначального проектирования сети, должны храниться, чтобы вы всегда могли проверить, почему были приняты те или иные проектные решения.

Это хорошая идея, чтобы работать с пошаговыми рекомендациями по устранению неполадок или использовать шаблоны для определенных конфигураций, которые все сетевые администраторы согласны использовать.

Ниже показаны примеры, чтобы вы понимали, о чем идет речь:

```
interface FastEthernet0/1
description AccessPoint
switchport access vlan 2
switchport mode access
spanning-tree portfast
```

Вот пример интерфейсов доступа, подключенных к беспроводным точкам доступа. Portfast должен быть включен для связующего дерева, точки доступа должны быть в VLAN 2, а порт коммутатора должен быть изменен на "доступ" вручную.

```
interface FastEthernet0/2
description VOIP
interface FastEthernet0/2
description ClientComputer
switchport access vlan 6
switchport mode access
switchport port-security
switchport port-security violation shutdown
switchport port-security maximum 1
spanning-tree portfast
spanning-tree bpduguard enable
```

Вот шаблон для интерфейсов, которые подключаются к клиентским компьютерам. Интерфейс должен быть настроен на режиме "доступа" вручную. Безопасность портов должна быть включена, поэтому допускается только 1 MAC-адрес (компьютер). Интерфейс должен немедленно перейти в режим переадресации, поэтому мы настраиваем spanning-tree portfast, и, если мы получаем BPDU, интерфейс должен перейти в err-disabled. Работа с предопределенными шаблонами, подобными этим, уменьшит количество ошибок, потому что все согласны с одной и той же конфигурацией. Если вы дадите каждому сетевому администратору инструкции по ""защите интерфейса", вы, вероятно, получите 10 различных конфигураций

```
interface GigabitEthernet0/1
description TRUNK
switchport trunk encapsulation dot1q
switchport mode trunk
switchport trunk nonegotiate
```

Вот еще один пример для магистральных соединений. Если вы скажете 2 сетевым администраторам "настроить магистраль", вы можете в конечном итоге получить один интерфейс, настроенный для инкапсуляции 802.1Q, а другой-для инкапсуляции ISL. Если один сетевой администратор отключил DTP, а другой настроил интерфейс как "**dynamic desirable**", то он также не будет работать. Если вы дадите задание им настроить магистраль в соответствии с шаблоном, то у нас будет одинаковая конфигурация с обеих сторон.

8. Port-Security

Поговорим о базовой сетевой безопасности, а именно о **Port-Security** и о том, как его настроить на коммутаторах **Cisco**.

Для начала разберемся, что же вообще такое Port-Security. Port-Security – это функция коммутатора, при помощи которой мы можем указать каким устройствам можно пропускать трафик через определенные порты. Устройство определяется по его **MAC**-адресу.

Эта функция предназначена для защиты от несанкционированного подключения к сети и атак, направленных на переполнение таблицы MAC-адресов. При помощи нее мы можем указывать конкретные адреса, с которых разрешен доступ или указывать максимальное количество MAC-адресов, которые могут передавать трафик через порт.

Типы Port-Security

Существует несколько способов настройки port-security:

- **Статические MAC-адреса** – MAC-адреса, которые вручную настроены на порту, из режима конфигурации порта при помощи команды **switchport port-security mac-address [MAC-адрес]** . MAC-адреса, сконфигурированные таким образом, сохраняются в таблице адресов и добавляются в текущую конфигурацию коммутатора.
- **Динамические MAC-адреса** - MAC-адреса, которые динамически изучаются и хранятся только в таблице адресов. MAC-адреса, сконфигурированные таким образом, удаляются при перезапуске коммутатора.
- **Sticky MAC-адреса** - MAC-адреса, которые могут быть изучены динамически или сконфигурированы вручную, затем сохранены в таблице адресов и добавлены в текущую конфигурацию.

Sticky MAC-адреса

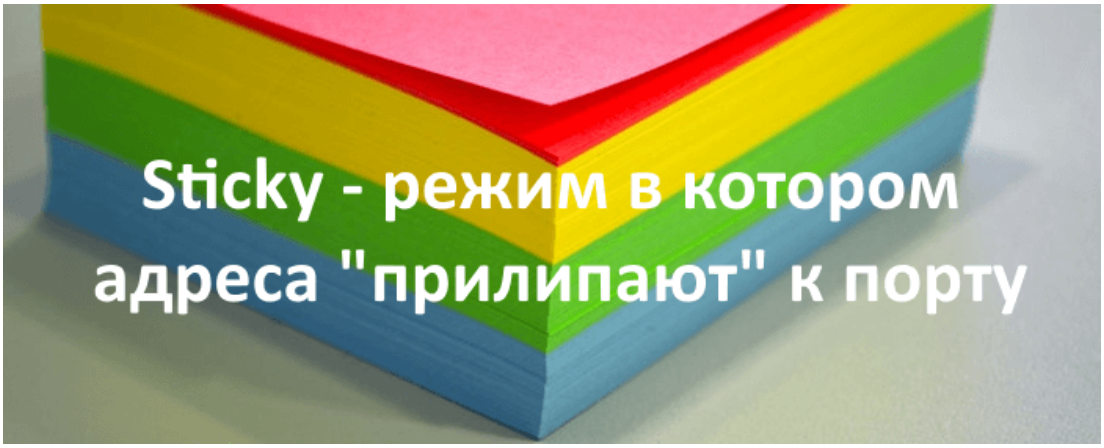
Если необходимо настроить port-security со **sticky** MAC-адресами, которые преобразуются с из динамически изученных адресов и добавляются в текущую конфигурацию, то необходимо настроить так называемое sticky изучение. Для того чтобы его включить необходимо на интерфейсе коммутатора выполнить команду **switchport port-security mac-address sticky** из режима конфигурации интерфейса.

Когда эта команда введена, коммутатор преобразует все динамически изученные MAC-адреса (включая те, которые были динамически изучены до того, как было включено sticky обучение) к sticky MAC-адресам. Все sticky MAC-адреса добавляются в таблицу адресов и в текущую конфигурацию.

Также sticky адреса можно указать вручную. Когда sticky MAC-адреса настроены при помощи команды **switchport port-security mac-address sticky [MAC-адрес]**, все указанные адреса добавляются в таблицу адресов и текущую конфигурацию.

Если sticky MAC-адреса сохранены в файле конфигурации, то при перезапуске коммутатора или отключении интерфейса интерфейс не должен будет переучивать адреса. Если же sticky адреса не будут сохранены, то они будут потеряны.

Если sticky обучение отключено при помощи команды **no switchport port-security mac-address sticky** , то эти адреса будут оставаться в таблице адресов, но удалятся из текущей конфигурации.



Обратите внимание, что port-security не будут работать до тех пор, пока не будет введена команда, включающая его - **switchport port-security**

Нарушение безопасности

Нарушением безопасности являются следующие ситуации:

- Максимальное количество MAC-адресов было добавлено в таблицу адресов для интерфейса, а устройство, MAC-адрес которого отсутствует в таблице адресов, пытается получить доступ к интерфейсу.
- Адрес, полученный или сконфигурированный на одном интерфейсе, отображается на другом интерфейсе в той же VLAN.

На интерфейсе может быть настроен один из трех режимов реагирования при нарушении:

- **Protect** - когда количество MAC-адресов достигает предела, разрешенного для порта, пакеты с неизвестными исходными адресами отбрасываются до тех пор, пока не будет удалено достаточное количество MAC-адресов или количество максимально допустимых адресов для порта не будет увеличено. Уведомление о нарушении безопасности отсутствует в этом случае.
- **Restrict** – то же самое, что и в случае Protect, однако в этом случае появляется уведомление о нарушении безопасности. Счетчик ошибок увеличивается
- **Shutdown** – стандартный режим, в котором нарушения заставляют интерфейс немедленно отключиться и отключить светодиод порта. Он также увеличивает счетчик нарушений. Когда порт находится в этом состоянии (**error-disabled**), его можно вывести из него введя команды **shutdown и no shutdown** в режиме конфигурации интерфейса.

Чтобы изменить режим нарушения на порту коммутатора, используется команда **port-security violation {protect | restrict |shutdown}** в режиме конфигурации интерфейса.

Режим реагирования	**Передача траффика**	**Отправка сообщения syslog**	**Отображение сообщения об ошибке**	**Увеличение счетчика нарушений**	**Выключение порта**
Protect	Нет	Нет	Нет	Нет	Нет
Restrict	Нет	Да	Нет	Да	Нет
Shutdown	Нет	Нет	Нет	Да	Да

Настройка

Рассмотрим пример настройки:

```
Switch#interface fa0/1 – заходим в режим конфигурации порта
Switch(config-ig)#switchport mode access – делаем порт access
Switch(config-ig)#switchport port-security – включаем port-security
Switch(config-ig)#switchport port-security maximum 50 – задаем максимальное количество адресов на порту
Switch(config-ig)#switchport port-security mac-address sticky – включаем sticky изучение
```

Если мы не будем ничего уточнять и просто включим port-security командой **switchport port-security** в режиме конфигурации интерфейса, то максимальное количество адресов на порту будет один, sticky изучение будет выключено, а режим нарушения безопасности будет shutdown.

Проверка порта

Чтобы отобразить параметры port-security используется команда **show port-security [номер_интерфейса]** .

Чтобы отобразить все защищенные MAC-адреса используется команда **show port-security address**.

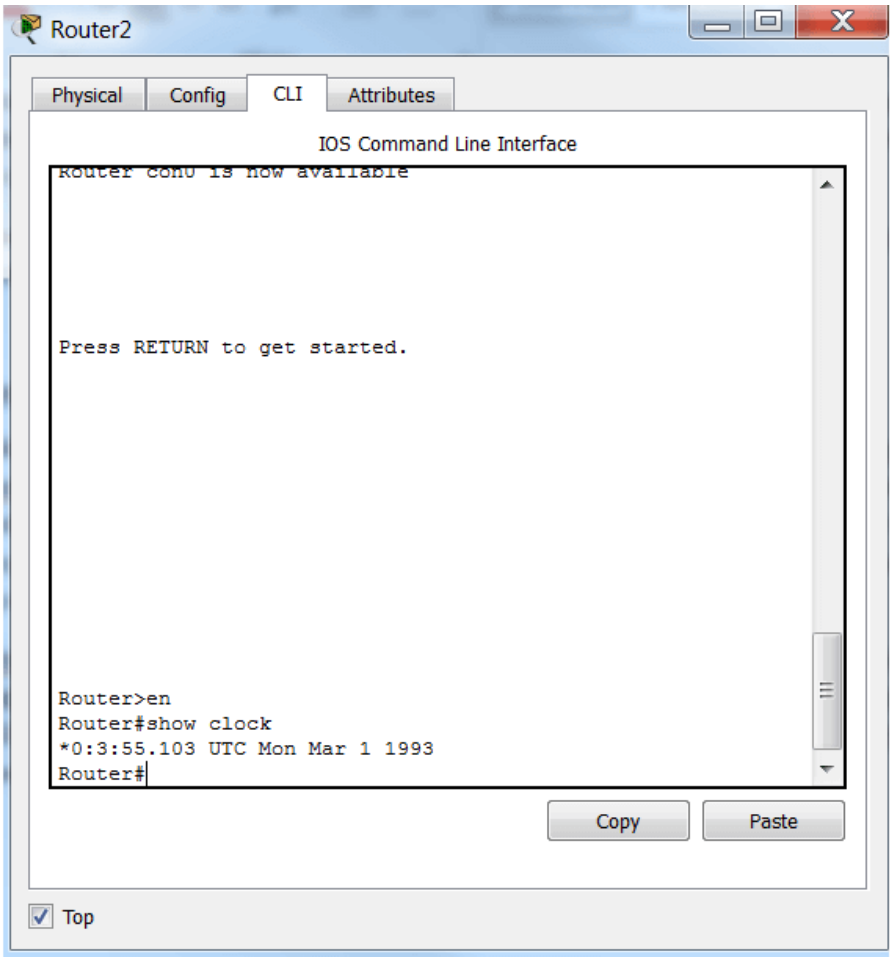
9. Настройка времени на Cisco: NTP и руками

Если вы забудете корректно настроить системное время на маршрутизаторах или коммутаторах Cisco, это может сыграть злую шутку. Просмотр лог – файлов или аудит в рамках безопасности может быть не реализуем, по причине невозможности установить точную дату события. В статье расскажем, как настроить корректную дату и время вручную, а также, как подключить NTP сервер к L2/L3 устройствам Cisco.

Ручная настройка

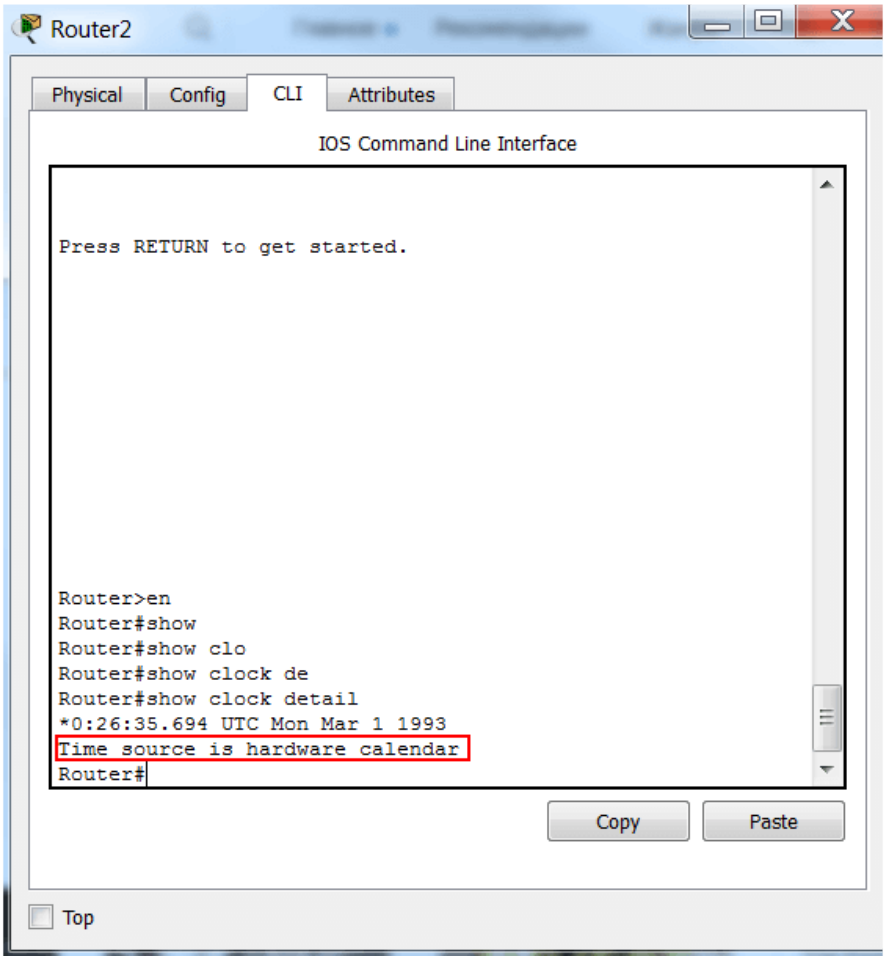
Устройства на базе Cisco IOS имеют два источника времени – железное/хардварное (hardware) и софтовое (программное) время. Первое, зачастую, в документации вендора именуется как **«calendar time»**. Программное время, при загрузке девайса (по питанию) тянет время из железного, ставя его важнее в приоритете. Давайте проверим этот момент с помощью Cisco Packet Tracer:

```
en
show clock
```



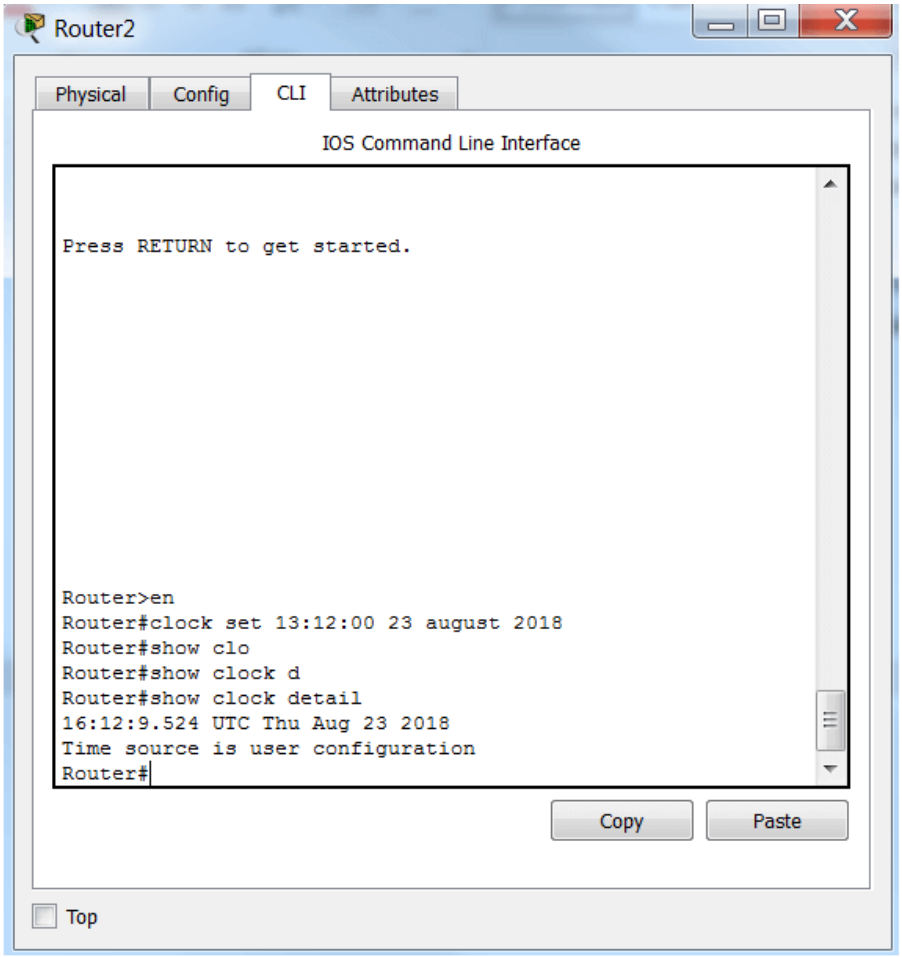
Обратите внимание, в нашем выводе, `*0:3:55.103 UTC Mon Mar 1 1993` помечена звездочкой сначала. Она говорит о том, что это время не вызывает доверия. Причина этого проста – оно синхронизировано с хардварного времени, это можно проверить командой `show clock detail`:

```
en
show clock
```



С помощью команды `clock set` (в привилегированном режиме, не в режиме глобальной конфигурации) мы можем в ручном режиме модифицировать время и дату:

```
en
conf t
clock set 13:12:00 23 august 2018
```



Обратите внимание, что источник времени сменился на «user configuration». Дело в том, что если мы перезагрузим наш девайс, время снова подтянется из хардварного источника (его можно проверить командой `show calendar`). Исправить это можно одной командой:

```
clock update-calendar
```

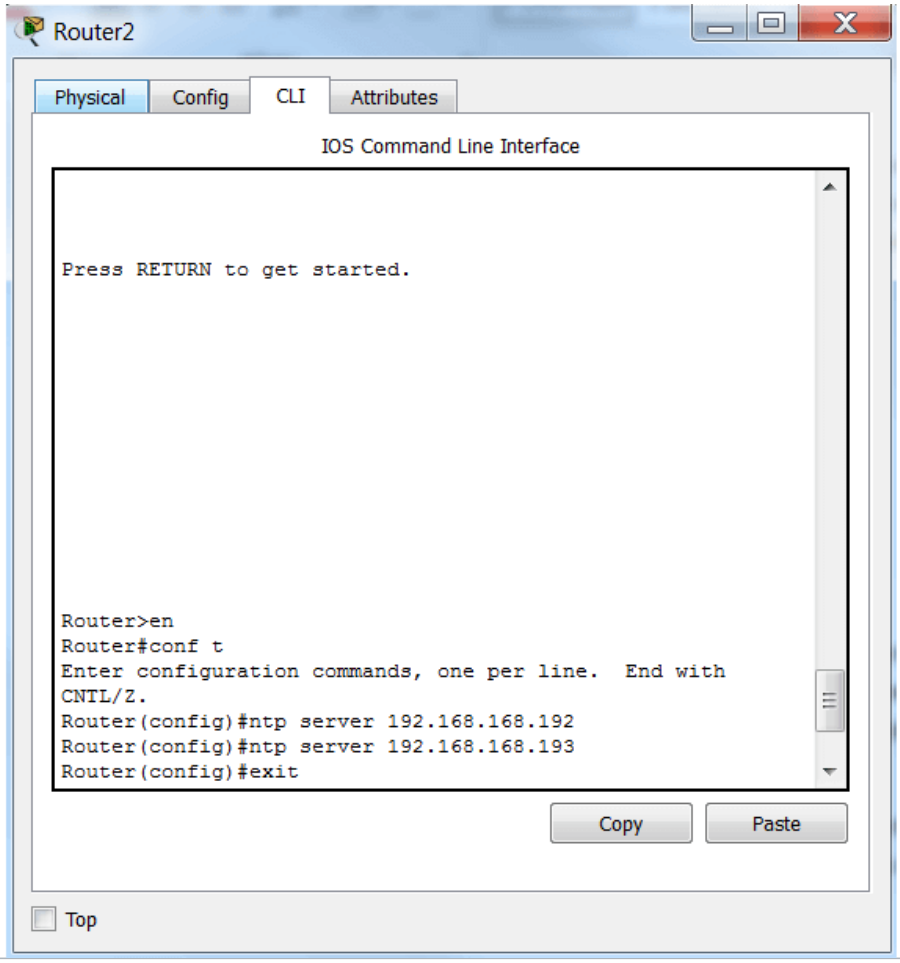
Готово :)

Лучший путь: настройка NTP

Дело в том, что бывают задачи, точность которых зависит от синхронизации сотых долей секунд на каждом из устройств в сети. В таком случае нам поможет синхронизация времени от единой точки по протоколу NTP (Network Time Protocol), а время они будут брать с NTP – сервера.

Перед настройкой, важно понять – откуда вы будете брать время. Есть некоторые публичные NTP, но конечно, гораздо безопаснее использовать сервер в собственном сетевом контуре. После того, как определитесь, приступаем к настройке NTP серверов:

```
en
conf t
ntp server 192.168.168.192
ntp server 192.168.168.193
```

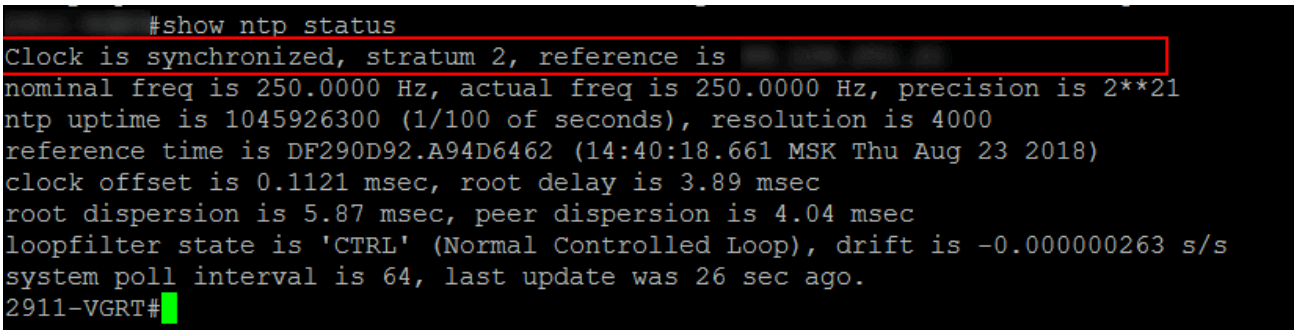


Далее, мы уходим из среды Cisco Packet Tracer на железный маршрутизатор Cisco 2911, так как программный эмулятор ограничен в командах :)

Ждем, пока время не будет синхронизировано и проверяем:

Вы можете отслеживать этапы синхронизации командой `show ntp associations` - команда будет полезна для траблшутинга NTP;

```
show ntp status
```



У нас статус **Clock is synchronized, stratum 2, reference is A.B.C.D**. Значит все работает хорошо. **Важно** - настройка NTP, которую мы описали в статье, касается только софтового (программного) времени. Для того, чтобы синхронизировать хардварное (железное) время даем команду:

```
ntp update-calendar
```

10. Повышаем безопасность коммутаторов и маршрутизаторов Cisco

Сетевая инфраструктура (роутеры, коммутаторы, МСЭ, АТС и так далее) являются очень важными ресурсами организации, и поэтому очень важно корректно настроить доступ к данным устройствам – для достижения нужного уровня защиты.

Множество корпораций фокусируются на защите своих серверов, приложений, баз данных и прочих компонентов сети, но они могут совершенно забыть о том, что часть установленных у них устройств содержат, к примеру, дефолтные логин и пароль.

К примеру, скомпрометированный маршрутизатор может доставить гигантское количество проблем – злоумышленники могут получить доступ к информации, трафик может улетать на другое направление и так далее. Так что корректная настройка устройств с точки зрения сетевой безопасности является крайне важным моментом при обеспечении защиты информации вашей организации.

К примеру Cisco разделяет любое сетевое устройство на 3 функциональных плоскости, а именно:

- **Плоскость менеджмента** – это все о том, как непосредственно управлять железкой. То есть данная плоскость используется для доступа, настройки и мониторинга устройства. В нашей статье мы непосредственно расскажем, как защитить данную плоскость;
- **Плоскость управления** – данная плоскость содержит в себе сигнальные протоколы и процессы, которые отвечают за связность между устройствами – например такие известные вам протоколы как OSPF, EIGRP и так далее;
- **Плоскость данных** – плоскость, ответственная за перемещение информации по сети от источника до ее назначения. В данной плоскости и происходит, как правило, обмен пакетами между устройствами;

Из этих трех плоскостей наиболее защитить первую и вторую плоскости, однако в нашей статье мы сконцентрируемся на плоскости менеджмента и обсудим 10 важных шагов по улучшению защищенности сетевого устройства Cisco с **IOS**.

Десять пунктов ниже не являются избыточными, но они включают в себя наиболее важные команды и настройки, которые позволят «закрыть» устройство от нежелательного доступа и повысить степень защищенности. Данные пункты применимы как к маршрутизаторам, так и к коммутаторам.

Создание секретного пароля

В целях предоставления доступа к IOS устройству только людям, имеющим право (например, сисадмину/эникею/инженеру) всегда нужно создавать сложный «секретный» пароль (enable secret). Мы советуем придумать/сгенерировать пароль минимум 12 знаков, содержащий цифры, буквы и специальные символы. Проверьте, что вы вводите именно

```
enable secret
```

- тогда в конфиге пароль будет отображаться в зашифрованном виде.

```
Router# config terminal
Router(config)# enable secret сложныйпароль
```

Зашифруйте пароли на устройстве

Все пароли, настроенные на устройстве (за исключением «секретного»), не шифруются от слова совсем и легко видны в конфиге. Чтобы зашифровать все пароли в конфиге, необходимо использовать глобальную команду

```
service password encryption
```

```
Router# config terminal
Router(config)# service password-encryption
```

Используйте внешний сервер авторизации для аутентификации пользователей

Вместо использования локальных учетных записей на каждом устройстве для доступа администратора, мы рекомендуем использование внешнего **AAA** сервера (TACACS+ или RADIUS) для обеспечения Аутентификации, Авторизации и Учета (вольный перевод Authentication, Authorization, Accounting).

С централизованным AAA сервером гораздо проще управлять учетными записями, реализовывать политики безопасности, мониторить использование аккаунтов и многое другое.

Ниже на схеме вы можете видеть как настроить **TACACS+** и **RADIUS** серверы с использованием enable secret пароля в случае отказа этих серверов.


```
TACACS+
Router# config terminal
Router(config)# enable secret K6dn!#scfw35    //создаем “секретный ” пароль
Router(config)# aaa new-model   //включаем AAA службу
Router(config)# aaa authentication login default group tacacs+ enable  //Используем TACACS
сервер и обычный пароль на случай отказа
Router(config)# tacacs-server host 192.168.1.10 //указываем внутренний AAA сервер
Router(config)# tacacs-server key ‘secret-key’ //указываем секретный ключ для AAA сервера
Router(config)# line vty 0 4
Router(config-line)# login authentication default //применяем AAA аутентификацию для линий
удаленного доступа (telnet, ssh)
Router(config-line)# exit
Router(config)# line con 0 //применяем AAA аутентификацию для консольного порта
Router(config-line)# login authentication default
RADIUS
Router# config terminal
Router(config)# enable secret K6dn!#scfw35    //создаем “секретный ” пароль
Router(config)# aaa new-model   //включаем AAA службу
Router(config)# aaa authentication login default group radius enable  //Используем RADIUS сервер
и обычный пароль на случай отказа
Router(config)# radius-server host 192.168.1.10 //указываем внутренний AAA сервер
Router(config)# radius-server key ‘secret-key’ //указываем секретный ключ для AAA сервера
Router(config)# line vty 0 4
Router(config-line)# login authentication default //применяем AAA аутентификацию для линий
удаленного доступа (telnet, ssh)
Router(config-line)# exit
Router(config)# line con 0 //применяем AAA аутентификацию для консольного порта
Router(config-line)# login authentication default
```

Создайте отдельные аккаунты для пользователей

Если у вас отсутствует возможность использовать внешний AAA сервер, по инструкции, описанной в предыдущем шаге, то как минимум, вам необходимо создать несколько отдельных локальных аккаунтов для всех, у кого должен быть доступ к устройству. Приведем пример создания трех локальных аккаунтов для троих системных администраторов.

> Кроме того, в версии IOS начиная с **12.2(8)T** и позднее, есть возможность настроить повышенную надежность паролей (Enhanced Password Security) для локальных учетных записей – это зашифрует пароли с помощью MD5 хэша.

Ниже пример настройки трех учетных записей:

```
Router# config terminal
Router(config)# username efstafiy-admin secret Lms!a2eZf*%_rete
Router(config)# username evlampiy-admin secret d4N3%sffeger
Router(config)# username vova-admin secret 54sxSFT*&_(!zsd
```

Настройте лимит возможных попыток подключения

Для того, чтобы избежать взламывания вашей учетной записи на маршрутизаторе с помощью брутфорса, вы можете настроить ограничение количества попыток подключения, когда после определенного предела система заблокирует пользователя. Это работает для локальных учетных записей.

```
Router# config terminal
Router(config)# username john-admin secret Lms!a2eZSf*%
Router(config)# aaa new-model
Router(config)# aaa local authentication attempts max-fail 5  //max 5 failed login attempts
Router(config)# aaa authentication login default local
```

Открытие доступа на управление устройством только для определенных IP – адресов

Данный пункт является одним из наиболее важных для сетевых устройств Cisco – необходимо оставить доступ к Telnel или SSH только для определенных сетевых адресов (например, рабочей станции системного администратора). В нашем примере сисадмин находится в пуле 192.168.1.0/28

```
Router# config terminal
Router(config)# access-list 10 permit 192.168.1.0 0.0.0.15
Router(config)# line vty 0 4
Router(config)# access-class 10 in  //применить ограничения на все VTY линии (SSH/Telnet)
```

Включить логирование

Логирование является очень полезной функцией для отслеживания, аудита и контроля инцидентов. Вы можете включить логирование во внутренний буфер устройства или на внешний лог-сервер. Вторая опция является более предпочтительной, так как вы можете хранить там больше информации и проще производить различного рода аналитику.

Всего существует 8 уровней логирования (от 0 до 7), каждый из которых делает лог более насыщенным деталями. Лучше всего избегать 7 уровень логирования (дебаг), т.к это может легко потратить все ресурсы вашего устройства.

Ниже пример, как включить логирование и на внешний сервер, и на сам девайс (можно использовать два варианта одновременно).

```
Router# config terminal
Router(config)# logging trap 6  //Включить 6 уровень логирования для логов, отправляемых на внешний сервер
Router(config)# logging buffered 5  //Включить 5 уровень логирования для логов, хранимых на самом девайсе
Router(config)# service timestamps log datetime msec show-timezone  //Включить таймстампы с милисекундной точностью
Router(config)# logging host 192.168.1.2 //Отправлять логи на внешний сервер
Router(config)# logging source-interface ethernet 1/0  //Использовать интерфейс Eth1/0 для отправки логов
```

Включение NTP (Network Time Protocol)

Данный шаг необходим для корректной работы логирования – т.к вам необходимо синхронизированное и точное системное время на всех сетевых устройствах, для правильного понимания ситуации при траблшутинге. Вы можете использовать как публичный, так и свой собственный **NTP** сервер.

```
Router# config terminal
Router(config)# ntp server 3.3.3.3
Router(config)# ntp server 4.4.4.4
```

Использование безопасных протоколов управления

По умолчанию, протоколом, с помощью которого можно управлять устройством является Telnet. Однако весь трафик передается в незашифрованном виде – поэтому предпочтительно использовать SSH.

> Важно – для использования SSH необходимо настроить хостнейм и доменное имя, а также сгенерировать SSH ключи. Также следует разрешить только протокол SSH на VTY линиях

Защитить SNMP доступ

Про SNMP мы писали в одной из наших статей – это протокол для управления сетью, который, однако, также может служить «дырой» для доступа в вашу сеть. Для защиты данного направления, вам необходимо установить сложную **Community String** (что-то вроде пароля для SNMP) и разрешить доступ только с определенных рабочих станций.

Давайте настроим две Community String – одну с правами на чтение, и другую с правами на чтение и изменение. Также добавим ACL с нужными сетевыми адресами.

```
Router# config terminal
Router(config)# access-list 11 permit 192.168.1.0 0.0.0.15
Router(config)# access-list 12 permit 192.168.1.12
Router(config)# snmp-server community Mer!0nET RO 11 //создание community string с правами на чтение и использование ACL 11 для SNMP доступа
Router(config)# snmp-server community Mer!0NeTRules RW 12 //создание community string с правами на чтение/запись и использование ACL 12 для SNMP доступа
```

Команды выше позволят сети сисадмина 192.168.1.0/28 иметь доступ на чтение и хосту 192.168.1.12 иметь полный доступ на SNMP чтение / запись к устройствам.