Чего должно быть достаточно: «show», «?»

Но мы же знаем что это не так, так что пройдёмся далее.

**Заметки:**

1. Соединить компьютер и маршрутизатор у меня так и не получилось, получилось только через switch.
2. Роутер предпочитаю брать 2911 – на его первый вопрос в терминале говорим «no»

Switch - 2960

**Куча левых команд, не относящихся прямо к настройке сети:**

1. *Hostname DeviceX* – просто задаёт имя
2. *Enable password cisco* – аналог пароля в linux
3. *Enable secret san-fran* – пароль супервизора (он может менять обычный пароль)
4. *Service password-encryption* – по умолчанию все пароли храняться в незашифрованном виде, и эта команда включает шифрование
5. *Banner* – просто указывает приветствие при входе на коммутатор
6. *Line console 0* - позволяет сохранить отдельно историю ввода команд
7. *Exec-timeout* – указывает, время, через которое будет отключена сессия при бездействии
8. *Logging synchronous* – вроде как – это даёт удобство при вводе, и вообще при работе в консоли.
9. *Password cisco*
10. *Login*
11. *Show running-config* – выдаёт полную конфигурацию (текущую)
12. *Show startup-config* – конфигурация при перезагрузке
13. *Copy running-config startap-config* – для сохранения config

**Назначение адреса в router и поднятие интерфейса:**

Router>enable

Router#config term

Router(config)#interface Gig0/0

Router(config-if)#ip addr 10.0.1.1 255.255.255.0

Router(config-if)#no shutdown

Router(config-if)#exit

**Назначение адреса в компьютере:**

Компьютер –> Desktop –> Ip Configuration

**Поднятие VLAN (+trunk + virtual interface)** (подробнее см. «Семинар 2.docx»)**:**

Не забудьте, что vlan с номером 1 – есть везде и по умолчанию всё находится в нём, поэтому выбирайте адреса начиная с 2.

Устанавливать trunk со стороны маршрутизатора не обязательно – и так всё работает. А вот со стороны switch – обязательно.

Конфигурация switch:

Switch>enable

Switch#configure terminal

Switch(config)#vlan 2

Switch(config-vlan)#name vlan2

Switch(config-vlan)#exit

Switch(config)#vlan 3

Switch(config-vlan)#name vlan3

Switch(config-vlan)#exit

Switch(config)#interface Fa0/1

Switch(config-if)#switchport access vlan 2 -- то, что придёт на этот интерфейс будет упаковано в vlan с номером 2

Switch(config-if)#exit

Switch(config)#interface Fa0/2

Switch(config-if)#switchport access vlan 3

Switch(config-if)#exit

Switch(config)#interface Fa0/3

Switch(config-if)#switchport mode trunk -- на этом интерфейсе летают пакеты из разных vlan

Switch(config-if)#no shutdown – теоритически это нужно указывать при выходе из каждого интрефейса, но в cisco такие настройки (точнее наверно у того коммутатора, который я настраивал (просто берём поновее) )

Switch(config-if)#exit

Switch(config)#

Конфигурация router:

Router>enable

Router#configure terminal

Router(config)#interface gig0/0

Router(config-if)#no ip address

Router(config-if)#no shutdown

Router(config-if)#exit

Router(config)#interface Gig0/0.1 -- подняли виртуальный интерфейс

Router(config-subif)#ip address 10.0.1.254 255.255.255.0

Router(config-subif)#encapsulation dot1Q 2 -- сказали, что всё, что из vlan с номером 2 на этом интерфейсе должно быть распаковано и обработано

Router(config-subif)#no shutdown

Router(config-subif)#exit

Router(config)#interface gig0/0.2

Router(config-subif)#ip address 10.0.2.254 255.255.255.0

Router(config-subif)#encapsulation dot1Q 3

Router(config-subif)#no shutdown

Router(config-subif)#exit

Router(config)#

**Безопасность портов (port-security)** (подробнее см. «Семинар 2.2.docx»)**:**

*Switch>enable*

*SwitchX#config term*

*SwitchX(config)#interface fa0/5* - вход в соответствующий интерфейс

*SwitchX(config-if)#switchport mode access* - включаем конфигурирование порта

*SwitchX(config-if)#switchport port-security* - включаем port-security

*SwitchX(config-if)#switchport port-security maximum 1* - говорим, что максимальное количество мак-адресов, которые будут пропускаться через switch – «1»

*SwitchX(config-if)#switchport port-security mac-address 00d0.58ad.cb1f* – указываем адрес (причём этот адрес будет задан не жёстко, т.е. по прошествии некоторого времени (если оно будет указано) адрес сбросится и возьмётся тот, который первый) (для указания, что этот mac-адрес указан жёстко, нужно слово *sticky*)

*SwitchX(config-if)#switchport port-security violation shutdown* - указываем действие в случае нарушения (в данном случае shutdown)

*SwitchX(config-if)#switchport port-security aging time 10* - указываем, что адреса будут устаревать со скоростью 10 минут (у меня на cisco packet tracer команда не опозналась, препод не знает, в чём дело, так что увы)

**Настройка маршрутизации (в ручную)** (подробнее см «Семинар 3.docx»)**:**

Задание (“topology – done.pkt”): взять topology.pkt и настроить руками маршрутизацию так, чтобы пинговалось между компьютерами.

Причём прописывать правила нужно в обе стороны

Нужно прописать команду ip route в каждом роутере, чтобы работало:

RouterA>enable

RouterA#config term

RouterA(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.2

RouterA(config)#exit

RouterB>enable

RouterB#conf term

RouterB(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.3

RouterB(config)#ip route 172.16.0.0 255.255.0.0 10.1.1.1

RouterB(config)#exit

RouterC>enable

RouterC#conf term

RouterC(config)#ip route 172.16.0.0 255.255.0.0 10.2.2.2

RouterC(config)#exit

**Настройка маршрутизации (через rip – routing ip protocol)** (подробнее см «Семинар 3.docx»)**:**

Фактически роутер, настроенный на rip – выкидывает таблицы маршрутизации в те network, которые были указаны, т.е. он их соединяет (по умолчанию маршрутизация проходит в пределах одной подсети)

Задание (“topology - done2.pkt”) – настроить rip, а потом в таблице посмотреть, сколько до кого hop-ов при помощи show ip protocols

RouterA>enable

RouterA#conf term

RouterA(config)#router rip

RouterA(config-router)#network 172.16.0.0

RouterA(config-router)#network 10.1.0.0

RouterA(config-router)#exit

RouterA(config)#exit

RouterA#

RouterB>enable

RouterB#conf term

RouterB(config)#router rip

RouterB(config-router)#network 10.1.0.0

RouterB(config-router)#network 10.2.0.0

RouterB(config-router)#exit

RouterB(config)#exit

RouterB#

RouterC>enable

RouterC#conf term

RouterC(config)#router rip

RouterC(config-router)#network 10.2.0.0

RouterC(config-router)#network 192.168.1.0

RouterC(config-router)#exit

RouterC(config)#exit

RouterC#

Прочие доп. команды:

1. show ip protocols – расскажет о настройках
2. show ip route – покажет таблицу маршрутизации
3. debug ip rip – после этой команды роутер начнёт рассказывать, когда ему приходят сообщения из сети

**Настройка маршрутизации (через ospf)** (подробнее см «Семинар 3.docx»)**:**

Тут везде будет указана одна и таже эрия с номером 1 и в пределах этой эрии построится единая таблица маршрутизации

Как настраивать:

Router ospf <process-id>

Network <wildcard-mask> area area-id

wildcard-mask – это просто маска из нулей и единиц – например может быть 255.0.255.0

(в этих масках – где есть единица – то не важно) (эти маски сделаны для того, чтобы одной маской можно было покрыть несколько подсетей)

Задание (“topology – done3.pkt”) – нужно настроить в тополонии протокол ospf

(нужно задать wildmask и задать там одну зону)

RouterA>enable

RouterA#conf term

RouterA(config)#router ospf 100

RouterA(config-router)#network 172.16.1.1 0.0.0.0 area 1

RouterA(config-router)#network 10.1.1.1 0.0.0.0 area 1

RouterA(config-router)#exit

RouterA(config)#exit

RouterB>enable

RouterB#conf term

RouterB(config)#router ospf 104

RouterB(config-router)#network 10.1.1.2 0.0.0.0 area 1

RouterB(config-router)#network 10.2.2.2 0.0.0.0 area 1

RouterB(config-router)#exit

RouterB(config)#exit

RouterC>enable

RouterC#conf term

RouterC(config)#route ospf 113

RouterC(config-router)#network 10.2.2.3 0.0.0.0 area 1

RouterC(config-router)#network 192.168.1.1 0.0.0.0 area 1

RouterC(config-router)#exit

RouterC(config)#exit

**Настройка standart ACL:**

Помним, что маска в ACL – инвертированна: значимые биты – там, где «0»

В стандартном ACL запрет идёт лишь по source ip

Задание:

Изолируем PC5 так, чтобы он не мог даже пальцем тронуть остальных

Router>enable

Router#conf term

Router(config)#access-list 1 deny host 10.0.1.13

Router(config)#access-list 1 permit any

Router(config)#interface Gig0/0

Router(config-if)#ip access-group 1 in

Router(config-if)#exit

Router(config)#

**Настройка расширенного, именованного ACL:**

Задание – изолировать PC5, так, чтобы telnet – не проходил, используя нумерованный ACL

И изолировать PC7 так, чтобы соединение telnet по порту 1234 не проходило никуда, используя именованный ACL

show access-lists – покажет существующие ACL

no access-list ? – удаляет ACL

Router>enable

Router#config term

Router(config)#access-list 101 deny tcp host 10.0.1.13 any eq telnet

Router(config)#access-list 101 permit ip any any

Router(config)#interface Gig0/0

Router(config-if)#ip access-group 101 in

Router(config-if)#exit

Router(config)#

Расширенный ACL :

Router(config)#ip access-list extended trouble\_acl

Router(config-ext-nacl)#deny tcp host 10.0.3.13 any eq 1234

Router(config-ext-nacl)#permit ip any any

Router(config-ext-nacl)#exit

Router(config)#interface Gig0/2

Router(config-if)#ip access-group trouble\_acl in

Router(config-if)#exit

Router(config)#

**Настройка NAT:**

show ip nat translations – показывает текущие трансляции

debug ip nat

show ip nat statistics

clear ip nat translation – очищает таблицу трансляции

Для статического NAT – просто указывается, что мы такой-то ip подменяем на такой-то, необходимые команды:

(config) ip nat inside source static local-ip global-ip

(config-if) ip nat inside

(config-if) ip nat outside

(количество используемых адресов 1 к 1 – это не PAT)

Для NAT с перегрузкой нужно создать ACL – который указывает, какие соединения должны прокидывыться через NAT, остальное под юрисдикцию NAT не подпадает.

Задание – нужно поместить PC5 и PC7 полностью под NAT с ip-адресом сети 10.0.2.0

Router>enable

Router#conf term

Router(config)#ip access-list standard nat\_acl

Router(config-std-nacl)#permit 10.0.3.0 0.0.0.255

Router(config-std-nacl)#permit 10.0.1.0 0.0.0.255

Router(config-std-nacl)#exit

Router(config)#interface Gig0/0

Router(config-if)#ip nat inside

Router(config-if)#exit

Router(config)#interface Gig0/2

Router(config-if)#ip nat inside

Router(config-if)#exit

Router(config)#interface Gig0/1

Router(config-if)#ip nat outside

Router(config-if)#exit

Router(config)#

Router(config)#ip nat inside source list nat\_acl interface Gig0/1 overload

Router(config)#

Как понять, что NAT заработал – нужно просто включить режим «online» и послать ping в 10.0.2.13, потом, когда пакет дойдёт до компьютера назначения – щёлкнуть по нему и выскочат его подробности, в частности там будет виден source-ip – который будет подменён на 10.0.2.1 (без NAT там стоял бы адрес отправки настоящего компьютера)

Ещё можно просто сделать «show ip nat translations» - и там просто покажут. Что и куда оттранслировалось.