АКАДЕМИЯ IT

DarkMaycal

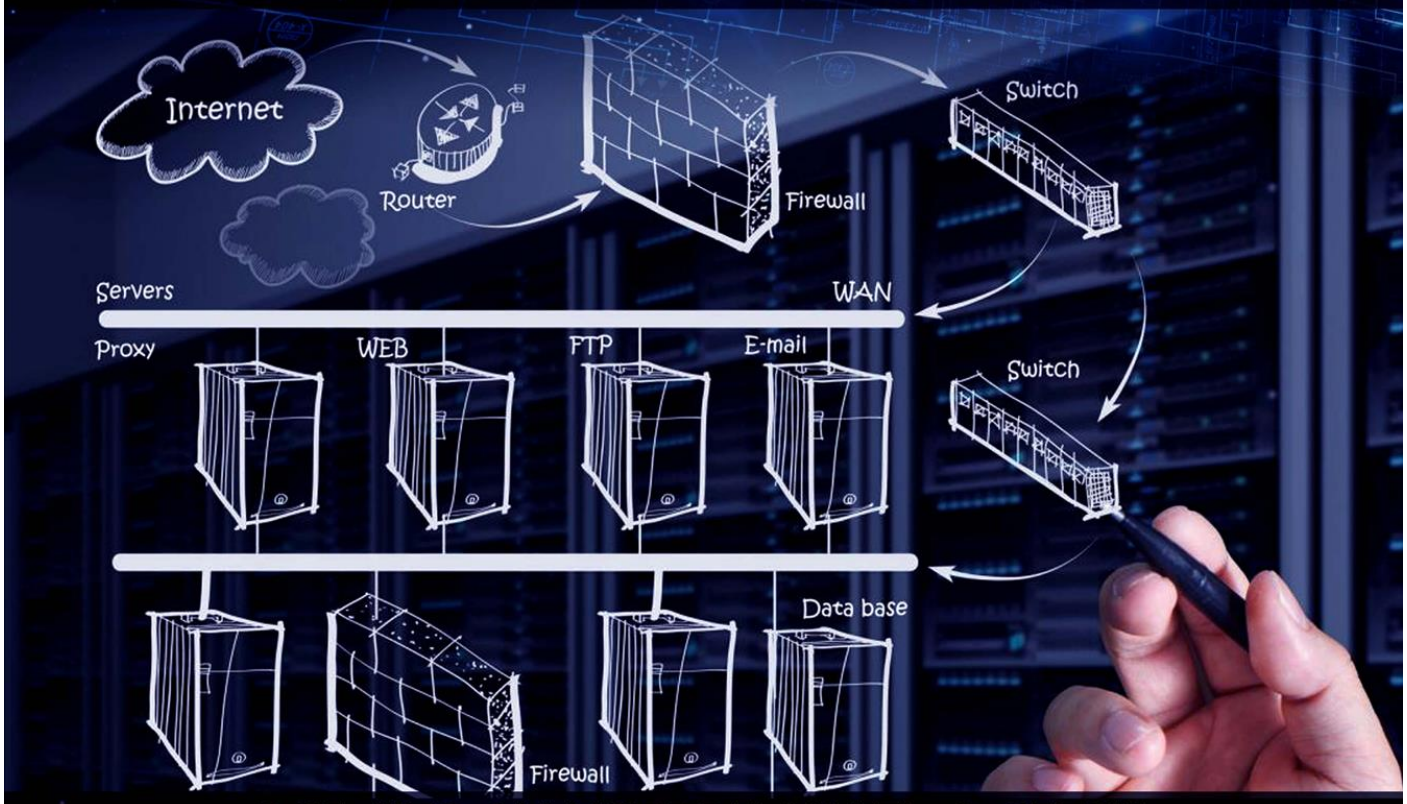
Sysadmins

ПОСТРОЕНИЕ СЕТЕЙ

CISCO

С НУЛЯ

от простой одноранговой сети
до трехуровневой иерархической модели



ВСЕ, ЧТО ВЫ ХОТЕЛИ ЗНАТЬ О СЕТЯХ

КОМАНДЫ CISCO

Часть I

Содержание

1. Общие команды

Router>	**enable**	перейти в привилегированный режим
Router#	**show running-config**	показать текущую (запущенную) конфигурацию
Router#	**show startup-config**	показать как устройство будет работать после перезагрузки
Switch(config)#	**hosname switch**	изменить имя хоста на switch
Router#	**configure terminal**	перейти в режим конфигурации оборудования
Router(config)#	**interface fastethernet 0/0**	перейти к настройке интерфейса fast ethernet 0/0
Router(config)#	**interface range fastethernet 0/0 - 3**	перейти к настройке пачки интерфейсов от 0/0 до 0/3
Router(config-if)#	**shutdown**	выключить интерфейс (no shutdown включить интерфейс)
Router(config-if)#	**ip address 192.168.0.1 255.255.255.0**	установить ip адрес на интерфейс
Router(config-if)#	**ip address dhcp**	получить ip адрес на интерфейс по DHCP
Router(config)#	**ip route 192.168.30.0 255.255.255.0 192.168.20.50**	прописать маршрут
Router(config)#	**ip name-server 192.168.0.1**	прописать dns сервер
Router(config)#	**no service config**	убрать сообщения вида «Error opening tftp://255.255.255.255/networkconfg (Timed out)» (работает с последующей перезагрузкой маршрутизатора)
Router(config)#	**cdp timer 5**	изменение частоты отправки пакетов CDP (в секундах)
Router(config)#	**cdp holdtime 10**	через сколько секунд признавать соседа недоступным
Router(config)#	**no ip cef**	глобальное отключение механизма Cisco Express Forwarding (на всех интерфейсах)
Router(config-if)#	**no ip route-cache cef**	отключение механизма Cisco Express Forwarding на интерфейсе
Router(config)#	**service compress-config**	сжимать running-config при каждом его сохранении в nvram
Router(config-if)#	**ip proxy-arp**	включить технологию Proxy ARP
Router#	**configure replace nvram:startup-config**	поместить содержимое startup-config в running config. В этом случае, содержимое текущего running config полностью заменится содержимым startup-config. При использовании команды copy, данные из файла startup-config будут добавляться к running-config
Router(config)#	**end**	выйти из режима конфигурирования
Router#	**show ip interface fastethernet 0/0**	показать конфигурацию интерфейса с точки зрения сетевого уровня (ip)
Router#	**show interface fastethernet 0/0**	показать конфигурацию интерфейса с точки зрения канального уровня
Switch#	**show interface status (свитч)**	показать статусы всех портов свитча

Router#	**show ip interface brief**	показать все интерфейсы и их ip адреса
Router#	**show cdp neighbors**	показать информацию о соседнем оборудовании по протоколу cdp
Router#	**show ip route**	показать таблицу маршрутизации
Router#	**show ip arp**	показать таблицу соответствия ip и MAC адресов
Router#	**copy running-config startup-config (write)**	сохранить рабочую конфигурацию
Router#	**debug ip nat**	отслеживание nat в реальном времени (может перегрузить процессор)
Router#	**show logging**	показать последние события
Router#	**terminal monitor**	передавать все сообщение из консоли (например дебаг) в telnet или ssh
Router#	**show errdisable recovery**	посмотреть порты, которые находятся в состоянии errdisable
Router(config)#	**ip name-server 192.168.1.1**	задать ip DNS сервера
Router#	**show control-plane host open-ports**	показать открытые порты на устройстве
Router(config)#	**show processes cpu**	показать загрузку CPU
Router(config)#	**logging synchronous**	по умолчанию журнальные сообщения могут выводиться в независимости от того набирает пользователь какие либо команды или нет, прерывая исполнение текущих команд. Включая эту команду, маршрутизатор начинает дожидаться завершения текущей команды и вывода ее отчета.
Switch#	**show mac address-table int fa0/1**	показать MAC адреса за интерфейсом fast ethernet 0/1
Switch#	**show ip arp**	показать таблицу ARP
Switch(config-if)#	**no switchport**	перевод порта свитча в роутерный режим
Switch#	**show file system**	показать сколько осталось flash и nvram памяти
Switch#	**show memory**	показать сколько осталось оперативной памяти
Switch#	**show flash:**	показать содержимое flash памяти
Switch#	**show license**	показать лицензии, установленные на устройстве
Switch#	**show platform tcam utilization**	показать оставшиеся ресурсы таблицы TCAM (сколько маршрутов еще можно добавить в этот L3 свитч или роутер, сколько ACL листов, сколько QoS и так далее)
Switch#	**show sdm prefer**	посмотреть какой SDM шаблон используется (как используется память TCAM)
Switch#	**show lldp (cdp)**	показать состояние протокола lldp или cdp (включен/выключен)
Switch#	**show lldp neighbords**	показать соседние устройства по lldp
Switch(config)#	**sdm prefer dual-ipv4-and-ipv6 default**	переключить использование TCAM памяти на работу как с ipv4 так и ipv6
Switch(config)#	**lldp run**	включить LLDP глобально
Switch(config-if)#	**lldp enable**	включить LLDP на интерфейсе

Switch(config)#	no cdp run	выключить CDP глобально
Switch#	copy startup-config ftp://Maycal:HZmLr16N@172.10.1.2/Core_Switch_2.txt	скопировать startup-config на ftp, где Maycal - логин, HZmLr16N – пароль, 172.10.1.2 – адрес сервера, Core_Switch_2.txt – имя файла
Switch#	copy ftp://Maycal:HZmLr16N@172.10.1.2/Core_Switch_2.txt startup-config	обратная операция, скопировать конфигурацию из файла Core_Switch_2.txt и поместить её в startup-config

2. Конфигурация EIGRP

Конфигурация:

Router#	router eigrp 1	перейти в контекст настройки eigrp
Router(config-router)#	no auto-summary	перевести EIGRP в бесклассовый режим (работа по маскам) + отключить автоматическое суммирование маршрутов
Router(config-router)#	network 192.168.1.0 0.0.0.255	анонсировать сеть 192.168.1.0 0.0.0.255 всем другим участникам
Router(config-router)#	network 192.168.2.0 0.0.0.255	анонсировать сеть 192.168.2.0 0.0.0.255 всем другим участникам
Router(config)#	int fa0/0	перейти к редактированию интерфейса fast ethernet 0/0
Router(config-if)#	ip hellow-interval eigrp 1 10	посылать hellow пакеты каждые 10 секунд
Router(config-if)#	ip hold-timer eigrp 1 30	признать соседа недействительным, если мы от него не получили hellow пакеты в течении 30 секунд
Router(config)#	router eigrp 1	перейти в контекст настройки eigrp
Router(config-router)#	eigrp 1 stub [connected | receive-only | redistributed | static | summary]	получать все маршруты, но анонсировать только коннектед-сетки и суммированные маршруты (можно указать и другие параметры, например не анонсировать вообще никакие маршруты, а получать все)
Router#	debug eigrp	включение вывода информации о процессах eigrp для его отладки (после просмотра дебага - undebug all)
Router(config-if)#	ip summary-address eigrp 1 10.64.0.0/14	анонсировать по интерфейсу суммированный маршрут
Router(config-router)#	metric weights <TOS> <K1> <K2> <K3> <K4> <K5>	изменение K-коэффициентов. У каждого коэффициента диапазон значений от 0 до 255. TOS — Type Of Service. Диапазон значений от 0 до 8. Поддерживается только значение 0.
Router(config-router)#	metric maximum-hops <1-255>	изменение max-hop. По умолчанию у EIGRP значение maximum-hops равно 100. То есть, маршрут, который достижим через 100 hop'ов считается недостижимым.

Router(config-router)#	**traffic-share balanced**	без дополнительных настроек с помощью команды traffic-share, EIGRP балансирует нагрузку между маршрутами независимо от метрики. Включение балансировки по принципу, чем меньше метрика маршрута, тем больше передается по нему трафика (обратно пропорционально метрике)
Router(config-router)#	**traffic-share min**	несмотря на то, что есть несколько маршрутов, отправлять трафик только по маршрутам с наименьшей метрикой
Router(config-router)#	**variance 2**	включение балансировки между маршрутами с разной метрикой. Значение variance может быть от 1 до 128. Это множитель, указывающий во сколько раз основной маршрут будет лучше, чем feasible successor. В этом примере, если feasible successor будет не дороже, чем в 2 раза по сравнению с основным маршрутом successor, то он будет использоваться для балансировки трафика.
Router(config-router)#	**maximum-paths 6**	изменение количества маршрутов с одинаковой метрикой, между которыми выполняется балансировка нагрузки. Для того чтобы отключить балансировку нагрузки, надо установить количество маршрутов равным 1
Router(config-router)#	**neighbor 192.168.2.0**	вручную прописать соседа (используется в NBMA сетях)
Router(config-if)#	**no ip split-horizon eigrp 100**	отключение функции **split-horizon на интерфейсе**

Таймеры на двух соседних роутеров не обязаны совпадать, в отличие от OSPF.

Диагностика:

Router#	**show ip eigrp neighbors**	Посмотреть роутеры, с которыми у нас установлено соседство по eigrp
Router#	**show ip eigrp interfaces**	Посмотреть интерфейсы, которые включены в анонс eigrp
Router#	**show ip eigrp topology (all-links)**	Посмотреть топологию eigrp
Router#	**show ip protocols**	показать конфигурацию динамических протоколов (по какой сети работают, какой id имеют и так далее)
Router#	**show ip eigrp interface detail fa0/0**	посмотреть значение таймеров на интерфейсе
Router#	**show ip eigrp traffic**	показать счетчики трафика eigrp (сколько через eigrp прошло ip пакетов)

Настройка аутентификации по ключевым цепочкам:

Прежде обязательна установка правильного времени на всех роутерах. Например настройка NTP.

Router(config)#	key chain MYKEYS	создать ключевую цепочку с названием MYKEYS
Router(config)#	key 1	в ключевой цепочке создаем ключ 1
Router(config)#	key-string cisco	задаем значение ключа 1 как cisco
Router(config)#	accept-lifetime 18:00:00 may 21 2015 18:00:00 may 22 2015	указываем промежуток времени, в течении которого будем принимать ключ от соседа
Router(config)#	send-lifetime 18:00:00 may 21 2015 18:00:00 may 22 2015	указываем промежуток времени, в течении которого будем посылать ключ соседу
Router(config)#	key 2	когда закончиться действие ключа 1, автоматически начинается действие ключа 2. Создаем его
Router(config)#	key-string cisco2	задаем значение ключа 2 как cisco2
Router(config)#	accept-lifetime 17:00:00 may 22 2015 18:00:00 may 23 2015	указываем промежуток времени, в течении которого будем принимать ключ от соседа
Router(config)#	send-lifetime 17:00:00 may 22 2015 18:00:00 may 23 2015	указываем промежуток времени, в течении которого будем посылать ключ соседу
Router(config)#	int e0/1	переходим к редактированию интерфейса ethernet 0/1
Router(config-if)#	ip authentication mode eigrp 1 md5	включаем аутентификацию eigrp за экземпляр 1
Router(config-if)#	ip authentication key-chain eigrp 1 MYKEYS	указываем ключевую цепочку, которую будем использовать для аутентификации. В данном случае MYKEYS

3. Конфигурация OSPF

Правила анонсирования сетей в OSPF. Для того, чтобы 2 роутера увидели друг друга и обменялись маршрутами нужно:

1. Анонсировать те сети, по которым они соединены друг с другом. При этом они установят ТОЛЬКО СОСЕДСТВО.

2. Анонсировать те сети, про которые мы хотим рассказать соседу. Только тогда второй роутер увидит сеть «за спиной» первого роутера

Общая конфигурация:

Router(config)#	**router ospf 1**	перейти в режим настройки ospf экземпляра 1
Router(config)#	**router-id 0.0.0.32**	назначить id роутеру
Router(config-router)#	**network 192.168.0.0 0.0.255.255 area 0**	Анонсировать сеть 192.168.0.0 0.0.255.255 в регион 0
Router(config-router)#	**passive-interface fa0/0.2**	не посылать hellow пакеты по интерфейсу fa0/0.2 (используется в интерфейсах, смотрящих на пользователей, за которыми точно нет роутеров)
Router(config-router)#	**auto-cost reference-bandwith**	10000 - изменить реферсную полосу пропуская (в мегабитах) (нужно делать на всех роутерах)
Router#	**clear ip ospf 1 pro**	сбросить маршрутную информацию
Router(config-if)#	**ip ospf hellow-interval 8**	посылать hellow-пакеты каждые 8 скунд
Router(config-if)#	**ip ospf dead-interval 30**	признавать соседа недоступным, если мы от него не получили hellow пакет через 30 секунд
Router(config-if)#	**ip ospf priority 100**	изменить приоритет для OSPF на интерфейсе (влияет на выбор DR/BDR)
Router(config-if)#	**ip mtu 1400**	изменить размер MTU (необходимо выполнять на всех роутерах в одной канальной среде)

Диагностика:

Router#	**show ip ospf database**	показать LSA от соседних роутеров
Router#	**show ip ospf database router 1.1.1.1**	Показать детальное содержимое LSA, которую сгенерировал роутер с ID 1.1.1.1
Router#	**show ip protocols**	показать конфигурацию динамических протоколов (по какой сети работают, какой id имеют и так далее)
Router#	**show ip ospf database router (summary) 172.16.3.50**	показать содержание самой LSA типа router (summary) с id 172.16.3.50
Router#	**show ip ospf interface (brief)**	показать интерфейсы, на которых работает OSPF (brief выводит сокращенную информацию,

		но в PacketTracer не работает), кроме того, выводит информацию о стоимости интерфейса
Router#	**show ip ospf neighbor**	показать соседние роутеры, с которыми наш роутер обменивается маршрутами
Router#	**show ip ospf route**	показать список маршрутов, которые получил ospf и должен был передать в таблицу маршрутизации. Внимание! Не путать со следующей командой
Router#	**show ip route ospf**	показать таблицу маршрутизации и сделать выборку по тем маршрутам, которые попали в таблицу маршрутизации
Router#	**show ip route 192.168.3.2**	показать детальную информацию по маршруту к 192.168.3.2

Изменение стоимости интерфейса:

Router(config)#	**int e0/1**	заходим на нужный интерфейс
Router(config-if)#	**ip ospf cost 12**	устанавливаем стоимость интерфейса равной 12

Дополнительно:

Router(config-router)#	**auto-cost reference-bandwidth 100000000**	изменить референсную стоимость полосы пропускания (для правильного расчета стоимости маршрутов свыше 100 мегабит)

Внимание! Эта настройка должна быть одинакова у всех роутеров. Стоимость маршрута определяется по простой формуле:

референсная полоса пропуская (reference bandwidth) / скорость интерфейса (interface bandwidth).

Указывается в килобитах. По умолчанию референсная полоса пропускания равна 100000

Кроме того, можно поменять расчетную скорость интерфейса (reference bandwidth). Указывается в килобитах:

Router(config-router)#	**bandwidth 10000**	установить расчетную скорость интерфейса в 10000 килобит

Это может использоваться в сценарии, когда физический интерфейс 10 гигабит (данные взяты для примера), а скорость на нем только 45 мегабит (сценарий подключение к провайдеру). И вот для того, чтобы маршрут оценивался адекватно (по фактической скорости а не по скорости интерфейса) и используется функция, описанная выше.

3.1 Фильтрация маршрутов

Задача 1:

запретить анонс сетей 10.0.1.0 10.0.2.0 10.0.3.0 .... из региона Area 0 (они в нем зародились) в регион Area 1 (а по умолчанию в регион Area 1 попадает все то, что есть в регионе Area 0)

Решение:

Решение идет от обратного - мы разрешаем те сети, которые хотим анонсировать из региона 0 в регион 1, а все остальные по умолчанию будут запрещены.

Router(config)#	**ip prefix-list OSPF permit 192.168.0.0/16**	создаем prefix-list и разрешаем подсеть 192.168.0.0/16
Router(config-router)#	**area <area-id> filter-list prefix <prefix-name> <in | out>**	применяем этот фильтр на ABR роутер

in — фильтрация сетей, которые передаются в указанную зону

out — фильтрация сетей, которые передаются из указанной зоны

Таким образом, на роутере из региона 1 мы уже не увидим сетей 10.0.1.0, 10.0.2.0, 10.0.3.0....., а у видим только разрешенную - 192.168.0.0.

Внимание! Это не распространяется на дефолтный маршрут 0.0.0.0 - он все равно попадет на другой роутер. Представим ситуацию - мы развернули туннель с удаленным филиальным роутером и установили OSPF соседство с ним. В этой ситуации филиальный роутер будет ходить в интернет не через обычный маршрут, а через этот туннель, поскольку наш роутер в главном филиале по OSPF анонсирует ему дефолтный маршрут 0.0.0.0. Для того, чтобы исправить эту ситуацию можно ухудшить метрику для внешних ospf маршрутов:

Fil_Edge_Router1(config-router)#	**distance ospf external 255**	установить метрику для внешних ospf маршрутов равной 255

Благодаря этому, в таблице маршрутизации филиального роутера Fil_Edge_Router1 останется статический дефолтный маршрут, а дефолтный маршрут пришедший от роутера в главном офисе будет проигнорирован.

Задача 2: разрешить роутеру из региона 1 увидеть еще одну подсеть 10.0.2.0. Для этого на ABR роутере добавляем:

Router (config)#	ip prefix-list OSPF permit 10.0.2.0/16	создаем prefix-list и разрешаем еще одну подсеть 10.0.2.0/16

В итоге на роутере из Area 1 мы увидим 2 маршрута:

до 192.168.0.0

и до 10.0.2.0

Альтернативный способ фильтрации маршрутов:

Задача:

Нам нужно разрешить роутеру вбрасывать в свою таблицу маршрутизации только определенные маршруты, чтобы злоумышленники не могли вбросить какой-либо неверный маршрут. Для этого создадим ACL:

Fil_Edge_Router(config)#	access-list 3 permit 172.10.0.0 0.0.255.255	создаем access-list 3 и разрешаем подсеть 172.10.0.0 0.0.255.255
Fil_Edge_Router(config)#	access-list 3 permit 30.0.0.0 0.0.255.255	создаем access-list 3 и разрешаем подсеть 30.0.0.0 0.0.255.255
Fil_Edge_Router(config)#	access-list 3 permit 20.0.0.0 0.0.255.255	создаем access-list 3 и разрешаем подсеть 20.0.0.0 0.0.255.255
Fil_Edge_Router(config)#	access-list 3 permit 192.168.0.0 0.0.255.255	создаем access-list 3 и разрешаем подсеть 192.168.0.0 0.0.255.255
Fil_Edge_Router(config)#	router ospf 1	заходим в контекст конфигурации ospf
Fil_Edge_Router(config-router)#	distribute-list 3 in	применяем созданными нами ACL к движку ospf

Таким образом, в таблицу маршрутизации филиального роутера Fil_Edge_Router попадут только указанные маршруты в ACL 3.

Разница между первым и вторым способом – в первом случае мы запрещали одному роутеру анонсировать маршруты другому роутеру. Во втором случае мы запрещаем роутеру получать маршруты.

*Маршрутизатор фильтрует маршруты, которые помещаются в таблицу маршрутизации, но LSDB остается неизменной

Фильтрация маршрутов при редистрибуции:

Router (config)#	ip prefix-list FILT deny 172.10.2.0/16	создаем prefix-list и разрешаем подсеть 10.0.2.0/16, которую будем вбрасывать в OSPF
Router (config)#	router ospf 1	переходим к контекст настройки протокола ospf экземпляра 1
Router (config-router)#	redistrubute eigrp 1 subnets	включаем редистрибуцию из eigrp в ospf
Router (config-router)#	destribute-list prefix FILT out eigrp 1	применяем prefix-list на маршруты, которые из EIGRP вбрасываются в OSPF

3.2 Суммаризация маршрутов

На ABR роутере:

Router(config-router)#	area 1 range 192.168.0.0 255.255.0.0	разослать в другие регионы суммированный маршрут 192.168.0.0 за регион 1

На ASBR роутере:

Router(config-router)#	summary-address 192.168.0.0 255.255.0.0	разослать от ASBR суммированный маршрут 192.168.0.0

3.3 Редистрибуция маршрутов

Способ 1: разослать все роутером дефолтный маршрут. То есть по всем неизвестным ip адресам все роутеры будут идти к нам:

Router(config-router)#	default-information originate always | metric | metric-type | route-map	вбрасываем всем соседям дефолтный маршрут до нас (через нас можно ходить в интернет). Делается на ASBR. Для того, чтобы эта команда сработала, на ASBR роутере нужно прописать статический маршрут 0.0.0.0 0.0.0.0 и замкнуть его на любой интерфейс, либо к команде default-information originate добавить always

Способ 2: разослать всем роутерам все маршруты до всех внешних сетей которые мы сами знаем. Нельзя использовать если сетей очень много (нельзя в OSPF вбрасывать BGP)

Router(config-router)#	**redistribute connected (static,eigrp, bgp и т.д) subnets**	вбросить в OSPF коннектед-сетки (статические маршруты, маршруты eigrp, bgp и т.д) ASBR роутера. Аналогично выполняется и для EIGRP
Router(config-router)#	**redistrubute eigrp 1 metric-type 1 subnets**	вбросить в OSPF маршруты из EIGRP с учетом стоимости транзита по сети OSPF

Изменение административного расстояния:

Router(config-router)#	**distance ospf external | inter-area | intra-area 1-255**	изменить административное расстояние для всех OSPF маршрутов
Router(config-router)#	**distance 1-255 172.20.0.0 0.0.255.255**	изменить административное расстояние только для одно маршрута 172.20.0.0 с перевернутой маской 0.0.255.255

3.4 Регионы OSPF

Регион stub:

Если вбросить маршруты способом 2, то можно в некоторые регионы посылать не все маршруты, а заменить их дефолтным маршрутом. Это делает ABR роутер.

Для этого регион помечаем как stub. Например:

Router(config-router)#	**area 1 stub**	пометить регион 1 как stub

Это делается на всех роутерах, входящих в регион 1.

Таким образом, все другие регионы получат полноценные маршруты в интернет (зародившиеся на роутере ASBR), а регион 1 получит маршрут-восьми нулёвку.

Другие словами, роутеры из региона 1 начинают получать LSA не 5-того типа, а LSA 3-го типа

Регионы Stub и Totally Stubby Area:

Этот режим заменяет LSA 3-го типа и LSA 5-го типа одним единственным дефолтным маршрутом.

Для включения этого режима необходимо роутер из одного региона включить просто как stub, а на ABR роутере задать режим stub no-summary. Например:

Router(config-router)#	**area 2 stub**	помечаем регион 2 как stub (на роутере второго региона)
Router(config-router)#	**area 2 stub no-summary**	или помечаем регион как Totally Stubby Area

Таким образом, на роутерах из второго региона вброшенные маршруты от роутера ASBR (LSA 5-го типа) и внутренние маршруты из других регионов (LSA 3-го типа) будут в таблице маршрутизации замещены одним единственным дефолтным маршрутом.

Изменение стоимости маршрута для регионов:

Мы можем изменить стоимость дефолтного маршрута для определенного stub региона. Это может быть полезно если у нас есть два ABR роутера и мы хотим пропускать трафик больше через первого, чем через второго. На самом деле, поскольку OSPF не умеет балансировать трафик по маршрутам неравной стоимости (в отличие от EIGRP), трафик будет ходить только через тот ABR, который анонсировал маршрут меньшей стоимости

На ABR роутере:

Router(config-router)#	**area 2 default-cost 50**	определяем стоимость маршрута для региона 2 в 50

Внимание! По умолчанию, ASBR может находится только в нормальном регионе (не stub)

Однако есть обход этого запрета - stub регион нужно сделать типа NSSA. В таком случае, ASBR находящийся в NSSA регионе будет порождать не LSA 5-го типа, а LSA 7-го типа, которая на ABR будет преобразовываться в LSA 5-го типа.

На ABR роутере:

Router(config-router)#	**area 2 NSSA (no-autosammary)**	помечаем регион 2 как область NSSA (если добавить команду no-autosammary, то регион будет не просто NSSA, а еще и Totally Stubby Area)

И на всех роутерах 2-го региона

Router(config-router)#	area 2 NSSA (no-autosammary)	помечаем регион 2 как область NSSA (и Totally Stubby Area при задании команды no-autosammary)

3.5 Дополнительные фукнции

Дополнительно:

В случае работы по frame-relay у нас отсутствуют бродкасты и мультикасты. Поэтому соседей нужно прописывать вручную (они не могут друг друга автоматически обнаружить разослав мультикастовый hellow-пакет):

Router(config-router)#	neighbor 10.0.2.1	указываем соседа с ip адресом 10.0.2.1

Кроме того, в случае с frame-relay, необходимо так же указывать канальный адрес соседа (если не работает InverseARP). Поскольку схема frame-relay point to multipoint сложна, (там нужно отключать защиту от петель), лучше использовать систему point-to-point с помощью суб интерфейсов, настраивать InversARP и тогда OSPF будет работать почти как в ethernet.

Защита OSPF. Настройка аутентификации:

Router(config)#	int e0/1	переходим на интерфейс ethernet 0/1. Данный интерфейс смотрит на соседний роутер с включенным ospf.
Router(config-if)#	ip ospf authentication message-digest	включаем аутентификацию ospf с алгоритмом хеширования message-digest
Router(config-if)#	ip ospf message-digest-key 1 md5 cisco	задаем ключ аутентификации. В нашем случае Cisco
Router#	show ip ospf int e0/1	посмотреть включена ли аутентификация на интерфейсе

4. Конфигурация RIP

Общая конфигурация:

Router(config)#	router rip	перейти в контекст настройки протокола RIP
Router(config-router)#	network 192.168.0.0	анонсировать сеть 192.168.0.0 (маска сети будет взята с интерфейса)
Router(config-router)#	version 2	включить RIP версии 2
Router(config-router)#	timers basic 10 150 150 200	установить таймеры RIP
Router(config-router)#	passive-interface fa8/0	установить интерфейс fa8/0 как пассивный
Router(config-router)#	no auto-summary	отключить автоматическую суммаризацию
Router(config-router)#	ip summary-address rip 192.168.0.0 255.255.0.0	отдать просуммированную сеть 192.168.0.0/16 на другие роутеры
Router(config-router)#	default-information originate	анонсировать дефолтный маршрут 0.0.0.0/0 на все другие роутеры
Router(config-router)#	redistrubure rip ospf 1 metric 10	вбросить в RIP маршруты из OSPF процесса 1 с метрикой 10
Router(config-router)#	ip rip triggered	функция, которая позволяет RIP отправлять полную информацию о всех маршрутах только один раз и затем отправлять её только при изменениях в сети
Router(config-if)#	no ip split-horizon	отключить split-horizon

Диагностика:

Router#	show ip rip database	посмотреть базу данных RIP

5. Конфигурация BGP

Общая конфигурация:

Router#	router bgp 65000	создать экземпляр bgp с номером автономной системы 65000
Router(config-router)#	neighbor 10.0.2.2 remote-as 65100	добавить BGP соседа из автономной системы 65100
Router(config-router)#	neighbor 10.0.2.2 password cisco	установить пароль (опционально)
Router(config-router)#	neighbor 10.0.2.2 update-source Loopback 0	используется для установления соседства с другим роутером не по физическому интерфейсу, а по loobpack интерфейсу. То есть подключение с loobpack на loobpack
Router(config-router)#	network 192.168.2.0 mask 255.255.255.0	анонсировать сеть в BGP (Внимание! Нужно указывать конкретную подсеть, а не общую, как в IGP протоколах)
Router(config-router)#	timers bgp 10 20	изменить значение таймеров
Router(config-router)#	neighbor 10.0.2.2 next-hop-self	отправлять соседу с ip адресом 10.0.2.2 маршруты, полученные нами от EBGP соседа и указывать в качестве next-hop свой ip адрес
Router(config-router)#	neighbor 10.0.2.2 ebgp-multihop 3	указать, что на пути между двумя EBGP соседями может возникнуть 3 роутера
Router#	clear ip bgp *	сбросить таблицу маршрутизации и начать закачивать маршруты заново (при full view процесс может занять несколько часов)
Router#	clear ip bgp neighbor-id out	начать выгрузку своих маршрутов на соседа (старые маршруты на соседнем роутере будут постепенно заменяться новыми, поэтому соседний роутер будет оставаться в рабочем состоянии). Будет использована технология soft-reconfiguration.
Router#	clear ip bgp neighbor-id in	начать загрузку маршрутов с соседа (старые маршруты будут постепенно заменяться новыми, поэтому роутер будет оставаться в рабочем состоянии). Будет использована технология soft-reconfiguration.

Диагностика:

Router#	show ip bgp	посмотреть, какие "сырые" маршруты пришли по BGP
Router#	show ip route bgp	посмотреть, какие BGP маршруты поступили в таблицу маршрутизации
Router#	show ip bgp summary	показать соседей по BGP
Router#	show ip bgp neighbor 10.0.1.1	показать детальную информацию по BGP соседу

Настройка ограничений анонсирования:

Router(config)#	ip prefix-list ISP permit 0.0.0.0/0	создаем prefix-list и указываем сети, которые мы разрешаем принимать от соседа (отправлять) соседу
Router(config-router)#	neighbor 20.0.2.2 prefix-list ISP in (out)	указываем соседа и применяем prefix-list

Настройка приоритетности провайдера:

Router(config)#	route-map FILTER permit 10	создаем route-map
Router(config-route-map)#	set local-preference 150	устанавливаем local-preference в значение 150 для всех маршрутов, которые мы получим от соседа
Router(config-router)#	neighbor 147.54.76.45 route-map FILTER in	применяем route-map к соседу.

Теперь, все маршруты, полученные от соседа 147.54.76.45 будут иметь local-preference в значении 150, то есть они будут более приоритетны по отношению к обычным маршрутам, у которых значение по умолчанию 100

Настройка приоритетности обратного трафика (через какого провайдера будет возвращаться трафик)

Router(config)#	route-map SET-ASPATH permit 10	создаем route-map SET-ASPATH
Router (config-route-map)#	set as-path prepend 64100 64100 64100 64100	специально ухудшаем путь до нашей автономной системы
Router(config-route-map)#	exit	выходим из конфигурирования route-map
Router(config)#	router bgp 64100	переходим в экземпляр bgp с номером автономной системы 64100
Router(config-router)#	neighbor 217.145.14.2 route-map SET-ASPATH out	вешаем наш route-map на соседа

Теперь сосед 217.145.14.2 получит путь до нашей автономной системы равный 64100 64100 64100 64100 и расскажет об этом другим, добавив еще и свой номер автономной системы. В результате обратный интернет-трафик пойдет через другого соседа, для которого мы не ухудшали as-path.

Установить вес для всех маршрутов, полученных от указанного соседа:

Router(config-router)#	neighbor 147.54.76.45 weight 100	указываем вес для маршрутов, полученных от соседа. Если вес больше, значит маршрут лучше. По умолчанию вес 0

Установить вес для определенных маршрутов, полученных от указанного соседа:

Router(config)#	ip prefix-list WEIGHT permit 55.30.30.0/24	создаем prefix-list, который будет отлавливать маршрут до сети 55.30.30.0 по маске 255.255.255.0
Router(config)#	route-map MAP1 permit 10	создаем route-map с именем MAP1
Router(config-route-map)#	match ip address prefix-list WEIGHT	подключаем prefix-list к route-map
Router(config-route-map)#	set weight 150	устанавливаем вес на маршрут до сети 55.30.30.0/24 равный 150
Router(config)#	route-map MAP1 permit 20	создаем еще одну ветку того же самого route-map
Router(config-route-map)#	set weight 0	устанавливаем вес на все остальные маршруты равным 0
Router(config)#	router bgp 65010	переходим к редактированию протокола BGP
Router(config-router)#	neighbor 50.0.1.1 route-map MAP1 in	применяем созданный route-map к IBGP соседу

Атрибут MED (Multi Exit Discriminator):

Router(config)#	route-map SET-MED permit 10	создаем route-map SET-MED
Router(config-route-map)#	set metric 200	устанавливаем значение MED равное 200
Router(config-route-map)#	exit	выходим из конфигурирования route-map
Router(config)#	router bgp 64100	переходим в экземпляр bgp с номером автономной системы 64100
Router(config-router)#	neighbor 217.145.14.2 route-map SET-MED out	устанавливаем наш route-map на соседа

Внимание! Атрибут MED сравнивается только для маршрутов, которые пришли из одной и той же автономной системы.

Фильтрация маршрутов с помощью AS-Path Access-Lists:

Router(config)#	ip as-path access-list 1 permit ^$	создаем as-path access-list, который будет разрешать только те маршруты, которые зародились в нашей автономной системе. На это указывает регулярное выражение ^$. Символ ^ означает начало строки, $ означает конец строки. Между этими символами ничего нет, что означает пустой атрибут AS PATH.
Router(config)#	router bgp 65010	переходим к конфигурированию bgp за 65010 автономную систему

Router(config-router)#	neighbor 217.145.14.2 filter-list 1 out	применяем As-Path Access-List к соседу
Router#	clear ip bgp * out	загружаем новые маршруты на соседей

*вместо ip as-path access-list 1 permit ^$ можно написать другое регулярное выражение, например ip as-path access-list 1 permit _65030$ В этом случае AS-Path Access-Lists отловит все маршруты, которые зародились в автономной системе 65030 и не важно, через сколько транзитных автономных систем прошел маршрут, прежде чем пришел в нашу автономную систему (отловит все as-path, у которых последняя автономная система будет 65030).

Фильтрация маршрутов с помощью prefix-list's:

Router(config)#	ip prefix-list 1 permit 0.0.0.0/0 ge 8 le 24	разрешаем анонсировать соседу (либо принимать от соседа) все сети, маска которых больше или равна 8, но меньше или равна 24
Router(config)#	router bgp 65010	переходим к конфигурированию bgp за 65010 автономную систему
Router(config-router)#	neighbor 217.145.14.2 prefix-list 1 out	применяем prefix-list к соседу

Фильтрация маршрутов с помощью Route-Map's:

Router(config)#	ip as-path access-list 1 permit ^65020$	создаем as-path access-list, который будет разрешать маршруты, полученные из автономной системы 65020
Router(config)#	ip prefix-list default-only permit 0.0.0.0/0	создаем prefix-list, который будет разрешать получение только дефолтного маршрута
Router(config)#	route-map FILTERING permit 10	создаем route-map с именем FILTERING и веткой 10
Router(config-route-map)#	match ip address prefix-list default-only	подключаем prefix-list к route-map
Router(config-route-map)#	match as-path 10	подключаем as-path access-list к route-map
Router(config-route-map)#	set local-preference 150	если route-map сработает, на маршрут 0.0.0.0/0 будет установлен local-preference в значение 150
Router(config)#	route-map FILTERING permit 20	создаем route-map с именем FILTERING и веткой 20
Router(config-route-map)#	match ip address prefix-list default-only	создаем prefix-list, который будет разрешать получение только дефолтного маршрута. Значение local-preference останется по умолчанию и будет равно 100
Router(config)#	router bgp 65010	переходим к конфигурированию bgp за 65010 автономную систему
Router(config-router)#	neighbor 172.10.10.1 route-map FILTERING in	подключаем route-map к соседу
Router(config-router)#	neighbor 134.15.15.1 route-map FILTERING in	подключаем route-map ко второму соседу

Конфигурация Peer Group:

Router(config)#	router bgp 65010	переходим к конфигурированию bgp за 65010 автономную систему
Router(config-router)#	neighbor ISP peer-group	создаем пир-группу с именем ISP
Router(config-router)#	neighbor ISP filter-list 1 out	подключаем к пир-группе filter-list 1
Router(config-router)#	neighbor ISP prefix-list 25 in	подключаем к пир-группе as-path access-lists 1
Router(config-router)#	neighbor ISP route-map filter out	подключаем к пир-группе route-map с именем FILTER
Router(config-router)#	neighbor 172.10.10.1 remote-as 65020	указываем соседа
Router(config-router)#	neighbor 172.10.10.1 peer-group ISP	подключаем пир-группу ISP к первому соседу
Router(config-router)#	neighbor 134.15.15.1 remote-as 65030	указываем второго соседа
Router(config-router)#	neighbor 134.15.15.1 peer-group ISP	подключаем ту же самую пир-группу ISP ко второму соседу

6. VLANs

6.1 Стандартные VLANs

Switch(config-if)#	switchport mode access	установить порт свитча в access режим (за ним будет только клиент)
Switch(config-if)#	switchport access vlan 2	повесить vlan 2 на порт свитча
Switch(config-if)#	switchport nonegotiate	выключить авто согласование режима работы (транк или аксесс)
Switch(config-if)#	switchport trunk allowed vlan 2,3,4,5,99	разрешить передачу по транк-порту только определенных vlan'ов
Switch(config)#	vlan 2	перейти в режим конфигурирования vlan 2
Switch(config-vlan)#	name sales	задать имя для vlan и назвать его sales
Switch(config-if)#	switchport trunk encapsulation dot1q	установить vlan протокол в dot1q (не сработает, если устройство не поддерживает протокол ISL)
Switch(config-if)#	switchport mode trunk	переключить порт в режим транка
Switch(config-if)#	switchport trunk native vlan 99	изменить native vlan для порта на 99
Switch(config)#	vtp mode transparent	выключить протокол VTP (обмен базой VLANов с соседними свитчами + хранить базу VLANов в конфигурационном файле, а не отдельной flash-памяти). В running config или startup config будет написано: "vlan internal allocation policy ascending"

Switch(config)#	**vtp mode server**	обратная команда
Switch(config)#	**vlan dot1q tag native**	тегировать даже native vlan (для безопасности)

Диагностика:

Switch#	**show vlan**	посмотреть какие есть vlan'ны
Switch#	**show vlan id 2**	посмотреть конкретный vlan
Switch#	**show int fasteth 0/1 switchport**	посмотреть информацию по порту с точки зрения vlan'ов
Switch#	**show int trunk**	посмотреть порты, которые находятся в состоянии транков

6.2 VTP

Общая конфигурация:

Switch(config)#	**vtp mode [transparent|server|client|off]**	установить режим VTP. Transparent – VTP частично выключен (передает объявления от других коммутаторов, сам их не генерирует), server – полнофункциональный режим работы VTP , client – ограниченный режим работы VTP (нельзя создавать, изменять и удалять VLAN из командной строки коммутатора), off – полностью выключен (новый режим работы VTP, который добавился в 3 версии, не передает объявления от других коммутаторов)
Switch(config)#	**vtp version 2 (3)**	выбор версии VTP
Switch(config)#	**vtp domain darkmaycal**	указать имя VTP домена
Switch(config)#	**vtp password 123 [hidden | secret]**	указать пароль VTP домена
Switch(config)#	**no vtp**	отключение VTP на интерфейсе
Switch(config)#	**vtp primary-server**	обозначаем свитч как главный сервер VTP
Switch#	**show vtp status**	показать статус VTP
Switch#	**show vtp password**	показать пароль домена VTP
Switch#	**show vtp devices [conflict]**	показать устройства, входящие в домен VTP (только для v3)
Switch#	**show vtp interface e0/1**	посмотреть включен ли VTP на интерфейсе (только для v3)

6.3 Настройка виртуальных интерфейсов SVI

Конфигурация производится на свитче.

Switch(config)#	ip routing	включаем движок маршрутизации на свитче
Switch(config)#	int vlan 2	создаем виртуальный SVI интерфейс vlan 2
Switch(config-if)#	ip address 192.168.2.50 255.255.255.0	назначаем ip на виртуальный SVI интерфейс
Switch(config-if)#	no shut	включаем виртуальный интерфейс
Switch(config)#	int vlan 3	создаем виртуальный SVI интерфейс vlan 3
Switch(config-if)#	ip address 192.168.3.50 255.255.255.0	назначаем ip на виртуальный SVI интерфейс
Switch(config-if)#	no shut	включаем виртуальный интерфейс

7. Access-Lists (ACL)

Заметка: когда мы устанавливаем access-list на OUT то он срабатывает только тогда, когда пакет ПРИХОДИТ на роутер из вне (например с другого интерфейса). А если мы устанавливаем на IN, то лист срабатывает сразу же как на него пришел пакет (не откуда-то из вне, а от подключенного к нему проводом клиента и отправляет во вне)

Рекомендации - стандартный access-list лучше вешать ближе к получателю, расширенный - ближе к отправителю

Стандартный ACL:

Задача: запретить одному хосту доступ в интернет, а все другим разрешить

Router(config)#	access-list 1 deny 192.168.10.5	запретить хост 192.168.10.5
Router(config)#	access-list 1 permit 192.168.10.0 0.0.0.255	разрешить остальную подсеть

далее зайти на интерфейс и повесить этот access-list на интерфейс:

Router(config-if)#	ip access-group 1	вешаем созданный нами access-list на интерфейс

Расширенный ACL:

Задача: запретить одному хосту доступ по 80 протоколу, а все остальным - разрешить

Router(config)#	access-list 110 deny tcp host 192.168.10.1 any eq 80	запретить хосту 192.168.10.1 доступ к любому хосту по протоколу 80
Router(config)#	access-list 110 permit ip 192.168.10.0 0.0.0.255 any	разрешить остальную подсеть
Router(config)#	interface fa0/1	заходим на интерфейс fast ethernet 0/1
Router(config-if)#	ip access-group 110 out	И применяем наш ACL на выход интерфейса fa0/1

Задача 2: запретить доступ из VLAN 2 в VLAN 3 с помощью ACL.

На роутере, который выполняет маршрутизацию делаем следующее:

Switch(config)#	access-list 101 deny ip 192.168.10.0 0.0.0.255 192.168.20.0 0.0.0.255	запрещаем доступ из 192.168.10.0 подсети в 192.168.20.0 по любым протоколам и любым портам
Switch(config)#	access-list 101 permit ip any any	разрешаем всё остальное
Switch(config-if)#	ip access-group 101 out	применяем этот access лист на выходной интерфейс подсети 192.168.20.0

Либо:

Switch(config-if)#	ip access-group 101 in	на вход интерфейса с подсетью 192.168.10.0

Либо (если VLANs обслуживаются виртуальные SVI интерфейсы свитча):

Switch(config)#	int vlan 2	заходим на интерфейс vlan 2
Switch(config-if)#	ip access-group 101 in	применяем access-list на вход

Именованный ACL (стандартный или расширенный):

Router(config)#	ip access-list extended (standard) MY_LIST	создаем расширенный или стандартный ACL с именем MY_LIST и переходим к его редактированию
Router(config-ext-nacl)#	permit ip host 192.168.0.2 host 172.20.20.2	разрешаем доступ от хоста 192.168.0.2 на хост 172.20.20.2 по любому протоколу
Router(config)#	ip access-list resequence MY_LIST 10 20	перенумеровать все строки ACL с именем MY_LIST. 10 – первый номер, 20 – шаг нумерации

VACL:

Switch(config)#	mac access-list extended MY_MAC_LIST	создаем mac ACL (если необходимо)
Switch(config-ext-macl)#	permit host 0000.3131.0110 any	разрешаем хосту с MAC адресом 0000.3131.0110 подключаться на любой хост
Switch(config-ext-macl)#	exit	выходим из настройки MY_MAC_LIST
Switch(config)#	ip access-list extended MY_IP_LIST	создаем ip ACL (если необходимо)
Switch(config-ext-nacl)#	permit ip host 192.168.0.1 any	разрешаем хосту с ip адресом 192.168.0.1 подключаться на любой хост
Switch(config-ext-nacl)#	exit	выходим из настройки MY_IP_LIST
Switch(config)#	vlan access-map MY_VLAN_MAP	создаем vlan access-map с именем MY_VLAN_MAP
Switch(config-access-map)#	match mac address MY_MAC_LIST	подключаем созданный ранее MAC ACL
Switch(config-access-map)#	match ip address MY_IP_LIST	подключаем созданный ранее IP ACL
Switch(config-access-map)#	action forward	разрешаем прохождение трафика, если он попадает по access-list
Switch(config-access-map)#	exit	выходим из режима конфигурирования MY_VLAN_MAP
Switch(config)#	vlan filter MY_VLAN_MAP vlan-list 150-170	указываем VLANы, к которым будет применяться MY_VLAN_MAP

PACLs

На интерфейсах, к которым подключаются сервера со статическими IP (DHCP, основной шлюз) можно повесить Port ACL:

Switch(config)#	**ip access-list standard SERVER1**	создаем стандартный access-list с именем SERVER1
Switch(config-std-nacl)#	**permit 192.168.1.1**	разрешаем ip адрес 192.168.1.1
Switch(config-std-nacl)#	**deny any log**	все остальное запрещаем. Благодаря параметру log, в консоль будут генерироваться сообщения в случае срабатывания этого deny-правила
Switch(config)#	**int fa0/3**	заходим на интерфейс fast ethernet 0/3 (который, к примеру, смотрит на DHCP сервер)
Switch(config-if)#	**ip access-group SERVER1 in**	применяем access-list на интерфейс

Диагностика:

Switch#	**show access-lists**	показать все access листы
Switch#	**show ip interface (интерфейс) | include access-lists**	посмотреть весит ли на интерфейсе access-list
Switch#	**show run | include access-list**	Показать секцию access-list из running-config

8. Защита от петель. Spanning-Tree Protocol (STP)

Общая конфигурация:

Switch(config)#	spanning-tree vlan 1 root primary (secondary)	стать рутом (запасным рутом) за первый VLAN
Switch(config)#	spanning-tree vlan 1 prioroty 110	установить приоритет роутера в иерархии spanning-tree в 110 за первый VLAN
Switch(config)#	spanning-tree vlan 1 forward-time 12	установить время "схождения сети" за первый VLAN
Switch(config)#	spanning-tree pathcost method (long | short)	при значении «long» будет включен стандарт 802.1t с поддержкой поля «path cost» в BPDUs кадре в 32 бита. Стоимость интерфейсов будет рассчитываться по формуле (20 000 000 000) / (скорость интерфейса Kb/s)
Switch(config-if)#	spanning-tree vlan 1 cost 5	изменить стоимость интерфейса
Switch(config-if)#	spanning-tree portfast	включение функции portfast на интерфейсе
Switch(config)#	spanning-tree portfast default	включить portfast на всех интерфейсах (нужно будет вручную выключить на аплинках)
Switch(config-if)#	spanning-tree bpduguard enable	отключение порта, если он получит BPDU от другого свитча
Switch(config)#	spanning-tree portfast bpdufilter default	глобально включить bpdufilter на всех portfast портах
Switch(config-if)#	spanning-tree bpdufilter enable	выключить приём и передачу BPDU на интерфейсе
Switch(config-if)#	spanning-tree vlan 1 port-priority 50	установить приоритет порта в 50
Switch(config-if)#	spanning-tree guard loop или spanning-tree loopguard default	включить loopguard на интерфейсе (или глобально)
Switch(config)#	spanning-tree backbonefast	включить функцию backbonefast (для PVST+)
Switch(config)#	spanning-tree uplinkfast	включить функцию uplinkfast (для PVST+)
Switch(config-if)#	spanning-tree guard root	включить защиту от другого рута на интерфейсе
Switch(config-if)#	spanning-tree link-type point-to-point	установить среду передачи данных p2p (для rapid pvst)
Switch(config)#	spanning-tree mode rapid-pvst	переключить spanning tree на rapid-pvst
Switch#	debug spanning-tree events	включить протоколирование событий spanning-tree
Switch(config)#	udld (enable | aggressive)	включить UDLD глобально (включается только на оптических интерфейсах)
Switch(config-if)#	udld port (enable | aggressive)	принудительно включить UDLD на медном интерфейсе
Switcth#	udld reset	восстановить интерфейсы, которые были заблокированы udld

Диагностика:

Switch#	show spanning-tree summary	показать все включенные функции в spanning-tree (bpduguard, loopguard, uplinkfast и т.д)
Switch#	show spanning-tree (vlan 1)	показать информацию по spanning tree (за первый VLAN)
Switch#	show spanning-tree int fa0/1 portfast	показать, включен ли на интерфейсе fast ethernet 0/1 режим portfast
Switch#	show udld	показать состояние UDLD
Switch#	debug spanning-tree events	включить вывод отладочной информации

9. Отказоустойчивость шлюза. FHRP, VRRP, GLBP

Протоколы класса FHRP поддерживаются как на маршрутизаторах, так и на L3 коммутаторах.

Конфигурация HSRP. Конфигурация на роутер 1:

Router(config)#	int fa0/1	переходим на интерфейс fast ethernet 0/1 (этот интерфейс смотрит в локальную сеть на коммутаторы)
Router(config-if)#	ip address 192.168.1.2 255.255.255.0	задаем ip адрес для физического интерфейса
Router(config-if)#	standby 1 ip(ipv6) 192.168.1.1	задаем виртуальный ip адрес (который будет основным шлюзом для свитчей, смотрящих на конфигурируемый роутер). У обоих роутеров он одинаковый
Router(config-if)#	stanby 1 priority 110	устанавливаем приоритет данного роутера в 110 (по умолчанию приоритет 100)
Router(config-if)#	standby 1 preempt	задаем режим приемтинга
Router(config-if)#	standby 1 authentication md5 key-string MyPassword	задаем аутентификацию, если необходимо. Пароль будет передаваться с защитой алгоритмом хеширования md5, пароль будет MyPassword (опционально)
Router(config-if)#	standby 1 timers 200 750	регулировка таймеров hsrp, где 200 – hellow интервал в секундах (как часто посылаются пакеты hellow пакеты keep-alive) и 750 – hold interval в секундах (через какой промежуток времени признавать соседа недоступным) (настройка таймеров опциональна)
Router(config-if)#	standby 1 preempt delay minimum 300	настройка времени задержки (в секундах), через которое роутер будет становиться главным

Конфигурация на роутер 2:

Router(config)#	int fa0/1	переходим на интерфейс fast ethernet 0/1 (этот интерфейс смотрит в локальную сеть на свитчи)
Router(config-if)#	ip address 192.168.1.3 255.255.255.0	задаем ip адрес для физического интерфейса
Router(config-if)#	standby 1 ip 192.168.1.1	задаем виртуальный ip адрес (который будет основным шлюзом для свитчей, смотрящих на конфигурируемый роутер). У обоих роутеров он одинаковый
Router(config-if)#	standby 1 preempt	задаем режим приемтинга
Router(config)#	track 1 interface fa0/1 line-protocol	отслеживаем состояние интерфейса fa0/1, если он падает, то сработает объект отслеживания track 1
Router(config-if)#	standby 1 track 1 decrement 20	если сработает объект отслеживания track 1, то текущий приоритет будет понижен на 20 едениц.
Router(config-if)#	standby 1 track 1 fa0/1 20	работает только в HSRP. Позволяет отслеживать интерфейс без дополнительного создания объекта отслеживания
Router(config-if)#	standby 1 authentication md5 key-string MyPassword	задаем аутентификацию, если необходимо. Пароль будет передаваться с защитой алгоритмом хеширования md5, пароль будет MyPassword (опционально)
Router(config-if)#	standby 1 timers 200 750	регулировка таймеров hsrp, где 200 – hellow интервал в секундах (как часто посылаются пакеты hellow пакеты keep-alive) и 750 – hold interval в секундах (через какой промежуток времени признавать соседа недоступным) (настройка таймеров опциональна)
Router(config-if)#	standby 1 preempt delay minimum 300	настройка времени задержки (в секундах), через которое роутер будет становиться главным

Конфигурация VRRP:

Кроме HSRP можно использовать протокол VVRP. Включается точно так же, только слово standby меняется на vrrp. Команды по насройке приоритета, приемтига, аутентификации и таймеров аналогичны HSRP. Внимание! Приемптинг у VRRP по умолчанию включен, а у HSRP выключен.

Router(config-if)#	vrrp 1 ip 192.168.1.1	включение vrrp

Конфигурация GLBP:

Router(config-if)#	**glbp 1 ip 192.168.1.1**	включение glbp
Router(config-if)#	**glbp 1 priority 110**	установить приоритет для AVG в 110
Router(config-if)#	**glbp 1 preempt**	установить режим приемптинга для AVG
Router(config-if)#	**glbp 1 weighting 130**	установить вес для AVF в 130
Router(config-if)#	**glbp 1 weighting 130 lower 20 upper 50**	установить вес для AVF в 130, при этом нижний предел будет 20, верхний 50. Это означает, что AVF перестанет быть Forwarder, если её вес упадет до 19, и снова станет Forwarder, только если вес станет больше 50. По умолчанию lower равен 1, upper равен 100.
Router(config)#	**track 1 interface fa0/1 line-protocol**	отслеживаем состояние интерфейса fa0/1
Router(config)#	**track 2 interface fa0/2 line-protocol**	отслеживаем состояние интерфейса fa0/2
Router(config-if)#	**glbp 1 weighting track 1 decrement 50**	если упадет интерфейс fa0/1, то вес понизится на 50
Router(config-if)#	**glbp 1 weighting track 2 decrement 20**	если упадет интерфейс fa0/2, то вес понизится на 20
Router(config-if)#	**glbp 1 load-balancing host-depended | round-robin | weighted**	установить режим распределения нагрузки между AVF

HSRP – проприетарный протокол, поддерживается только в оборудовании Cisco

GLBP – так же проприетарный протокол, поддерживается только в оборудовании Cisco

VRRP – протокол открытого стандарта, поддерживается другими вендорами

Диагностика:

Router#	**show standby (vrrp or glbp)**	показать общую информацию по протоколу группы FHRP
Router#	**show standby brief**	показать информацию по протоколу группы FHRP в виде таблицы

Конфигурация L2 EtherChannel:

Switch(config)#	int range fa0/1-2	заходим в режим конфигурирования группы интерфейсов от fast ethernet 0/1 до fast ethernet 0/2
witch(config-if-range)#	switchport mode trunk	переключаем группу интерфейсов в режим транка
Switch(config-if-range)#	switchport nonegotiate	отключаем авто согласование режима работы (отключаем протокол DTP)
Switch(config-if-range)#	switchport trunk allowed vlan 1,2,...	указываем разрешенные VLAN на транках
Switch(config-if-range)#	channel-group 1 mode active	включаем etherchannel с проверкой правильности сборки LACP (mode auto - проверка PAgP, mode on - без проверки)
Switch(config-if-range)#	exit	выходим из режима конфигурирования группы интерфейсов
Switch(config)#	port-channel load-balance dst-ip	включить балансировку трафика по ip назначения

Конфигурация L3 EtherChannel на коммутаторе:

Switch(config)#	int port-channel 1	вручную создаем интерфейс port-channel 1
Switch(config-if)#	no switchport	переводим интерфейс в роутерный режим
Switch(config-if)#	ip address 10.0.1.1	задаем ip адрес
Switch(config)#	int range fa0/1, fa0/2	переходим к конфигурированию группы портов
Switch(config-if-range)#	no switchport	переводим группу портов в роутерный режим
Switch(config-if-range)#	channel-group 1 mode active	включаем EtherChannel с протоколом проверки правильности сборки LACP

Особенности конфигурации L3 EtherChannel на роутерах:

- Поддерживается только статическое агрегирование, без использования протоколов;
- Можно создать только 2 агрегированных интерфейса;
- Максимальное количество интерфейсов в EtherChannel – 4;
- Метод балансировки использует IP-адреса отправителя и получателя, включен по умолчанию и не может быть изменен;
- Агрегировать можно только те интерфейсы, которые находятся на модулях одинакового типа.

Диагностика:

Switch#	**show int port-channel 1**	показать состояние виртуального интерфейса (не работает в PacketTracer)
Switch#	**show etherchannel summary**	показать общение состояние etherchannel
Switch#	**show etherchannel port-channel**	более детальная информация
Switch#	**show etherchannel load-balance**	посмотреть в какой логике работает балансировка EtherChannel

Cisco Catalyst 45 и 65 серии умеют балансировать по вложениям TCP и UDP

11. Отказоустойчивость. FlexLinks

Общая конфигурация:

Switch(config)	**int fa0/1**	зайти на интерфейс fast ethernet 0/1 (для которого в последствии будет указан резервный интерфейс)
Switch(config-if)#	**switchport backup interface fa0/2**	установить интерфейс fa0/2 в качестве запасного (на который будет переключаться поток данных если основной интерфейс упадет)
Switch#	**show interface switchport backup**	показать порты, которые являются запасными

12. Сетевая трансляция адресов. NAT

Static NAT:

Router(config)#	**int fa0/0**	заходим на интерфейс fast ethernet 0/0
Router(config-if)#	**ip nat inside**	устанавливаем интерфейс fa0/0 как внутренний (локальная сеть)
Router(config)#	**int fa0/1**	заходим на интерфейс fast ethernet 0/1
Router(config-if)#	**ip nat outside**	устанавливаем интерфейс fa0/0 как внешний (интернет)
Router(config)#	**ip nat inside source static 192.168.1.2 215.215.215.20**	включаем статический NAT. IP адрес 192.168.1.2 будет транслироваться на 215.215.215.20

Dynamic NAT:

Router(config)#	int fa0/0	заходим на интерфейс fast ethernet 0/0
Router(config-if)#	ip nat inside	устанавливаем интерфейс fa0/0 как внутренний (локальная сеть)
Router(config)#	int fa0/1	заходим на интерфейс fast ethernet 0/1
Router(config-if)#	ip nat outside	устанавливаем интерфейс fa0/0 как внешний (интернет)
Router(config)#	access-list 1 permit 192.168.1.0 0.0.0.255	создаем стандартный нумерованный ACL, который будет указывать диапазон частных ip адресов, которым будет разрешено транслироваться на внешние ip адреса. (Внимание! Те, которые в лист не попали - транслироваться не будут!)
Router(config)#	ip nat pool TRANS 215.215.215.20 215.215.215.30 netmask 255.255.255.0	создаем пул ip адресов с именем TRANS. Это ip адреса на которые будут транслироваться частные ip адреса, указанные в access-list 1. В данном случае частные ip адреса будут транслироваться на внешние ip адреса начиная с 215.215.215.20 и заканчивая 215.215.215.30
Router(config)#	ip nat inside source list 1 pool TRANS	включаем динамический NAT, где list 1 – диапазон частных ip адресов, pool TRANS – диапазон публичных ip адресов

NAT с перегрузкой (PAT):

Router(config)#	int fa0/0	заходим на интерфейс fast ethernet 0/0
Router(config-if)#	ip nat inside	устанавливаем интерфейс fa0/0 как внутренний (локальная сеть)
Router(config)#	int fa0/1	заходим на интерфейс fast ethernet 0/1
Router(config-if)#	ip nat outside	устанавливаем интерфейс fa0/0 как внешний (интернет)
Router(config)#	ip nat pool OVRLD 172.16.10.1 172.16.10.1 netmask 255.255.255.0	указываем на какой ip будем транслировать внутренние ip адреса локальной сети (здесь он только один, поэтому повторяется два раза)
Router(config)#	access-list 7 permit 192.168.1.0 0.0.0.255	указываем пул внутренних ip адресов, которые будем транслировать (Внимание! Те, которые в лист не попали - транслироваться не будут!)
Router(config)#	ip nat inside source list 7 pool OVRLD overload	включаем PAT, где list 7 – диапазон частных ip адресов, pool OVRLD – диапазон публичных ip адресов
Router(config)#	ip nat inside source static 192.168.1.5 interface fa0/0	проброс ВСЕХ портов на ip 192.168.1.5. fa0/0 – интерфейс, направленный в интернет (аналог DMZ в простых роутерах)

Router(config)#	ip nat inside source static tcp 192.168.1.3 80 172.20.20.15 80	проброс веб порта, где 192.168.1.3 - адрес компьютера в локальной сети и 172.20.20.15 белый ip адрес, то есть тот адрес, на который NAT транслирует наши внутренние ip адреса
Router(config)#	ip nat inside source static tcp 192.168.1.3 80 interface Ethernet0/0 80)	проброс веб порта с указанием интерфейса смотрящего в интернет
Router#	clear ip nat translation *	сброс таблицы динамической трансляции (настройки не сбрасываются)

Внимание! Команда *«ip nat inside source list 7 pool ovrld overload»* в таблице представленной выше, перебивает внутренние (частные) ip адреса на любой адрес, который мы указываем в ip nat pool ovrld. Дело в том, что этот ip адрес (на который мы перебиваем) не будет привязан к ip адресу, который весит на интерфейсе смотрящего на провайдера. Это может создать множество проблем, поэтому есть альтернативная команда:

Router(config)#	ip nat inside source list 7 interface e0/1 overload	эта команда будет перебивать наши частные ip адреса не на выдуманный нами ip адрес, а на ip адрес, который весит на интерфейсе, который смотрит на провайдера!

NVI NAT (на примере PAT):

Router(config)#	int fa0/0	заходим на интерфейс fast ethernet 0/0
Router(config-if)#	ip nat enable	включаем NAT на интерфейсе
Router(config)#	int fa0/1	заходим на интерфейс fast ethernet 0/1 (смотрящий в интернет)
Router(config-if)#	ip nat enable	включаем NAT на интерфейсе
Router(config)#	access-list 7 permit 192.168.1.0 0.0.0.255	указываем пул внутренних ip адресов, которые будем перебивать (Внимание! Те, которые в лист не попали - транслироваться не будут!)
Router(config)#	ip nat source list 7 interface fa0/1 overload	включаем PAT, где list 7 – диапазон частных ip адресов, fa0/1 – интерфейс, на ip адрес которого будут транслироваться частные ip адреса. Внимание! Параметр «inside» не используется.

Диагностика:

Router#	show ip nat translation	показать текущую трансляцию
Router#	show ip nat statistics	показать количество срабатываний (счетчики) NAT
Router#	show ip nat nvi translation	показать текущую трансляцию (для NVI NAT)
Router#	show ip nat nvi statistics	показать количество срабатываний (счетчики) NAT (для NVI NAT)
Router#	debug ip nat	включить вывод отладочной информации

13. Настройка DHCP

Общая настройка DHCP сервера:

Router(config)#	ip dhcp excluded-address 192.168.10.50	исключаем из выдачи DHCP этот ip адрес (или диапазон)
Router(config)#	ip dhcp pool VLAN2POOL	создаем пул DHCP и присваиваем ему имя VLAN2POOL
Router(dhcp-config)#	network 192.168.2.0 255.255.255.0	указываем какую сеть нужно раздавать по DHCP в этом пуле
Router(dhcp-config)#	default-router 192.168.2.50	какой шлюз по умолчанию будем раздавать
Router(dhcp-config)#	dns-server 217.217.217.2	какой ip адрес DNS сервера будем раздавать
Router(dhcp-config)#	bootfile FILENAME	задать имя загрузочного образа
Router(dhcp-config)#	option 33 ip 156.42.45.0 192.168.1.1	задать опцию 33 (клиенту будет отсылаться статический маршрут)
Router(dhcp-config)#	netbios-name-server 192.168.1.2	указать адрес WINS сервера
Router(dhcp-config)#	lease 2	устанавливаем время аренды адреса на 2 дня

Настройка интерфейса роутера для работы DHCP:

Router(config)#	interface fa0/0.2	добавление виртуального интерфейса (суб интерфейса)
Router(config-if)#	encapsulation dot1Q 2	заставляем виртуальный интерфейс работать с VLAN2
Router(config-if)#	ip address 192.168.2.2	вешаем ip на виртуальный интерфейс

В результате по этому интерфейсу клиенты из VLAN2 будут получать ip адреса из VLAN2POOL

Жесткая привязка определенного ip адреса к MAC адресу (чтобы клиенту всегда выдавался только определенный ip):

Router(config)#	**ip dhcp pool CLIENT**	под одного клиента заводим целый пул
Router(dhcp-config)#	**host 192.168.50.5 255.255.255.255**	указываем ip адрес который будем ему выдавать
Router(dhcp-config)#	**client-identifier 0001.976B.291D**	указываем MAC адрес клиента, к которому будем привязывать IP адрес

Внимание! Для компьтеров, работающих по управлениям Microsoft, идентификатором является число 01 перед MAC-адресом. Для UNIX компьютеров необходимо проставлять 00.

То есть, для компьютера с MAC-адресом 00.04.76.10.6c.bc, который работает в среде Windows строчка client-identifier будет выглядеть как:

```
Router(dhcp-config)#client-identifier 0100.0476.106c.bc
```

Для компьютера с MAC-адресом 00.04.76.10.6c.bc который работает в среде UNIX строчка client-identifier будет выглядеть как:

```
Router(dhcp-config)#client-identifier 0000.0476.106c.bc
```

Дополнительная настройка DHCP сервера:

Router(config-if)#	**ip helper-address 10.0.1.4**	настройка агента DHCP (DHCP Relay). IP 10.0.1.4 – адрес DHCP сервера, на который агент будет перенаправлять DHCP сообщения от конечных узлов
Router(config)#	**ip dhcp ping packets <0-10>**	изменить количество отправляемых ICMP запросов (0 отключает ping)
Router(config)#	**ip dhcp ping timeout 200**	изменение таймаута между запросами
Router#	**clear ip dhcp binding**	очистка таблицы соответствия физических адресов и адресов выданных с пула DHCP-сервером
Router#	**clear ip dhcp binding 192.168.2.4**	очистить привязку для конкретного IP-адреса
Router#	**clear ip dhcp conflict ***	очистить базу конфкликтный ip адресов
Router(config)#	**ip dhcp database ftp://user:password@192.168.1.5/router-dhcp timeout 80**	изменить место хранение базы данных IP адресов

Диагностика:

Router#	**show ip dhcp pool**	показать оставшиеся и использованные DHCP адреса на роутере
Router#	**show ip dhcp binding**	показать какие ip адреса были использованы
Router#	**show ip dhcp binding 192.168.10.1**	показать детальную информацию по ip адресу 192.168.10.1 (с указанием MAC адреса)
Router#	**show ip dhcp conflict**	просмотр информации о конфликтах, при назначении IP-адресов
Router#	**show ip dhcp database**	посмотреть информацию о состоянии базы данных DHCP
Router#	**show ip dhcp server statistics**	просмотр статистики DHCP сервера
Router#	**debug ip dhcp server packet**	включение дебага (вывод информации для отладки)

14. Протоколы канального уровны (PPP, HDLC, Frame-Relay)

14.1 Соединение двух устройств по L2 протоколу PPP

Сначала роутер 1 назовем Router1, а роутер 2 - Router2 (команда hostname).

Затем:

Router1(config)#	**username Router2 password cisco123**	создаем учетную запись для того, чтобы мы могли подключиться к Router2
Router1(config)#	**int ser9/0**	переходим на интерфейс serial 9/0
Router1(config-if)#	**ppp authentication chap**	запрещаем подключение к нам (к роутеру 1) если роутер 2 не предоставит аутентификационные данные. У роутера 2 эту запись включать не обязательно. Однако данная конфигурация рассчитана на то, что роутер 2 будет запрашивать аунтентификацию.
Router2(config)#	**username Router1 password cisco123**	создаем учетную запись для того, чтобы Router2 мог подключиться к нам
Router2(config)#	**int ser8/0**	переходим на интерфейс serial 8/0
Router2(config-if)#	**ppp authentication chap**	запрещаем подключение к роутеру 2 если роутер 1 не предоставит аутентификационные данные
Router(config)#	**debug ppp authentication**	выполняем диагностику, если что-либо не работает

Теперь можно выполнять передачу данных. Канальных адресов нет, так как это Point-to-Point среда.

14.2 Соединение несольких устройств по L2 протоколу Frame-Relay

Сначала роутер 1 назовем Router1, а роутер 2 назовем Router2 (команда hostname).

Затем:

Router1(config)#	int ser9/0	переходим на интерфейс serial 9/0
Router1(config-if)#	encapsulation frame-relay	настраиваем интерфейс на работу по L2 протоколу frame-relay
Router1(config-if)#	ip address 192.168.0.1 255.255.255.0	назначаем IP адрес на интерфейс
Router2(config)#	int ser8/0	переходим на интерфейс serial 8/0
Router2(config-if)#	encapsulation frame-relay	настраиваем интерфейс на работу по L2 протоколу frame-relay
Router2(config-if)#	ip address 192.168.0.2 255.255.255.0	назначаем IP адрес на интерфейс
Router(config-if)#	frame-relay map ip 10.1.1.1 110 broadcast	вручную задать соответствие ip и DLCI

Теперь можно выполнять передачу данных.

Конфигурация Multipoint Frame Relay (на примере hub роутера):

Router(config)#	interface serial 0/0	переходим к конфигурированию физического интерфейса serial 0/0
Router(config-if)#	no ip address	снимаем ip адрес (если ранее он был сконфигурирован)
Router(config-if)#	encapsulation frame-relay	настраиваем интерфейс на работу с протоколом канального уровня frame-relay
Router(config-if)#	interface serial 0/0.1 multipoint	создаем виртуальный суб-интерфейс serial 0/0.1 и помечаем его как multipoint
Router(config-subif)#	ip address 10.0.1.1 255.255.255.0	задаем ip адрес на виртуальный суб-интерфейс
Router(config-subif)#	frame-relay map ip 10.0.1.2 100 broadcast	задаем соответствие ip адреса первого канального соседа и DLCI за которой он доступен
Router(config-subif)#	frame-relay map ip 10.0.1.3 200 broadcast	задаем соответствие ip адреса второго канального соседа и DLCI за которой он доступен

Внимание! При конфигурировании Multipoint или Point-to-Point интерфейса, Inverse ARP отключается. Необходимо вручную прописывать сопоставления ip адресов соседей по каналу и DLCI.

На spoke роутерах нет необходимости создавать виртуальный суб-интерфейс и помечать его как multipoint.

Конфигурация Point-to-Point Frame Relay (на примере hub роутера):

Router(config)#	**interface serial 0/0**	переходим к конфигурированию физического интерфейса serial 0/0
Router(config-if)#	**no ip address**	снимаем ip адрес (если ранее он был сконфигурирован)
Router(config-if)#	**encapsulation frame-relay**	настраиваем интерфейс на работу с протоколом канального уровня frame-relay
Router(config-if)#	**interface serial 0/0.100 point-to-point**	создаем виртуальный суб-интерфейс serial 0/0.100 и помечаем его как point-to-point
Router(config-subif)#	**ip address 10.0.1.1 255.255.255.0**	задаем ip адрес на виртуальный суб-интерфейс
Router(config-subif)#	**frame-relay interface-dlci 100**	данная команда указывает, что если внутри роутера будет сформирован кадр у которого в поле «назначение» будет указана DLCI 100, то такой кадр будет обрабатывать суб-инетрфейс serial 0/0.100
Router(config-if)#	**interface serial 0/0.110 point-to-point**	создаем виртуальный суб-интерфейс serial 0/0.110 и помечаем его как point-to-point
Router(config-subif)#	**ip address 10.0.2.1 255.255.255.0**	задаем ip адрес на виртуальный суб-интерфейс
Router(config-subif)#	**frame-relay interface-dlci 110**	данная команда указывает, что если внутри роутера будет сформирован кадр у которого в поле «назначение» будет указана DLCI 110, то такой кадр будет обрабатывать суб-инетрфейс serial 0/0.110

Внимание! На spoke роутерах необходимо создать суб-интерфейс и пометить его как Point-to-Point.

Диагностика:

Router#	**show frame-relay pvc**	показать pvc frame-relay
Router#	**show frame-relay map**	показать соответствие ip и DLCI

15. Подключение к провайдеру

15.1 Подключение к двум провайдерам по схеме Multihomed с одним CE роутером

15.1.1 Мониторинг доступности ресурса. IP SLA

Конфигурация IP SLA на примере настройки подключения к двум провайдерам по схеме Multihomed без BGP с одним роутером:

Способ 1:

Router(config)#	ip sla 1	создаем зонд
Router(config-ip-sla)#	icmp-echo 20.0.1.2 source-interface e0/2	посылаем icmp echo ping на 20.0.1.2
Router(config-ip-sla-echo)#	frequency 10	посылаем icmp echo ping с частотой каждые 10 секунд
Router(config)#	ip sla schedule 1 start-time now life forever	задаем расписание работы ip sla. В данном случае зон будет запущен прямо сейчас, при этом время окончания не задано (навсегда)
Router(config)#	track 1 ip sla 1 reachability	устанавливаем объект отслеживания на доступность того хоста, на который посылаем icmp echo ping
Router(config)#	ip route 0.0.0.0 0.0.0.0 2.2.2.2 track 1	направляем трафик по этому маршруту если объект трекинга track 1 работает (хост пингуется)
Router(config)#	ip route 0.0.0.0 0.0.0.0 3.3.3.3 10	если не пингуется, направляем трафик в интернет по другому маршруту (Внимание! Здесь важно задать именно плохую метрику, например 10, иначе будут работать оба маршрута! (балансировка))
Router#	show track 1	показать состояние объекта отслеживания

Способ 2 (если первый способ не поддерживается):

Router(config)#	ip sla monitor 1	создаем монитор
Router(config-sla-monitor)#	type echo protocol IpIcmpEcho 20.0.1.2 source-interface e0/2	посылаем icmp echo ping на 20.0.1.2
Router(config-sla-monitor-echo)#	frequency 10	посылаем icmp echo ping с частотой каждые 10 секунд

Router(config)#	ip sla monitor schedule 1 life forever start-time now	задаем расписание работы ip sla. В данном случае зон будет запущен прямо сейчас, при этом время окончания не задано (навсегда)
Router(config)#	track 1 rtr 1 reachability	устанавливаем объект отслеживания на доступность того хоста, на который посылаем icmp echo ping
Router(config)#	ip route 0.0.0.0 0.0.0.0 2.2.2.2 track 1	направляем трафик по этому маршруту если объект трекинга track 1 работает (хост пингуется)
Router(config)#	ip route 0.0.0.0 0.0.0.0 3.3.3.3 10	если не пингуется, направляем трафик в интернет по другому маршруту (Внимание! Здесь важно задать именно плохую метрику, например 10, иначе будут работать оба маршрута! (балансировка))
Router#	show track 1	показать состояние объекта отслеживания

*Можно выполнять мониторинг не ip соседнего роутера, а какой-нибудь внешний ресурс. Но тогда доступ к этому ресурсу должен быть только по одному из провайдеров:

ip route 85.202.241.71 255.255.255.255 next-hop роутера провайдера 1

Таким образом мы будем ходить на 85.202.241.71 только через провайдера 1, и теперь можно выполнять мониторинг внешнего ресурса через этого одного, конкретного провайдера.

15.1.2 Динамическое изменение правил трансляции NAT в зависимости от активного провайдера

Проблема: каждый из провайдеров разрешает настраивать NAT-трансляцию на ip адрес, который он же и выдает. Наш роутер настроен на NAT-трансляцию только для одного провайдера.

Решение: поскольку у нас 2 провайдера и 1 роутер, нам необходимо заставить роутер автоматически изменять NAT-трансляцию в зависимости от того, какой провайдер активен в настоящий момент.

Для первого провайдера:

Router(config)#	route-map ISP1 permit 10	создаем route-map с именем IPS1
Router(config-route-map)#	match interface e0/1	отслеживаем интерфейс, через который в настоящий момент идет трафик в интернет (в него трафик направляется в зависимости от доступности провайдера благодаря треку отслеживания track 1)
Router(config)#	ip nat pool overld 217.145.14.4 217.145.14.4 netmask 255.255.255.0	если через него идет трафик, то тогда устанавливаем пул для трансляции внутренних ip адресов нашей сети на ip 217.145.14.4 (это ip первого провайдера)
Router(config)#	ip nat inside source route-map ISP1 pool ovrld overload	настраиваем nat таким образом, чтобы он брал параметры pool'а из route-map

Для второго провайдера:

Router(config)#	route-map ISP2 permit 10	создаем route-map с именем IPS1
Router(config-route-map)#	match interface e0/2	отслеживаем интерфейс, через который в настоящий момент идет трафик в интернет (в него трафик направляется в зависимости от доступности провайдера благодаря треку отслеживания track 1)
Router(config)#	ip nat pool overld2 147.54.76.4 147.54.76.4 netmask 255.255.255.0	если через него идет трафик, то тогда устанавливаем пул 2 для трансляции внутренних ip адресов нашей сети на ip 147.54.76.4 (это ip второго провайдера)
Router(config)#	ip nat inside source route-map ISP2 pool ovrld2 overload	настраиваем nat таким образом, чтобы он брал параметры pool'а из route-map

С двумя роутерами делается аналогично. Используется технология HSRP или VRRP. Для схемы с двумя роутерами уже не нужен route-map. Каждый роутер смотрит на своего провайдера и у каждого роутера запущен свой экземпляр NAT. В зависимости от того, какой роутер в FHRP будет активен, тот и будет обрабатывать трафик.

15.2 Особенности Multihomed подключения к двум провайдерам по BGP с использованием двух CE

При Multihomed подключении к двум провайдерам по BGP с использованием двух CE, оба CE анонсируют одну и ту же сеть (PI адрес) обоим ISP. Из-за этого обратный трафик может возвращаться не через тот CE, через которой он вышел. При включенном NAT на обоих CE работа такой сети не возможна, так как обратный трафик может прийти на тот CE роутер, у которого не было задано динамическое правило трансляции. Для решения данной проблемы, необходимо CE роутеру, который направлен на резервный ISP, запретить анонсировать PI адрес пока основной ISP находится в работоспособном состоянии. Для того, чтобы CE, направленный на резервный ISP, мог проверять состояние основного ISP, основной ISP помимо дефолтного маршрута должен присылать дополнительный маршрут. На основании наличия этого маршрута, с помощью route-map, CE роутер направленный на резервного провайдера будет либо анонсировать, либо не анонсировать PI адрес резервному ISP.

Конфигурация:

Router(config-router)#	**ip prefix-list NONEXIST seq 5 permit 1.2.3.0/24**	создаем prefix-list, который будет отлавливать специальный маршрут, который анонсирует основной провайдер
Router(config)#	**ip prefix-list our-network seq 5 permit 147.45.67.34/24**	указываем сеть, которую будет анонсировать резервный роутер резервному провайдеру (ту самую, в которую входит купленный нами PI адрес)
Router(config)#	**route-map NONEXIST_MAP permit 10**	создаем route-map, который будет срабатывать в том случае, если prefix-list NONEXIST будет отлавливать маршрут 1.2.3.0/24
Router(config-route-map)#	**match ip address prefix-list NONEXIST**	
Router(config-route-map)#	**route-map ournets permit 100**	создаем route-map ournets
Router(config-route-map)#	**match ip address prefix-list our-network**	route-map ournets будет срабатывать всегда, он нужен для реализации механизма advertise-map
Router(config)#	**router bgp 65100**	переходим к конфигурированию BGP за AS 65100
Router(config-router)#	**neighbor 132.56.43.21 route-map ournets out**	подключаем к соседу (который является для нас резервным провайдером) router-map с именем **ournets out**
Router(config-router)#	**neighbor 132.56.43.21 advertise-map ournets non-exist-map NONEXIST_MAP**	сеть, написанная в prefix-list our-network, который подключен к route-map ournets, будет анонсироваться только в том случае, если route-map NONEXIST_MAP будет срабатывать, то есть будет приходить специальный маршрут, который анонсирует основной провайдер

15.3 Подключение к провайдеру с использованием PPPoE

1. Создаем и настраиваем интерфейс Dialer:

Router(config)#	**interface Dialer1**	создаем интерфейс dialer1
Router(config-if)#	**ip address negotiated**	ip адрес будет присваиваться интерфейсу автоматически
Router(config-if)#	**ip mtu 1492int fa**	установим размер mtu в 1492 байта (это максимальный размер пакета, передаваемого через PPPoE)
Router(config-if)#	**ip nat outside**	включаем NAT и указываем направление
Router(config-if)#	**encapsulation ppp**	указываем режим инкапсуляции (протокола канального уровня, который будет использован)
Router(config-if)#	**dialer pool 1**	создаем dialer pool (будет использован для привязки к физическому интерфейсу)
Router(config-if)#	**ppp authentication chap callin**	указываем алгоритм проверки подлинности. В данном случае chap
Router(config-if)#	**ppp chap hostname Maycal**	задаем имя пользователя Maycal
Router(config-if)#	**ppp chap password 0 Ghd%4gdns**	задаем пароль Ghd%4gdns
Router(config-if)#	**exit**	выходим из режима конфигурирования dialer1

2. К интерфейсу, который подключен к сети провайдера, подключаем dialer pool, который мы указали при конфигурировании интерфейса Dialer1):

Router(config)#	**interface FastEthernet0/1**	переходим к редактированию интерфейса, направленного на провайдера
Router(config-if)#	**pppoe-client dial-pool-number 1**	указываем ранее созданный dialer pool, тем самым подключаем интерфейс dialer 1 к физическому интерфейсу FastEthernet0/1
Router(config-if)#	**exit**	выходим из режима конфигурирования

3. Задаем маршрут по умолчанию через интерфейс Dialer и настраиваем NAT:

Router(config)#	ip route 0.0.0.0 0.0.0.0 dialer 1	указываем маршрут через интерфейс dialer 1
Router(config)#	access-list 1 permit 192.168.0.0 0.0.255.255	создаем пул частных ip адресов, которые мы будем транслировать на ip интерфейса dialer 1
Router(config)#	ip nat inside source static list 1 interface dialer 1 overload	включаем NAT

16. Технологии защиты коммутируемой сети

16.1 Защита по MAC адресам. Port Security

Общая конфигурация:

Switch(config)#	int fa0/1	заходим на интерфейс fast ethernet 0/1
Switch(config-if)#	switchport mode access	переключаем порт в access режим
Switch(config-if)#	switchport port-security	включаем функцию port-security
Switch(config-if)#	switchport port-security maximum 5 [vlan <vlan-list>]	устанавливаем максимальное количество MAC адресов на интерфейсе в 5. Заданием опционального параметра vlan, при указании максимального количества безопасных MAC-адресов, можно ограничить количество MAC-адресов для [VLAN или <перечня VLAN>]
Switch(config-if)#	switchport port-security mac-address sticky	задаем режим, при котором port-security будет самостоятельно изучать MAC адреса и они останутся при перезагрузки коммутатора. Вместо ключевого слова sticky можно вручную прописать разрешенный MAC адрес. Кроме этого,
Switch(config-if)#	switchport port-security violation shutdown	указываем действие при превышении установленного количества MAC адресов. В данном случае порт будет выключен.
Switch(config-if)#	switchport port-security aging time 2	коммутатор по умолчанию не удаляет MAC адреса из CAM таблицы, на которую ориентируется port-security. Если применить эту команду, то автоматически выученные MAC адреса (не sticky, а имеено динамические) будут автоматически удаляться через 2 минуты.
Switch(config-if)#	switchport port-security aging type [absolute | inactivity]	если была применена предыдущая команда, то эта команда определяет в каком случае будут удаляться выученные MAC адреса. Absolute – MAC адреса будут удаляться в любом случае по истечению aging time. Inactivity - MAC адреса будут

		удаляться только в том случае, если данный MAC адрес не обращался на порт коммутатора
Switch(config)#	mls rate-limit layer2 port-security rate_in_pps [burst_size]	функция полезна при violation режимах protect и restrict. Она будет ограничивать количество кадров в секунду, которые поступают на порт от атакующего
Switch#	clear port-security sticky	очистить таблицу MAC адресов, которые выучил port-security
Switch(config)#	errdisable recovery cause psecure-violation	восстановление всех портов из состояния err-disable

Дополнительная настройка:

Часто встречается ситуация, при которой к порту Access коммутатора сначала подключен IP телефон, а к IP телефону подключен компьютер. В этом случае необходимо прописать максимум не 1, а 2 MAC-адреса. В новых IOS можно явно разнести MAC адрес компьютера и MAC адрес телефона в разные VLAN с точки зрения port security:

Switch(config-if)#	switchport port-security maximum 1 vlan access	указываем максимальное количество MAC адресов в обычном VLAN (в котором находится конечный абонент)
Switch(config-if)#	switchport port-security maximum 1 vlan voice	указываем максимальное количество MAC адресов в голосовом VLAN (в котором находится телефон)
Switch(config-if)#	switchport port-security mac-address mac_телефона vlan voice	указываем MAC адрес ip телефона
Switch(config-if)#	switchport port-security mac-address mac_компьютера vlan access	указываем MAC адрес компьютера

Диагностика:

Switch#	show port-security	посмотреть глобальное состояние port-security
Switch#	show port-security int fa0/1	посмотреть состояние port-security за интерфейс
Switch#	show port-security address	посмотреть MAC адреса, которые защищаются port-security
Switch#	show mls rate-limit	показать статус mls (rate-limit)

16.2 Storm-Control

Общая конфигурация:

Switch(config-if)#	int fa0/1	переходим на интерфейс fastethernet 0/1 на котором необходимо настроить storm-control
Switch(config-if)#	storm-control broadcast level 50 30	устанавливаем ограничение широковещательного L2 трафика в процентах, где 50 – верхний предел (Rising Threshold), 30 -–нижний предел (Falling Threshold)
Switch(config-if)#	storm-control multicast level pps 30k 20k	устанавливаем ограничение мультикаст трафика в пакетах в секунду. 30k это 30000.
Switch(config-if)#	storm-control unicast level bps 30m	устанавливаем ограничение юникаст трафика в битах в секунду. Буква m обозначает мегабиты. То есть в данном примере устанавливается ограничение в 30 мегабит на юникастовый трафик.
Switch(config-if)#	storm-control action <shutdown | trap>	устанавливаем действие при превышении указанных выше лимитов. В данном случае произойдёт отключение интерфейса
Switch#	show storm-control [broadcast | multicast | unicast]	посмотреть статистику strom-control

16.3 DHCP Snooping

Общая конфигурация:

Switch(config)#	ip dhcp snooping	глобальное включение dhcp snooping
Switch(config)#	ip dhcp snooping vlan 1	включение dhcp snooping за первый vlan. Необходимо включать за каждый существующий vlan, кроме того, выполнение первой команды необходимо
Switch(config)#	int fa0/1	переходим на интерфейс fast ethernet 0/1
Switch(config-if)#	ip dhcp snooping trust	устанавливаем этот интерфейс, как интерфейс от которого мы ожидаем получение пакетов от DHCP сервера
Switch(config-if)#	ip dhcp snooping limit rate 10	устанавливаем максимальное количество запросов DHCP адресов в 10 запросов в секунду. То есть от клиента в секунду максимум может быть 10 запросов, иначе это будет расценено как атака

Switch(config)#	ip dhcp snooping binding <mac-address> vlan <vid> <ip-address> interface <interface-id> expiry <seconds>	добавление статической записи в базу данных привязки DHCP
Switch(config)#	no ip dhcp snooping verify mac-address	по умолчанию, после включения DHCP snooping, на коммутаторе включена проверка соответствия MAC-адресов. Коммутатор проверяет соответствие MAC-адреса в DHCP-запросе MAC-адресу клиента. Если они не соответствуют, то коммутатор отбрасывает пакет. При необходимости можно отключить эту проверку
Switch(config-if)#	ip dhcp relay information trusted	таким образом указывается доверенный DHCP сервер, который находится вне канальной среды. Команда указывается на виртуальном SVI интерфейсе свитча
Switch(config)#	ip dhcp relay information trust-all	аналог предыдущей команды, однако делает DHCP сервер доверенным на всех SVI

Диагностика:

Switch#	show ip dhcp snooping	просмотр настроек dhcp snooping
Switch#	show ip dhcp snooping statistics	просмотр счетчиков dhcp snooping
Switch#	show ip dhcp snooping binding	просмотр базы данных привязки DHCP
Switch#	show ip dhcp snooping database	показать информацию по базе данных DHCP snooping

16.4 IP Source Guard

Общая конфигурация:

Switch(config)#	int fa0/1	переходим на интерфейс fast ethernet 0/1
Switch(config-if)#	ip verify source vlan dhcp-snooping	включаем функцию IP Source Guard
Switch(config)#	ip source binding 00:E0:F7:EC:D0:10 vlan 1 192.168.1.1 interface fa0/4	за интерфейсом fa0/4 у нас сидит сервер (либо роутер с основным шлюзом), ip которого настроен статически. Это означает, что в таблице DHCP Snooping нет информации о том, каким образом этот сервер получил IP. Поэтому на него сработает защита. Чтобы этого не было, нужно вручную задать соответствие MAC адреса и ip адреса

Switch#	show ip verify source	показать информацию по IP Source Guard
Switch#	show ip source binding	показать информацию о сопоставлении MAC адреса и IP адреса

*выполняется после включения dhcp snooping

16.5 Dynamic ARP Inspection

Общая конфигурация:

*выполняется после включения dhcp snooping

Switch(config)#	ip arp inspection vlan 1	включение функции Dynamic ARP Inspection (включать за каждый vlan)
Switch(config)#	int fa0/1	переходим на интерфейс fast ethernet 0/1
Switch(config-if)#	ip arp inspection trust	включается на интерфейсах, на которых потенциально не может быть атакующего. Например на uplink (между свитчами)
Switch(config-if)#	ip arp inspection limit rate 2	установить количество arp запросов в секунду не больше 2
Switch(config)#	errdisable recovery cause arp-inspection interval 600	вывести интерфейс из errdisable через 600 секунд (который попадет в errdisable при нарушении arp-inspection)
Switch(config)#	arp access-list ARP-INSPECTION-EXCEPTIONS	создаем специальный arp ACL
Switch(config-std-nacl)#	permit ip host 192.168.1.1 mac host 00:E0:F7:EC:D0:10	эта операция жестко задает соответствие ip адреса и MAC адреса. В данном случае применяется к основному шлюзу. То есть на ответ запроса "MAC адреса для ip 192.168.1.1"должен приходить ответ только из под настоящего шлюза с маком 00:E0:F7:EC:D0:10. Если ответ придет из-под другого MAC адреса, то порт перейдет в errdisable. (При нарушении dhcp snooping порт не переходит в состояние errdisable). Это нужно так же тогда, когда ip адрес задается вручную (в данной ситуации как раз у основного шлюза) и если порт, за которым находится основной шлюз не помечет как trusted (как trusted для Dynamic ARP Inspection, а не для DHCP Snooping)
Switch(config)#	ip arp inspection filter ARP-INSPECTION-EXCEPTIONS vlan 1	применение ACL на VLAN
Switch#	show ip arp inspection	показать состояние Dynamic ARP Inspection
Switch#	show ip arp inspection interface	показать на каких интерфейсах включена функция Dynamic ARP Inspection

17. Power Over Ethernet (PoE)

Общая конфигурация:

Switch(config)#	int e0/1	переходим к редактированию интерфейса ethernet e0/1
Switch(config-if)#	power inline (auto или never)	включаем подачу питания на интерфейсе
Switch(config-if)#	power inline {auto [max milli-watts] | never | static [max milli-watts]}	более дательная настройка подачи питания на интерфейсе
Switch#	show power inline	показать всю информацию по PoE (например сколько осталось ват на каждый порт)

18. Policy-based Routing (PBR)

Общая конфигурация:

Router(config)#	ip access-list extended CTRL-ACL	создаем ACL с именем CTRL-ACL
Router(config-ext-nacl)#	permit ip host 192.168.1.2 any	указываем ip адрес источника, который будет подпадать под действие route-map
Router(config)#	route-map CONTROL-RM	создаем route-map с именем CONTROL-RM
Router(config-route-map)#	match ip address CTRL -ACL	route-map будет срабатывать, если будет срабатывать access-list CTRL-ACL
Router(config-route-map)#	set ip next-hop 10.0.2.1	если сработает route-map, то next-hop будет 10.0.2.1
Router(config)#	int fa0/1	переходим на интерфейс fa0/1, к которому подключен конечный пользователь (или группа конечных пользователей)
Router(config-if)#	ip policy route-map CONTROL-RM	применяем route-map на интерфейс

Диагностика:

Router#	show route-map	показать настройки route-map
Router#	show ip policy	показать интерфейсы, на которых включен route-map
Router#	debug ip policy	включить вывод отладочной информации в режиме реального времени

19. Работа с Cisco IOS

19.1 Обновление прошивки (версии IOS)

Свежий IOS можно скачать с cisco.com. Предварительно для загрузки образа необходимо оплатить SmartNet (для разных устройств цены на эту подписку варьируются).

Подобрать подходящую прошивку для конкретного устройства можно здесь:
http://tools.cisco.com/ITDIT/CFN/jsp/SearchBySoftware.jsp

1. Выясняем сколько места осталось на flash памяти:

```
Router1#sho flash:
```

Выводом команды будет:

```
-#- --length-- -----date/time------ path
1     27624324 Apr 21 2009 03:48:56 c2801-ipbasek9-mz.124-24.T.bin - файл прошивки, который используется в настоящее время
2       2746 Apr 29 2008 13:22:40 sdmconfig-2801.cfg
3       931840 Apr 29 2008 13:23:02 es.tar
4       1505280 Apr 29 2008 13:23:24 common.tar
5       1038 Apr 29 2008 13:23:42 home.shtml
6       112640 Apr 29 2008 13:24:00 home.tar
7       1697952 Apr 29 2008 13:24:32 securedesktop-ios-3.1.1.45-k9.pkg
8       415956 Apr 29 2008 13:24:58 sslclient-win-1.1.4.176.pkg


31686656 bytes available (32309248 bytes used)
```

2. Копируем старый IOS на ftp сервер (чтобы откатиться назад в случае необходимости):

```
Router1#copy flash:c2801-ipbasek9-mz.124-24.T.bin ftp://Maycal:HZmLr16N@172.10.1.2/c2801-ipbasek9-mz.124-24.T.bin
```

Внимание! Если такая команда не сработает, то нужно будет выполнить команду:

```
Router1#copy flash: ftp:
```

и нажать Enter. После этого мастер по шагам предложит ввести все необходимые данные.

3. Если места на flash памяти достаточно, то просто заливаем новый IOS, если нет - удаляем старый.

3.1 Удаляем старый IOS:

```
delete c2801-ipbasek9-mz.124-24.T.bin
Delete filename [c2801-ipbasek9-mz.124-24.T.bin]?
Delete flash:/c2801-ipbasek9-mz.124-24.T.bin? [confirm]
```

3.2 Загружаем новый IOS:

```
Router1#copy ftp://Maycal:HZmLr16N@172.10.1.2/cNEW-ipbasek9-mz.124-24.T.bin flash:cNEW-ipbasek9-mz.124-24.T.bin
```

или

```
Router1#copy tftp: flash:
```

4. Проверяем целостность образа, который мы только что загрузили на наше устройство:

```
Router1#verify /md5 flash:cNEW-ipbasek9-mz.124-24.T.bin
```

В результате мы получим хеш:

```
verify /md5 (flash:cNEWnm-adventerprisek9_ivs_mz.124-24.T1.bin) = e8fab98a72c1516538da7686f8404fcf
```

Этот хеш необходимо сравнить с тем хешом, который указывает производитель (правильный MD5 показывается при скачивании файла с cisco.com). Он должен совпадать, иначе образ был поврежден либо является поддельным.

5. Указываем нашему устройству какой образ использовать при загрузке:

```
Router1(config)#boot system flash:cNEWnm-adventerprisek9_ivs_mz.124-24.T1.bin
```

6. Перезагрузить роутер и проверить работоспособность устройства:

```
Router1#reload
```

19.2 Сброс пароля IOS

Процедура сброса пароля на маршрутизаторах и на коммутаторах Cisco Catalyst отличаются друг от друга.

Все действия производятся только через консольное подключение. При подключении через SSH и Telnet данный метод на сработает.

На маршрутизаторе:

1. **Нам необходимо загрузится в ROMMON.** ROMMON это начальный загрузчик – совсем урезанная версия операционной системы, которая загружается до cisco IOS и используется для сервисных целей (обновление IOS, восстановление пароля).

   Для загрузки в ROMMON необходимо прервать процесс загрузки IOS одной из следующих команд:

Ctrl+Pause Break в Packet Tracer и hyperterminal

Alt+B в teraterm

Команда прерывания загрузки (break sequence) зависит от типа используемого терминала, то есть программы, с помощью которой осуществляется подключение к устройству через его консольный порт:

Программа	Платформа	ОС	Последовательность клавиш
Hyperterminal	IBM Compatible	Windows XP	Ctrl-Break
Hyperterminal	IBM Compatible	Windows 2000	Ctrl-Break
Hyperterminal	IBM Compatible	Windows 98	Ctrl-Break
Hyperterminal (version 595160)	IBM Compatible	Windows 95	Ctrl-F6-Break

Kermit	Sun Workstation	UNIX	Ctrl-\l или Ctrl-\b
MicroPhone Pro	IBM Compatible	Windows	Ctrl-Break
Minicom	IBM Compatible	Linux	Ctrl-a f
ProComm Plus	IBM Compatible	DOS or Windows	Alt-b
SecureCRT	IBM Compatible	Windows	Ctrl-Break
Telix	IBM Compatible	DOS	Ctrl-End
Telnet	N/A	N/A	Ctrl-], then type send brk
Telnet to Cisco	IBM Compatible	N/A	Ctrl-]
Teraterm	IBM Compatible	Windows	Alt-b
Terminal	IBM Compatible	Windows	Break или Ctrl-Break
Tip	Sun Workstation	UNIX	Ctrl-], then Break or Ctrl-c или ~#
VT 100 Emulation	Data General	N/A	F16
Windows NT	IBM Compatible	Windows	Break-F5 или Shift-F5 или Shift-6 Shift-4 Shift-b (^$B)
Z-TERMINAL	Mac	Apple	Command-b
N/A	Break-Out Box	N/A	Connect pin 2 (X-mit) to +V for half a second
	Cisco to aux port	N/A	Control-Shft-6, then b
	IBM Compatible	N/A	Ctrl-Break

2. После того, как мы зашли в ROMMON нам необходимо изменить конфигурационный регистр на 0x2142. Это позволит загрузить наше устройство с начальным running-config. То есть уже имеющийся startup-config не будет загружаться, но останется на устройстве.

```
rommon 2 > confreg 0x2142

rommon 3 > boot
```

3. После команды boot IOS загрузится с нулевой конфигурацией. Заходим в привилегированный exec режим (у нас не будут запрашиваться никакие пароли) и выполняем команду:

```
Router#copy startup-confiig running-config.
```

В результате наш старый startup-config "внедрится" в уже работающий роутер.

Обратите внимание! Мы копируем именно startup-config в running-config, а не наоборот.

4. Устанавливаем новый пароль:

```
Router(config)#enable secret NEW_PASSWORD
Router(config)#username Maycal secret Cisco
```

5. Перезагружаем устройство в ROMMON (прерываем загрузку) и возвращаем конфигурационный регистр обратно на стандартный 0x2102

```
rommon 2 > confreg 0x2102
rommon 3 > boot
```

6. После загрузки мы получим работоспособное устройство с новым паролем без потери старой конфигурации.

Внимание! После проделанных действий все интерфейсы будут в состоянии "administratively down". Их будет необходимо включить вручную.

7. Запрет смены пароля

Если в глобальной конфигурации IOS применить команду no service password-recovery, то будет включена защита ROMMON (ROMMON security). В этом случае, не будет возможности прервать загрузку и войти в ROMMON для изменения конфигурационного регистра. Так же не возможно будет поменять конфигурационный регистр на 0x2142 из самого IOS. При применение этой защиты, восстановить маршрутизатор с сохранением конфигурации будет невозможно – придется сбрасывать роутер до заводских настроек (с нулевым startup-config). Для этого, во время загрузки IOS необходимо нажать и удерживать сочетания клавиш для прерывания загрузки (например ctrl + pause break, зависит от типа используемого терминала). После этого, будет предложено сбросить роутер до заводских настроек.

На коммутаторе Cisco Catalyst:

Следует выключить коммутатор, затем включить, нажать и держать кнопку «Mode» 15 секунд. Это прервёт стандартный процесс загрузки и мы окажемся в приглашении загрузчика. Нам требуется удалить или переименовать конфигурационный файл, чтобы коммутатор загрузился без конфига. Но сразу мы этого сделать не можем, так как голый загрузчик даже не умеет работать с флэш-памятью.

Следует выполнить команды:

```
switch: flash_init
switch: load_helper
```

Теперь мы можем посмотреть содержимое флэш памяти с помощью команды dir flash. Например:

```
switch: dir flash:
```

Выводом команды будет:

```
Directory of flash:
13 drwx 192 Mar 01 1993 22:30:48 c2960-lanbase-mz.122-25.FX
11 -rwx 5825 Mar 01 1993 22:31:59 config.text
18 -rwx 720 Mar 01 1993 02:21:30 vlan.dat
16128000 bytes total (10003456 bytes free)
```

Очевидно, что нас интересует файл config.txt. Если конфигурация коммутатора нам не важна, то файл можно удалить, если требуется только сбросить пароль, но оставить конфигурацию, файл следует переименовать:

```
switch: rename flash:config.text flash:config.text.old
```

После чего можно загружаться:

```
switch: boot
```

Коммутатор запуститься вообще без конфигурации, так как не сможет найти файл config.text. После завершения загрузки, нужно перейти в привилегированный режим, переименовать обратно файл и скопировать содержимое переименованного файла в running-config. Мы загрузили коммутатор с нуливой конфигурацией, так что никто нас уже не спросит про пароль:

```
Switch >enable
Switch#rename flash:/config.text.old flash:/config.text – обратно переименовываем файлы
Switch#copy flash:config.text running-config – копируем содержимое confix.text в running-config
Source filename [config.text]?
Destination filename [running-config]?
```

Теперь конфигурация на месте – осталось сменить пароль:

```
Switch(config)#enable secret NEW_PASSWORD
Switch(config)#username Maycal secret Cisco
```

И сохранить конфигурацию:

```
Switch(config)#end
Switch#copy run start
```

19.3 Восстановление IOS с помощью режима ROMMON

Процедура доступа к ROMON описывалась в разделе «сброс пароля IOS». Нам необходимо загрузиться в ROMMON и скачать с tftp сервера новую прошивку. Обратите внимание, что ftp не поддерживается, только tftp.

Восстановление на маршрутизаторе или коммутаторе Cisco:

rommon>	IP_ADDRESS=192.168.0.1	указываем ip адрес
rommon>	IP_SUBNET_MASK=255.255.255.0	указываем маску подсети
rommon>	DEFAULT_GATEWAY=192.168.0.2	указываем основной шлюз. Даже если наш роутер и tftp сервер находятся в одной канальной среде, все равно нужно указывать основной шлюз. Если сервер и роутер находятся в одной канальной среде, в качестве основного шлюза можно указать адрес tftp сервера
rommon>	TFTP_SERVER=192.168.0.2	указываем ip адрес tftp сервера
rommon>	TFTP_FILE=c2600-ipbasek9-mz.124-13b.bin	указываем файл прошивки
rommon>	set	применяем конфигурацию к движку tftpdnld
rommon>	tftpdnld	выполняем конфигурацию
rommon>	boot	загружаемся в новую прошивку

Восстановление на Cisco ASA:

rommon #3>	ADDRESS=192.168.0.1	указываем ip адрес
rommon #4>	SERVER=192.168.0.2	указываем маску подсети
rommon #5>	GATEWAY=192.168.0.2	указываем основной шлюз. Даже если наш роутер и tftp сервер находятся в одной канальной среде, все равно нужно указывать основной шлюз. Если сервер и роутер находятся в одной канальной среде,

		в качестве основного шлюза можно указать адрес tftp сервера
rommon #6>	IMAGE=f1/asa800-232-k8.bin	указываем файл прошивки
rommon #7>	PORT=Ethernet0/0	указываем интерфейс, через который будет идти обращение к tftp серверу
rommon #8>	set	применяем конфигурацию
rommon #9>	ping server	проверяем доступность сервера
rommon #10>	tftp	загружаем прошивку с tftp сервера
rommon #11>	boot	загружаемся в новую прошивку

19.4 Восстановление порта из состояния err-disabled

Switch(config)#	errdisable recovery cause all	включить автоматическое восстановление порта из состояния err-disable для всех причин, по которым порт перешел в состояние err-disabled
Switch(config)#	errdisable recovery interval <30-86400>	установить таймер автоматического восстановления порта из состояния err-disabled. По умолчанию равен 300 секундам.
Switch#	show interface fa0/1 status	посмотреть, находится ли порт fa0/1 в состоянии err-disabled
Switch#	show interfaces status	состояние всех портов, в том числе состояние err-disabled
Switch#	show errdisable recovery	отображение периода времени, по истечении которого интерфейсы восстанавливаются из состояния err-disabled
Switch#	show errdisable detect	отображение причины состояния err-disabled

* Для ручного восстановления порта из состояния err-disabled необходимо устранить причину возникновения данного состояния, затем выключить порт командой «shutdown» и снова включить командой «no shutdown».

Ссылки на полезные ресурсы

1.	http://www.examcollection.com/100-101.html	дампы экзаменов
2.	http://www.cisco.com/comm/applications/PrepCenter/Images/Vue_CCNATutorial_Tlt_Sim_simlet_v4_010505.swf	эмулятор экзамена Cisco
3.	http://infocisco.ru/cisco_formula_subnetting.html	технология расчета количества хостов и подсетей
4.	http://jodies.de/ipcalc?host=172.16.55.87&mask1=255.255.255.192&mask2= и http://www.ip-ping.ru/netcalc/	автоматический калькулятор сетей
5.	http://www.fssr.ru/hz.php?name=News&file=article&sid=4831	как реализовать атаку ip spoofing
6.	http://www.networksorcery.com/enp/protocol/bootp/options.htm	опции DHCP
7.	http://tools.cisco.com/ITDIT/CFN/jsp/SearchBySoftware.jsp	Cisco Feature Navigator
8.	http://network-class.net/ru/baza-protokolov.html#PPTP	база дампов интернет протоколов
9.	http://safezone.cc/threads/zarezervirovannye-ip-adresa-chto-ehto-takoe-i-s-chem-edjat.23225/	зарезервированные ip адреса
10.	https://ru.wikipedia.org/wiki/%D0%A1%D0%BF%D0%B8%D1%81%D0%BE%D0%BA_%D0%BF%D1%80%D0%BE%D1%82%D0%BE%D0%BA%D0%BE%D0%BB%D0%BE%D0%B2,_%D0%B8%D0%BD%D0%BA%D0%B0%D0%BF%D1%81%D1%83%D0%BB%D0%B8%D1%80%D1%83%D0%B5%D0%BC%D1%8B%D1%85_%D0%B2_IP	вложения в IP
11.	http://www.cisco.com/c/en/us/products/index.html	полный модельный ряд оборудования Cisco
12.	http://nnetwork.ru/	официальный весселер Cisco (интернет-магазин оборудования Cisco)

Содержание второй части книги

Во второй части книги и видеокурса будут доступны следующие темы: