BGP или как украсть интернет

Мирослав Берков alumni@samaradom.ru

В апреле этого года не совсем верные данные маршрутизации были получены китайским государственным поставщиком услуг Интернета China Telecom, а затем распространились по всей сети, что повлияло на Интернет трафик главных корпоративных, военных и правительственных сайтов США. Трафик, большая часть которого берет начало в США, предназначался американским корпоративным и правительственным сайтам, и должен был пройти кратчайший возможный путь и, естественно, не через Китай. Определенный трафик направлялся сайтам, принадлежащим американскому Сенату, Министерству Обороны, НАСА и Министерству Торговли.

Как такое вообще возможно в современном мире? Конечно, всего предусмотреть в нашей работе нельзя, и все же, давайте посмотрим, как это работает.

Intro или как Пакистан отключал YouTube

Хроника событий:
Вообще YouTube анонсирует 5 префиксов во всемирную сеть, это:
— /19, /20, /22, и два /24-ых
— где 22 — это 208.65.152.0/22 (для тех кто не знает что это такое, можете посмотреть здесь http://bgp.he.net/ip/208.65.152.0)

И вот однажды Правительство Пакистана решает заблокировать YouTube и дает указание по этому поводу своему национальному оператору связи.

Пакистан-Телеком по своему усмотрению устанавливает более короткий маршрут (208.65.153.0/24) для префикса YouTube/22 до null0 интерфейса. Да, действительно в Cisco есть замечательный интерфейс Null0, конфигурируется он всего одной командой:

```
int Null0
ip unreachables
```

Теперь достаточно добавить еще один маршрут в конфигурацию Cisco (как раз наш любимый YouTube это сеть класса C — 208.65.153.0/24)

```
ip route 208.65.153.0 255.255.255.0 Null 0 100
```

В этом случае, если адрес используется, и маршрут на него известен Cisco, то именно этот маршрут и будет активен (поскольку его метрика меньше), если же адрес неизвестен, то активным станет маршрут на Null0 и Null0 ответит на пришедший пакет icmp. Таким образом в Пакистане YouTube уже не посмотрят…

Кстати, можно еще прописать такие же маршруты для внутренних сетей, это предотвратит их случайное выбрасывание во внешний мир.

```
ip route 10.0.0.0 255.0.0.0 Null0 100
ip route 172.16.0.0 255.240.0.0 Null0 100
```

ip route 192.168.0.0 255.255.0.0 Null0 100

Но по недосмотру или неграмотности персонала Пакистан-Телеком этот префикс начал объявляться вышестоящим провайдерам, которые в свою очередь начали рассылать этот маршрут для всех остальных сетей по всему миру, и тут началось…

Большинство абонентов, желающих посмотреть YouTube, понесло в Пакистан, где конечно они не увидели ничего!

YouTube в ответ на действия Пакистан-Телеком стала объявлять как /24 и еще два /25 префикса, что принесло к частичной победе над действиями Пакистан-Телеком

Двумя часами позднее PCCW в результате происшествия выключает Пакистан-Телеком из системы глобального обмена маршрутной информацией

Ну и не прошло 3..5 минут после отключения Пакистана, как связь всего мира и YouTube была восстановлена

Как это работает?

BGP (англ. *Border Gateway Protocol*, протокол граничного шлюза) — основной протокол динамической маршрутизации в Интернете, в отличие от других протоколов динамической маршрутизации, предназначен для обмена информацией о маршрутах не между отдельными маршрутизаторами, а между целыми автономными системами, и поэтому, помимо информации о маршрутах в сети, переносит также информацию о маршрутах на автономные системы. BGP не использует технические метрики, а осуществляет выбор наилучшего маршрута исходя из правил, принятых в сети.

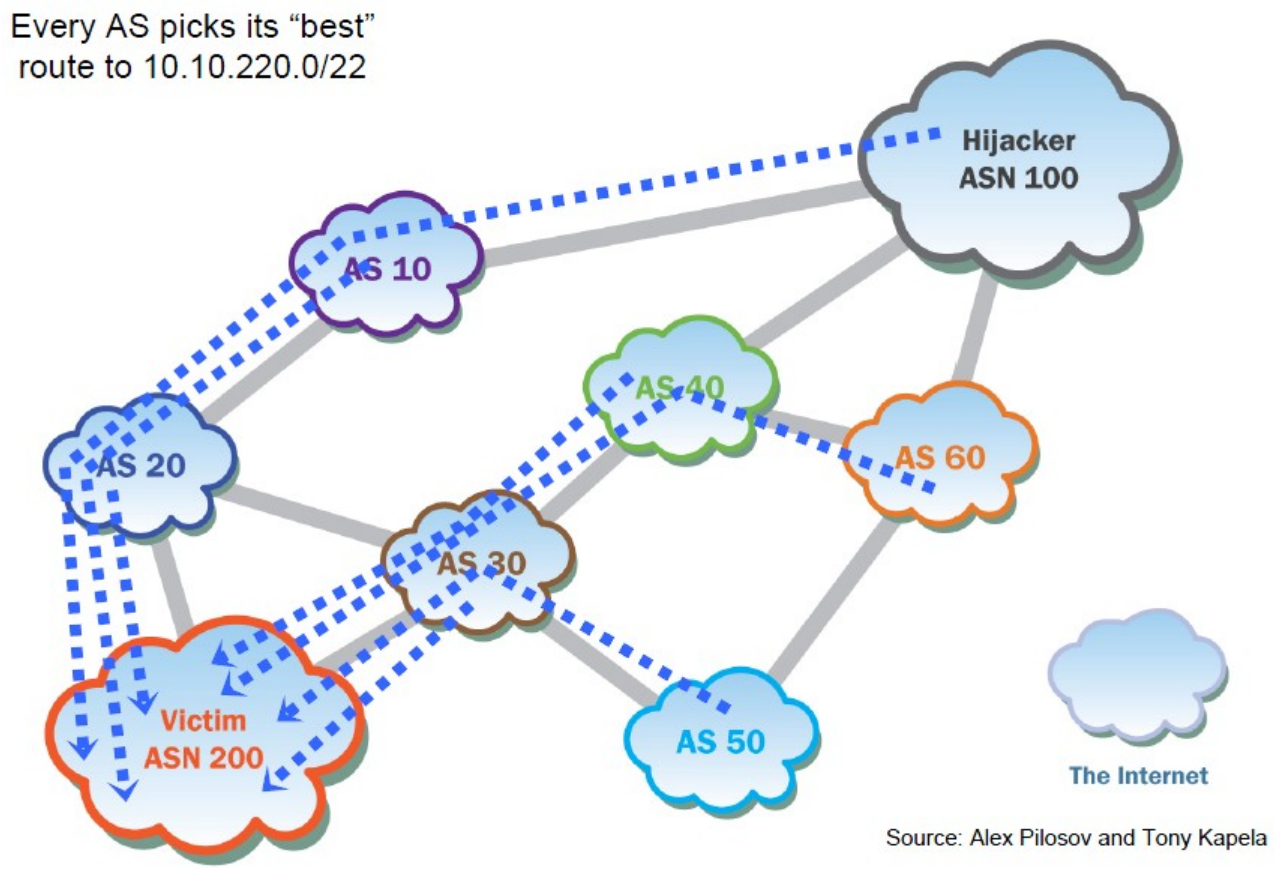
Родина слышит все с первого раза

Ранее перехватывать интернет-трафик, используя уязвимость BGP, были способны только разведывательные агентства. Они знали о недостатках протокола с 1998 года, с тех пор как эксперт по безопасности Пейтер Затко (Peiter Zatko) рассказал о ней Конгрессу США и предположил, что может за полчаса нарушить работу всего Интернета, используя обнаруженную уязвимость. Вероятно, но не факт конечно, что спецслужбы разных стран грешат этим и по сей день, в случае, если нужно перехватить трафик абонентов некоторых автономных систем, к которым у них нету физического доступа, находящихся в неподконтрольных или оффшорных юрисдикциях, сотрудничество с правоохранительными органами которых затруднительно.

Предположим, что автономная система AS 200 (а ведь точно, что сейчас любой провайдер или крупная компания имеют свои собственные автономные системы, и причем, некоторые даже не одну) приземляет свой префикс 10.10.220.0/22, рассылая его объявления для AS 20 и AS 30, интернет работает по правильному маршруту. Forwarding Information Base (FIB) для 10.10.220.0/22 до и после обновления содержит корректную информацию – то есть, мы пока еще ничего не меняли и все соответствует установленным правилам маршрутизации.

Сценарий атаки MITM на BGP

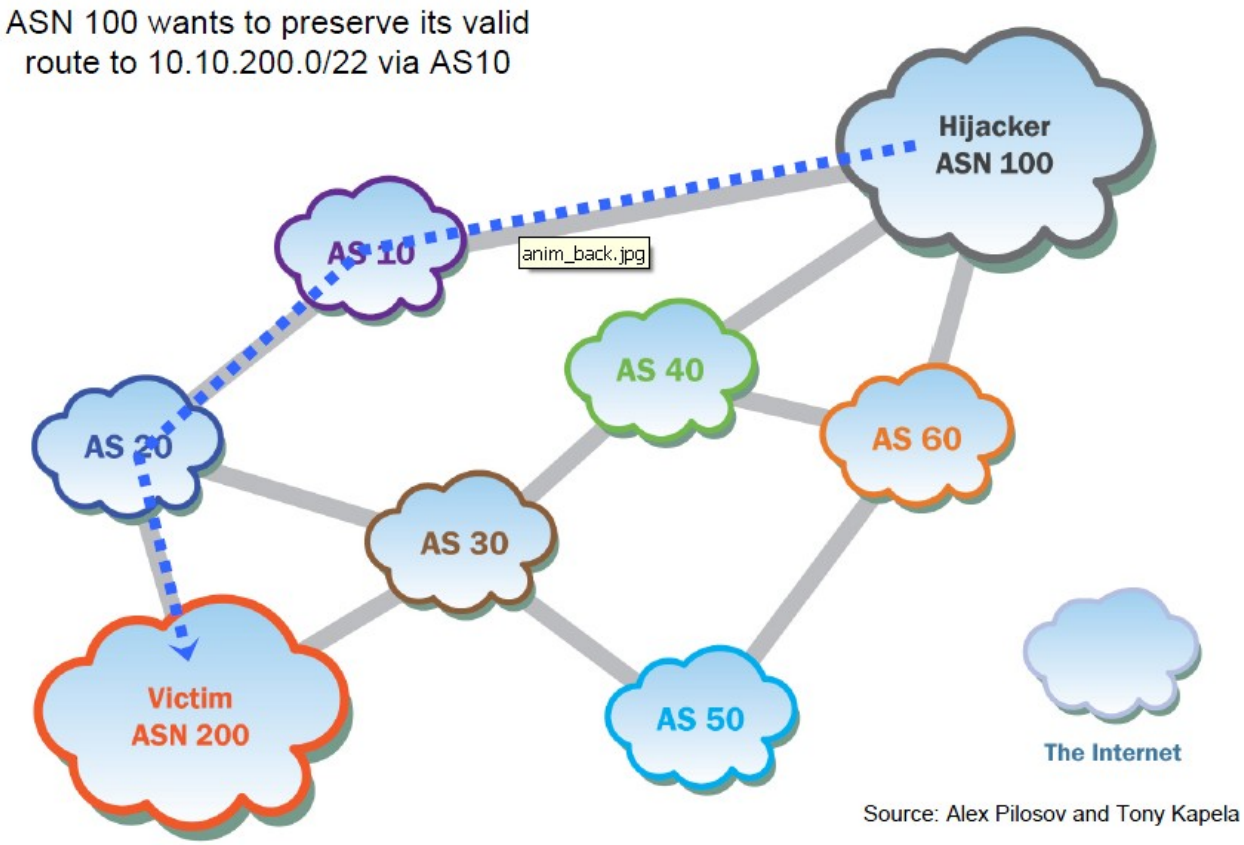
1. Вычисляем трассу и планируем расстояние до цели (целевой AS, трафик которой мы собираемся перехватить)
2. Смотрим, через какие AS лежит путь, и сохраняем их номера в нашей системе
3. Делаем так, чтобы наш новый маршрут не рассылался этим AS, которые будут пересылать трафик в конечную систему AS 200
4. Вписываем статический маршрут до конечной AS через первую граничную AS для префикса, который собираемся объявлять
5. Поехали



Сейчас каждая AS содержит правильную информацию о маршрутах и путях для префикса 10.10.220.0/22 к AS 200.

Планируем путь до цели

Очевидно, что для AS 10, 20 и 200 мы префикс из AS 200 объявлять не будем, - это нужно чтобы трафик все-таки дошел до абонентов, после того как он будет профильтрован, скопирован или модифицирован как вам будет удобно.



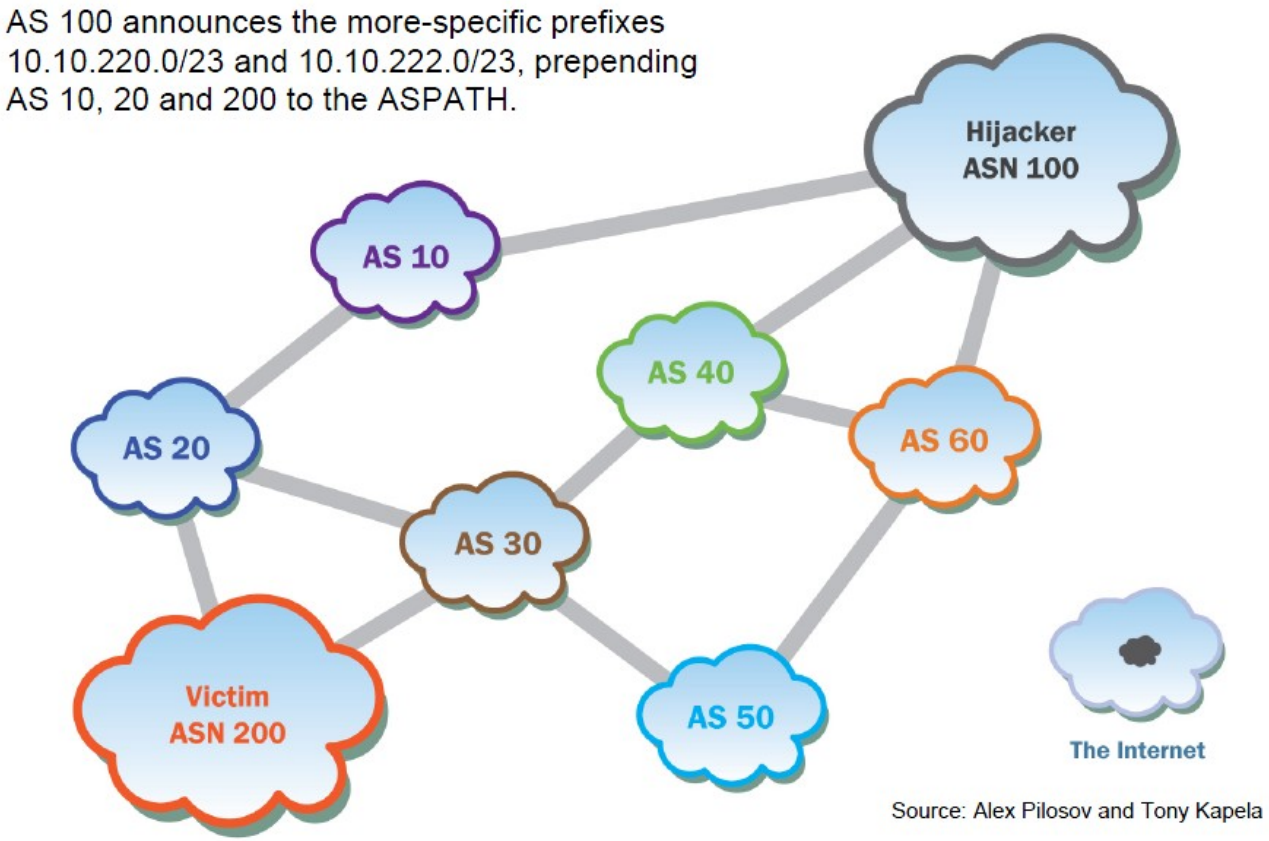
Итак, наша AS 100 хочет пропустить весь «полученный по ошибке» трафик префикса 10.10.220.0/22 через AS 10 и далее в пункт назначения – в AS 200

Прописываем глобальный маршрут

префикс 10.10.220.0/24 (23) теперь объявляется нашей AS 100 через route-map:

```
route-map intercept permit 10
match ip address prefix-list ACA
set as-path prepend 10 20 200
ip prefix-list ACA seq 5 permit 10.10.220.0/23
ip prefix-list ACA seq 5 permit 10.10.222.0/23
```

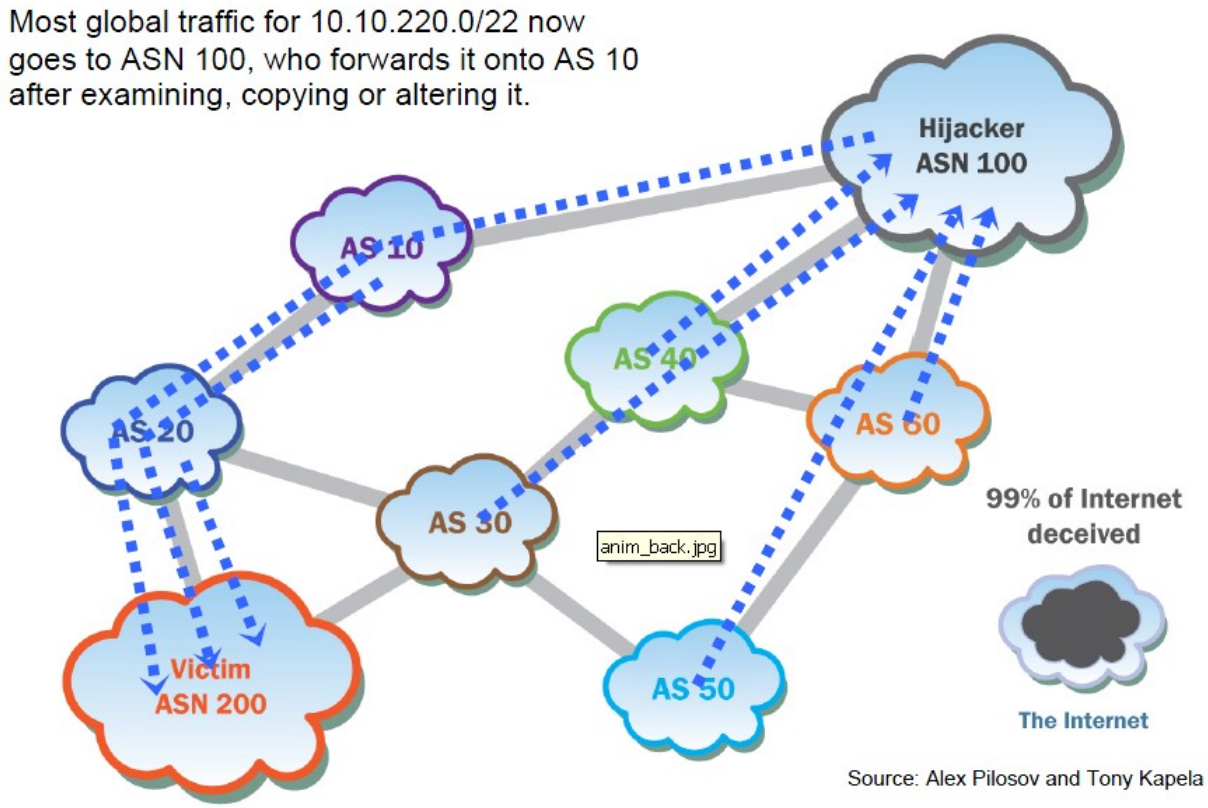
как и говорилось, в AS 10, 20 и 200 мы его не отдаем, пусть они работают, как они и работали.



AS 100 теперь рассылает особенные префиксы 10.10.220.0/23 и 10.10.222.0/23 (разумеется они включают в себя нужный тебе префикс /24) , при этом не передавая обновленную информацию о маршрутах в AS 10, 20 и 200.

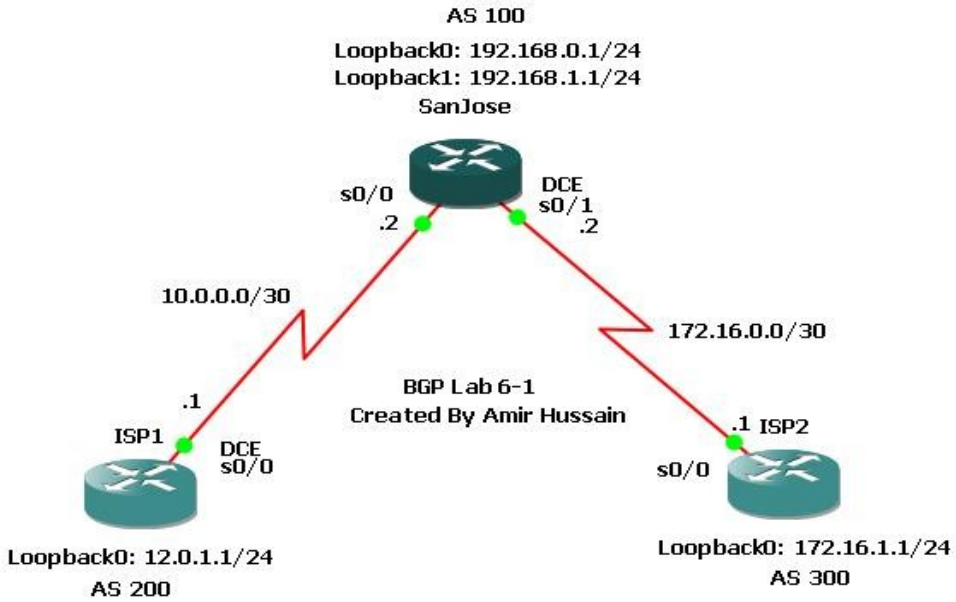
Теперь ручками прописываем статический маршрут в системе AS 100 для префикса 10.10.220.0/24 – чтобы он уходил напрямую в AS 10:

```
ip route 10.10.220.0 255.255.255.0 4.3.2.1
```



Теперь 99% общемирового трафика для 10.10.220.0/22, предназначенного AS 200 проходит через AS 100, далее через AS 10 и AS 20 и потом только попадает в AS 200, разумеется, после его просмотра, копирования или модификации (в случае если требуется).

Учимся настраивать BGP - маршрут по умолчанию



Теперь рассмотрим более конкретный пример. Сеть банка подключена к двум провайдерам интернета. Нужно сконфигурировать сеть так, чтобы обмен информацией о маршрутах осуществлялся по протоколу BGP между банком и этими провайдерами. Я ставил эксперименты на стенде, собранном из DYNAGEN и Cisco IOS 3640, но ты как настоящий гуру девайсов Cisco, конечно же, можешь все это проделать и на

горячей киске, где и как ее можно добыть есть в ранее выложенных видео на канале http://www.youtube.com/user/XakepRUVideo

Задача

Международный платежный банк активно использует систему on-line доступа к счетам для обслуживания платежей клиентов. Его бизнес целиком и полностью зависит от доступности услуг интернета. Банк пользуется услугами двух интернет провайдеров. Требуется сконфигурировать BGP для обмена маршрутами между пограничным роутером банка в San Jose и двумя роутерами провайдеров интернета.

Прописываем IP адреса

Сконфигурируем сеть, как показано на рисунке, но без протоколов маршрутизации, для loopback интерфейсов зададим IP статические адреса, они будут присутствовать вместо реальных сетей для проверки их достижимости на трассе от интернет-провайдеров до роутера банка. **Ping** используется для проверки соединений между роутерами, непосредственно подключенными друг к другу, причем роутер ISP1 недостижим для роутера ISP2, потому как AS банка не является транзитной.

Прописываем BGP в конфиги роутеров

Конфигурируем роутеры провайдеров и международного банка (San Jose).

В роутер первого провайдера ISP1 вводим следующий конфиг:

```
ISP1(config)#router bgp 200
ISP1(config-router)#neighbor 10.0.0.2 remote-as 100
ISP1(config-router)#network 12.0.1.0 mask 255.255.255.0
```

В роутер второго провайдера ISP2 вводим аналогичный конфиг BGP:

```
ISP2(config)#router bgp 300
ISP2(config-router)#neighbor 172.16.0.2 remote-as 100
ISP2(config-router)#network 172.16.1.0 mask 255.255.255.0
```

Теперь конфигурируем роутер банка в San Jose для общения по BGP с роутерами обоих провайдеров:

```
SanJose(config)#router bgp 100
SanJose(config-router)#neighbor 10.0.0.1 remote-as 200
SanJose(config-router)#neighbor 172.16.0.1 remote-as 300
SanJose(config-router)#network 192.168.0.0
SanJose(config-router)#network 192.168.1.0
```

Чтобы убедится в правильности проведенной конфигурации, проверим таблицу маршрутизации банка San Jose при помощи команды **show ip route**

```
SanJose#show ip route
Gateway of last resort is not set
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:00:03
10.0.0.0/30 is subnetted, 1 subnets
C 10.0.0.0 is directly connected, Serial0/0/0
C 192.168.0.0/24 is directly connected, Loopback0
12.0.0.0/24 is subnetted, 1 subnets
B 12.0.1.0 [20/0] via 10.0.0.1, 00:00:42
C 192.168.1.0/24 is directly connected, Loopback1
```

Роутер банка в San Jose имеет маршруты до loopback-ов (сетей) каждого провайдера. Давайте удостоверимся в этом, пропинговав каждый loopback из консоли роутера банка. Как вариант можно написать собственный TCL сценарий, который будет пинговать их каждые 30 секунд.

Теперь убедимся, что BGP в банке San Jose работает правильно, выполнив команду **show ip bgp**

```
SanJose#show ip bgp
BGP table version is 5, local router ID is 192.168.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal
Origin codes: i - IGP, e - EGP, ? - incomplete
Network Next Hop Metric LocPrf Weight Path
*> 12.0.1.0/24 10.0.0.1 0 0 200 i
*> 172.16.1.0/24 172.16.0.1 0 0 300 i
*> 192.168.0.0 0.0.0.0 0 32768 i
*> 192.168.1.0 0.0.0.0 0 32768 i
```

Звездочка (*) обозначает наилучший маршрут, галочка (>) указывает на то, что маршрут присутствует в таблице маршрутизации роутера. Можно немного поиграть с обновлениями, на роутере ISP1 выполним команду **shutdown** для интерфейса Loopback 0. Затем на роутере банка выполним команду **show ip bgp** снова. Это привело к обновлению таблицы маршрутизации (примерно как в случае с Пакистан Телеком, когда был добавлен Null 0 интерфейс в систему). Теперь снова включим Loopback0 интерфейс роутера ISP1 выполнив команду **no shutdown**. На роутере банка SanJose выполним команду **show ip bgp neighbors** и увидим примерно следующее:

```
BGP neighbor is 172.16.0.1, remote AS 300, external link
Index 2, Offset 0, Mask 0x4
BGP version 4, remote router ID 172.16.1.1
BGP state = Established, table version = 5, up for 00:02:24
Last read 00:00:24, hold time is 180
```

Из этого следует, что сессия BGP с ISP2 установлена и коннект длится уже 2 минуты 24 секунды.

Фильтруем маршруты

Проверим таблицу маршрутизации ISP2, выполнив команду **show ip route**. ISP2 должен иметь маршрут к сети 12.0.1.0, принадлежащей ISP1. Если роутер банка в San Jose рассылает объявления маршрутов, принадлежащих ISP1, то роутер ISP2 включит их в свою таблицу маршрутизации. После такого обновления ISP2 по идее должен пытаться маршрутизировать транзитный трафик через роутер банка. Вот так, давайте сконфигурируем роутер банка SanJose таким образом, чтобы он объявлял только сети банка 192.168.0.0 и 192.168.1.0 обоим провайдерам. На роутере банка SanJose зададим следующий список доступа:

```
SanJose(config)#access-list 1 permit 192.168.0.0 0.0.1.255
```

Для использования этого списка доступа в качестве фильтра используем **distribute-list** при определении BGP **neighbor** следующим образом:

```
SanJose(config)#router bgp 100
SanJose(config-router)#neighbor 10.0.0.1 distribute-list 1 out
SanJose(config-router)#neighbor 172.16.0.1 distribute-list 1 out
```

После того как мы применили список доступа к исходящим объявлениям, давайте проверим таблицу маршрутизации ISP2 снова. Очевидно, что маршрут до сети 12.0.1.0, принадлежащей ISP1 все еще присутствует в таблице роутера ISP2. Вернемся в консоль роутера банка SanJose и выполним команду **clear ip bgp ***. Подождем, пока роутер установит BGP соединения заново, это может занять какое-то время, и перепроверим таблицу маршрутизации ISP2 еще раз. Маршрут до сети 12.0.1.0, принадлежащей ISP1, исчез из таблицы маршрутизации ISP2. Маршрут до сети 172.16.1.0, принадлежащей ISP2, исчез из таблицы маршрутизации ISP1.

Настройка основного и резервного канала банка
с использованием плавающих статических маршрутов

Теперь, когда соединение с использованием BGP с каждым интернет провайдером установлено, самое время перейти к настройке основного и запасного каналов. Это можно реализовать при помощи статической плавающей маршрутизации BGP. Для начала выполним команду **show ip route** на роутере банка SanJose:

```
Gateway of last resort is not set
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:07:37
```

Сейчас шлюз по умолчанию не определен. Это огромная проблема, потому как роутер банка SanJose является шлюзом для всей его корпоративной сети. Возьмем в качестве основного провайдера ISP1, и ISP2 в качестве резервного. Сконфигурируем статические маршруты, отражающие данную политику:

```
SanJose(config)#ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
SanJose(config)#ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

Теперь убедимся, что маршрут по умолчанию определен, выполнив команду **show ip route** на роутере банка SanJose:

```
Gateway of last resort is 10.0.0.1 to network 0.0.0.0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:16:34
```

Теперь протестируем маршрут по умолчанию, создав не объявляемый по BGP loopback на роутере ISP1:

```
ISP1#config t
ISP1(config)#int loopback 100
ISP1(config-if)#ip address 210.210.210.1 255.255.255.0
```

Теперь выполним команду **clear ip bgp 10.0.0.1** для разрыва и установления BGP сессии вновь с роутером 10.0.0.1 (ISP1):

```
SanJose#clear ip bgp 10.0.0.1
```

Немного подождав, пока BGP сессия с ISP1 возобновится, выполним команду **show ip route** чтобы убедиться, что вновь добавленная сеть 210.210.210.0/24 не появилась в таблице маршрутизации роутера банка SanJose:

```
SanJose#show ip route
Gateway of last resort is 10.0.0.1 to network 0.0.0.0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:27:40
```

Теперь пропингуем 210.210.210.1 loopback из консоли роутера банка SanJose:

```
SanJose#ping
Protocol [ip]:
Target IP address: 210.210.210.1
Sending 5, 100-byte ICMP Echos to 210.210.210.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/32/36 ms
```

Шлюзом по умолчанию в данной конфигурации является ISP1, что и требовалось доказать.

Настройка основного и резервного канала банка с использованием статических маршрутов
Есть другой способ настройки основного и резервного каналов с использованием команды **defaultnetwork** вместо определения маршрутов по умолчанию 0.0.0.0/0 как в предыдущем примере. Итак, удалим плавающие статические маршруты из конфигурации роутера банка San Jose:

```
SanJose(config)#no ip route 0.0.0.0 0.0.0.0 10.0.0.1 210
SanJose(config)#no ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

Сделаем так, чтобы сеть 210.210.210.0/24 теперь объявлялась роутером ISP1 своим соседям по BGP:.

```
ISP1(config)#router bgp 200
ISP1(config-router)#network 210.210.210.0
ISP1#clear ip bgp 10.0.0.2
```

Сконфигурируем роутер банка SanJose, объявив сесть по умолчанию командой **default-network** для переопределения шлюза по умолчанию. Прежде чем объявлять сеть 210.210.210.0/24 по умолчанию с помощью команды **ip default-network**, убедимся что эта сеть присутствует в таблице маршрутизации:

```
Gateway of last resort is not set
B 210.210.210.0/24 [20/0] via 10.0.0.1, 00:04:51
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:21:19

SanJose(config)#ip default-network 210.210.210.0
```

Немного подождав, запросим таблицу маршрутизации роутера банка San Jose еще раз:

```
Gateway of last resort is 10.0.0.1 to network 210.210.210.0
B* 210.210.210.0/24 [20/0] via 10.0.0.1, 00:04:28
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:20:56
```

В этом случае ISP1 установлен как единственный шлюз по умолчанию. Этот маршрут может быть модифицирован с использованием политик маршрутизации. Исправим ситуацию добавив резервный маршрут к ISP2:

```
SanJose(config)#ip route 0.0.0.0 0.0.0.0 172.16.0.1 220
```

EBGP (внешний BGP) получает маршруты с административной дистанцией 20 и они являются предпочтительнее любых других маршрутов с административной дистанцией более 20, например, маршрут по умолчанию задан нами сейчас с административной дистанцией 220.

Давайте проверим, что вновь добавленный маршрут представляет из себя резервный маршрут по умолчанию, в то время как BGP сессия между роутером банка SanJose и провайдером ISP1 восстановлена после обновления.

```
SanJose#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile,
B - BGP D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area N1
- OSPF NSSA external type 1, N2 - OSPF NSSA external
type 2 E1 - OSPF external type 1, E2 - OSPF external type 2,
E - EGP i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS
inter area * - candidate default, U - per-user static route, o - ODR P -
periodic downloaded static route
Gateway of last resort is 10.0.0.1 to network 210.210.210.0
```

```
B* 210.210.210.0/24 [20/0] via 10.0.0.1, 00:19:17
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:35:45
10.0.0.0/30 is subnetted, 1 subnets
C 10.0.0.0 is directly connected, Serial0/0/0
C 192.168.0.0/24 is directly connected, Loopback0
12.0.0.0/24 is subnetted, 1 subnets
B 12.0.1.0 [20/0] via 10.0.0.1, 00:19:17
C 192.168.1.0/24 is directly connected, Loopback1
S* 0.0.0.0/0 [220/0] via 172.16.0.1
```

Заметим, что таблица маршрутизации имеет два маршрута по умолчанию (они отмечены *), но только один из них будет использоваться, потому что заданы различные административные дистанции.

```
SanJose#clear ip bgp 10.0.0.1

SanJose#show ip route
Gateway of last resort is 172.16.0.1 to network 0.0.0.0
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:45:31
```

Этот маршрут по умолчанию будет задействован только в том случае, когда сеть 210.210.210.0/24 случайно окажется недостижимой по причине отсутствия связи или в течение непродолжительного времени, которое нужно на восстановление BGP сессии после выполнения команды **clear ip bgp 10.0.0.1** если такое потребуется.

```
SanJose#show ip route
Gateway of last resort is 10.0.0.1 to network 210.210.210.0
B* 210.210.210.0/24 [20/0] via 10.0.0.1, 00:01:03
172.16.0.0/16 is variably subnetted, 2 subnets, 2 masks
C 172.16.0.0/30 is directly connected, Serial0/0/1
B 172.16.1.0/24 [20/0] via 172.16.0.1, 00:46:42
```

Как и ожидалось, пока поднималась BGP сессия между роутером банка SanJose и провайдером ISP1, маршрут до ISP2 был задействован в качестве маршрута по умолчанию. Однако, как только BGP сессия была восстановлена между роутерами SanJose и ISP1, маршрутом по умолчанию вновь стал 210.210.210.0 для роутера банка San Jose.

Outro

Инциденты с BGP происходят регулярно, по мелочи что-то подобное происходит каждый день, и как ты убедился сам, BGP работает достаточно просто. Команды **show**, как было описано в практической части, ты сможешь потренироваться запускать на реальном **BGP** роутере, например вот здесь http://ospfmon.com, только не вздумай переводить роутер в режим **enable** и менять таблицы маршрутизации, там все предусмотрено (ну или почти все) и вряд ли у тебя это получится. Конечно, привилегированный доступ к железке (level_15 access), где есть **BGP**, есть далеко не у каждого, даже дежурные инженеры телекоммуникационных компаний обычно имеют доступ к роутерам в режиме **Exec** (только просмотр) именно для того, чтобы они могли контролировать пересылки маршрутов из одной AS в другую в ручную на глаз. Тем не менее, как пойдет трафик через минуту и какие AS будут в этом участвовать – предсказать не может никто, в случае если в процесс маршрутизации вмешивается постороннее лицо. Вот именно поэтому обновления маршрутов и отслеживают. Остается только смотреть на монитор чаще и внимательнее и вовремя решать возникающие проблемы.