Маршрутизация в соответствии с политикой и ее влияние на ESP и пакеты ISAKMP с Cisco IOS

Содержание

Введение

Этот документ описывает эффект Маршрутизации на основе политик (PBR) и локального PBR, когда применился к пакетам Безопасного закрытия полезной нагрузки (ESP) и Протокола ISAKMP, когда вы используете Cisco IOS®.

Внесенный Михалом Гаркарзом, специалистом службы технической поддержки Cisco.

Предварительные условия

Требования

Cisco рекомендует иметь базовые знания об этих темах:

- Cisco IOS
- Конфигурация VPN на Cisco IOS

Используемые компоненты

Сведения в этом документе основываются на версии Cisco IOS 15. x.

Сведения, представленные в этом документе, были получены от устройств, работающих в специальной лабораторной среде. Все устройства, описанные в этом документе, были запущены с чистой (стандартной) конфигурацией. В рабочей сети необходимо изучить потенциальное воздействие всех команд до их использования.

Общие сведения

До установления Туннеля IPSec маршрутизатор инициирует обмен ISAKMP. Поскольку те пакеты генерируются маршрутизатором, пакеты рассматриваются как локально формируемый трафик, и применены любые локальные решения PBR. Кроме того, любые пакеты, генерируемые маршрутизатором (Протокол EIGRP, Протокол NHRP, Протокол BGP или эхо-запросы Протокола ICMP), также рассматривают как локально формируемый трафик и имеют локальное примененное решение PBR.

Трафик, который передан маршрутизатором и передан через туннель, который называют транзитным трафиком, не считают локально формируемым трафиком, и любая желаемая политика маршрутизации должна быть применена на входной интерфейс маршрутизатора.

Последствия, которые это имеет на трафике, который пересекает туннель, - то, что локально формируемый трафик придерживается PBR, но транзитный трафик не делает. Эта статья объясняет последствия этого различия в поведении.

Для транзитного трафика, который должен быть ESP, инкапсулировал, нет никакой потребности иметь любые записи маршрутизации, потому что PBR определяет исходящий интерфейс для пакета прежде и после ESP инкапсуляция. Для локально формируемого трафика, который должен быть ESP, инкапсулировал, необходимо иметь записи маршрутизации, потому что локальный PBR определяет исходящий интерфейс только для пакета перед инкапсуляцией, и маршрутизация определяет исходящий интерфейс для постинкапсулированного пакета.

Этот документ содержит пример типичной конфигурации, где используется один маршрутизатор с двумя каналами поставщика. Одна ссылка используется для доступа к Интернету, и второе для VPN. В случае любого отказа соединения трафик перенаправлен с другой ссылкой интернет-провайдера (ISP). Ловушки также представлены.

Заметьте, что PBR выполнен в технологии CEF, тогда как локальный PBR является процессной коммутацией.

Трафик, генерируемый локально на маршрутизаторе

В этом разделе описываются поведение трафика, инициируемого от маршрутизатора (R) 1. Тот трафик ESP инкапсулируется R1.

Топология

IPSec-туннель между локальными сетями создан между R1 и R3.

Представляющий интерес трафик между R1 Lo0 (192.168.100.1) и R3 Lo0 (192.168.200.1).

Маршрутизатор R3 имеет маршрут по умолчанию к R2.

R1 не имеет никаких записей маршрутизации, только непосредственно связанные сети.

!--- конфигурацию

R1 имеет локальный PBR для всего трафика:

```
interface Loopback0
 ip address 192.168.100.1 255.255.255.0
!
interface Ethernet0/0
 ip address 192.168.0.1 255.255.255.0
 crypto map CM

track 10 ip sla 10
ip sla 10
 icmp-echo 192.168.0.2 source-ip 192.168.0.1

route-map LOCALPBR permit 10
 set ip next-hop verify-availability 192.168.0.2 1 track 10
ip local policy route-map LOCALPBR
```

Отладка

Весь локально формируемый трафик на R1 передается R2, когда это подключено UP.

Для проверки то, что происходит, когда вы переводите туннель в рабочее состояние, передайте представляющий интерес трафик от самого маршрутизатора:

```
R1#debug ip packet
R1#ping 192.168.200.1 source lo0
```

> **Внимание. : Команда debug ip packet** могла бы генерировать большое количество отладок и оказывает огромное влияние на использование ЦПУ. Используйте его с осторожностью.

Эта отладка также позволяет access-list использоваться для ограничения объема трафика, обработанного отладками. **Команда debug ip packet** только отображает трафик, который является процессной коммутацией.

Вот отладки на R1:

```
IP: s=192.168.100.1 (local), d=192.168.200.1 (Ethernet0/0), len 100, local
feature, Policy Routing(3), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk
FALSE
IP: s=192.168.100.1 (local), d=192.168.200.1 (Ethernet0/0), len 100, sending
IP: s=192.168.100.1, d=192.168.200.1, pak EF6E8F28 consumed in output feature,
packet consumed, IPSec output classification(30), rtype 2, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, local feature, Policy
Routing(3), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, sending
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
IPSec output classification(30), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
IPSec: to crypto engine(64), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
Post-encryption output features(65), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, post-encap feature,
(1), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, post-encap feature,
FastEther Channel(3), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, sending full packet
```

Вот то, что происходит:

С представляющим интерес трафиком (192.168.100.1> 192.168.200.1) совпадает локальный PBR, и исходящий интерфейс определен (E0/0). Это действие инициирует крипто-код для инициирования ISAKMP. Тот пакет также маршрутизируется политикой локальным PBR, который определяет исходящий интерфейс (E0/0). Трафик ISAKMP передается, и о туннеле выполняют согласование

Когда вы пропинговываете снова, что происходит?

```
R1#show crypto session
Crypto session current status

Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 10.0.0.2 port 500
 IKEv1 SA: local 192.168.0.1/500 remote 10.0.0.2/500 Active
 IPSEC FLOW: permit ip host 192.168.100.1 host 192.168.200.1
       Active SAs: 2, origin: crypto map

R1#ping 192.168.200.1 source lo0 repeat 1

IP: s=192.168.100.1 (local), d=192.168.200.1 (Ethernet0/0), len 100, local
feature, Policy Routing(3), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.100.1 (local), d=192.168.200.1 (Ethernet0/0), len 100, sending
IP: s=192.168.100.1 (local), d=192.168.200.1 (Ethernet0/0), len 100, output
feature, IPSec output classification(30), rtype 2, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
IP: s=192.168.100.1, d=192.168.200.1, pak EEB40198 consumed in output feature,
```

```
packet consumed, IPSec: to crypto engine(64), rtype 2, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, output feature,
IPSec output classification(30), rtype 1, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, output feature,
IPSec: to crypto engine(64), rtype 1, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, output feature,
Post-encryption output features(65), rtype 1, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), g=10.0.0.2, len 172,
forward
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, post-encap
feature, (1), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, post-encap
feature, FastEther Channel(3), rtype 0, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 172, encapsulation
failed.
Success rate is 0 percent (0/1)
```

Вот то, что происходит:

Локально генерируемый представляющий интерес трафик, 192.168.100.1> 192.168.200.1, локально маршрутизируется политикой, и исходящий интерфейс определен (E0/0). Пакет используется функцией обработки исходящих данных IPsec на E0/0 и инкапсулируется. Инкапсулированный пакет (от 192.168.0. От 1 до 10. 0.0.2), проверен для маршрутизации для определения исходящего интерфейса, но нет ничего в таблицах маршрутизации R1, который является, почему отказывает инкапсуляция.

В этом сценарии туннель подключен UP, но трафик не передается, потому что, после ESP инкапсуляция, Cisco IOS проверяет таблицы маршрутизации для определения исходящего интерфейса.

Транзитный трафик через маршрутизатор

В этом разделе описываются поведение для транзитного трафика, который проникает через маршрутизатор, который ESP инкапсулируется тем маршрутизатором.

Топология

Туннель L2L создан между R1 и R3.

Представляющий интерес трафик между R4 (192.168.100.1) и R3 lo0 (192.168.200.1).

Маршрутизатор R3 имеет маршрут по умолчанию к R2.

Маршрутизатор R4 имеет маршрут по умолчанию к R1.

R1 не имеет никакой маршрутизации.

!--- конфигурацию

Когда маршрутизатор получает пакеты для шифрования (транзитный трафик вместо локально формируемого трафика), предыдущая топология модифицируется для показа потока.

Прямо сейчас представляющий интерес трафик, полученный от R4, маршрутизируется политикой на R1 (PBR на E0/1), и существует также маршрутизация локальной политики для всего трафика:

```
interface Ethernet0/1
 ip address 192.168.100.10 255.255.255.0
 ip policy route-map PBR

route-map LOCALPBR permit 10
 set ip next-hop verify-availability 192.168.0.2 1 track 10
!
route-map PBR permit 10
 set ip next-hop verify-availability 192.168.0.2 1 track 10

ip local policy route-map LOCALPBR
```

Отладка

Для проверки то, что происходит, когда вы переводите туннель в рабочее состояние на R1 (после получения представляющего интерес трафика от R4) войдите:

R1#**debug ip packet** R4#**ping 192.168.200.1**

Вот отладки на R1:

```
IP: s=192.168.100.1 (Ethernet0/1), d=192.168.200.1 (Ethernet0/0), len 100,
input feature, Policy Routing(68), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.100.1 (Ethernet0/1), d=192.168.200.1 (Ethernet0/0), len 100,
input feature, MCI Check(73), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.100.1, d=192.168.200.1, pak EEB4A9D8 consumed in output feature,
packet consumed, IPSec output classification(30), rtype 2, forus FALSE,
sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, local feature,
Policy Routing(3), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, sending
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
IPSec output classification(30), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
IPSec: to crypto engine(64), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, output feature,
Post-encryption output features(65), rtype 2, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, post-encap
feature, (1), rtype 2, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, post-encap
feature, FastEther Channel(3), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (local), d=10.0.0.2 (Ethernet0/0), len 192, sending full
packet
```

Вот то, что происходит:

Представляющий интерес трафик поражает PBR в E0/0 и инициирует крипто-код для

передачи Пакета ISAKMP. Тот Пакет ISAKMP локально маршрутизируется политикой, и исходящий интерфейс определен локальным PBR. Туннель создан.

Вот еще один эхо-запрос к 192.168.200.1 от R4:

```
R4#ping 192.168.200.1
```

Вот отладки на R1:

```
IP: s=192.168.100.1 (Ethernet0/1), d=192.168.200.1 (Ethernet0/0), len 100,
input feature, Policy Routing(68), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.100.1 (Ethernet0/1), d=192.168.200.1 (Ethernet0/0), len 100,
input feature, MCI Check(73), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.100.1 (Ethernet0/1), d=192.168.200.1 (Ethernet0/0), len 100,
output feature, IPSec output classification(30), rtype 2, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.100.1, d=192.168.200.1, pak EF722068 consumed in output feature,
packet consumed, IPSec: to crypto engine(64), rtype 2, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, input
feature, Policy Routing(68), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, input
feature, MCI Check(73), rtype 2, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, output
feature, IPSec output classification(30), rtype 2, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, output
feature, IPSec: to crypto engine(64), rtype 2, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, output
feature, Post-encryption output features(65), rtype 2, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), g=192.168.0.2, len
172, forward
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, post-encap
feature, (1), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172, post-encap
feature, FastEther Channel(3), rtype 0, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
IP: s=192.168.0.1 (Ethernet0/1), d=10.0.0.2 (Ethernet0/0), len 172,
sending full packet
```

Вот то, что происходит:

PBR соответствий представляющего интерес трафика на E0/0 и тот PBR определяют исходящий интерфейс (E0/0). На E0/0 пакет используется IPSec и инкапсулируется. После того, как инкапсулированный пакет проверен против того же правила PBR, и исходящий интерфейс определен, пакет передан и получен правильно.

Сводка для различий в поведении

Для локально формируемого трафика исходящий интерфейс для неинкапсулированного трафика (ISAKMP) определен локальным PBR. Для локально формируемого трафика исходящий интерфейс для постинкапсулированного трафика (ESP) определен таблицами маршрутизации (локальный PBR не проверен). Для транзитного трафика исходящий

интерфейс для постинкапсулированного трафика (ESP) определен интерфейсным PBR (дважды, прежде и после инкапсуляции).

Пример конфигурации

Это - практический пример конфигурации, который представляет проблемы, с которыми вы могли бы столкнуться с PBR и локальным PBR с VPN. R2 (CE) имеет два канала поставщика. Маршрутизатор R6 также имеет CE и один канал поставщика. Первая ссылка от R2 до R3 используется в качестве маршрута по умолчанию для R2. Вторая ссылка на R4 используется только для трафика VPN к R6. В случае любого сбоя канала поставщика трафик перенаправлен к другой ссылке.

Топология

!--- конфигурацию

Трафик между 192.168.1.0/24 и 192.168.2.0/24 защищен. Протокол OSPF используется в интернет-облаке для объявления адресов 10.0.0.0/8, которые рассматриваются как общие адреса, назначенные интернет-провайдером на клиента. В реальных условиях BGP используется вместо OSPF.

Конфигурация на R2 и R6 основывается на криптокарте. На R2 PBR используется на E0/0 для направления трафика VPN к R4, если это подключено UP:

```
route-map PBR permit 10
 match ip address cmap
 set ip next-hop verify-availability 10.0.2.4 1 track 20

ip access-list extended cmap
 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255

crypto map cmap 10 ipsec-isakmp
 set peer 10.0.4.6
 set transform-set TS
 match address cmap

interface Ethernet0/0
 ip address 192.168.1.2 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 ip policy route-map PBR
```

Здесь вы видите, что не необходим локальный PBR. Интерфейсный PBR направляет представляющий интерес трафик к 10.0.2.4. Это инициирует крипто-код для инициирования ISAKMP от корректного интерфейса (свяжитесь с R4), даже когда маршрутизация к точкам удаленного узла через R3.

На R6 используются два узла для VPN:

```
crypto map cmap 10 ipsec-isakmp
 set peer 10.0.2.2 !primary
 set peer 10.0.1.2
 set transform-set TS
```

```
 match address cmap
```

R2 использует соглашение об Уровне IP-сервиса (SLA) для прозванивания R3 и R4. Маршрут по умолчанию является R3. В случае сбоя R3 это выбирает R4:

```
ip sla 10
 icmp-echo 10.0.1.3
ip sla schedule 10 life forever start-time now
ip sla 20
 icmp-echo 10.0.2.4
ip sla schedule 20 life forever start-time now

track 10 ip sla 10
track 20 ip sla 20

ip route 0.0.0.0 0.0.0.0 10.0.1.3 track 10
ip route 0.0.0.0 0.0.0.0 10.0.2.4 100
```

Также R2 позволяет доступ в Интернет для всех внутренних пользователей. Для достижения резервирования в случае, где интернет-провайдер к R3 не работает, route-map необходим. Это Преобразования адресов портов (PAT) в трафике к другому исходящему интерфейсу (PAT к E0/1 взаимодействуют, когда R3 подключен UP и точки маршрута по умолчанию к R3 и PAT для взаимодействия через интерфейс E0/2, когда R3 не работает и R4, используется в качестве маршрута по умолчанию).

```
ip access-list extended pat
 deny  ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
 deny  udp any any eq isakmp
 deny  udp any eq isakmp any
 permit ip any any

route-map RMAP2 permit 10
 match ip address pat
 match interface Ethernet0/2
!
route-map RMAP1 permit 10
 match ip address pat
 match interface Ethernet0/1

ip nat inside source route-map RMAP1 interface Ethernet0/1 overload
ip nat inside source route-map RMAP2 interface Ethernet0/2 overload

interface Ethernet0/0
 ip address 192.168.1.2 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 ip policy route-map PBR

interface Ethernet0/1
 ip address 10.0.1.2 255.255.255.0
 ip nat outside
 ip virtual-reassembly in
 crypto map cmap

interface Ethernet0/2
 ip address 10.0.2.2 255.255.255.0
 ip nat outside
 ip virtual-reassembly in
 crypto map cmap
```

Трафик VPN должен быть исключен из трансляции, как делает ISAKMP. Если трафик ISAKMP не исключен из трансляции, это - PATed к внешнему интерфейсу, который идет к R3:

```
R2#show ip nat translation
Pro Inside global     Inside local      Outside local     Outside global
udp 10.0.1.2:500      10.0.2.2:500      10.0.4.6:500      10.0.4.6:500

*Jun 8 09:09:37.779: IP: s=10.0.2.2 (local), d=10.0.4.6, len 196, local
feature, NAT(2), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.2.2 (local), d=10.0.4.6 (Ethernet0/1),
len 196, sending
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, Post-routing NAT Outside(24), rtype 1, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, Common Flow Table(27), rtype 1, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, Stateful Inspection(28), rtype 1, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, IPSec output classification(34), rtype 1, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, NAT ALG proxy(59), rtype 1, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, IPSec: to crypto engine(75), rtype 1, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
output feature, Post-encryption output features(76), rtype 1, forus FALSE, sendself
FALSE, mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
pre-encap feature, IPSec Output Encap(1), rtype 1, forus FALSE, sendself FALSE,
mtu 0, fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
 pre-encap feature, Crypto Engine(3), rtype 1, forus FALSE, sendself FALSE, mtu 0,
fwdchk FALSE
*Jun 8 09:09:37.779: IP: s=10.0.1.2 (local), d=10.0.4.6 (Ethernet0/1), len 196,
sending full packet
```

Тестирование

С этой конфигурацией существует полное резервирование. VPN использует ссылку R4, и остаток трафика маршрутизируется с R3. В случае сбоя R4 трафик VPN установлен со ссылкой R3 (route-map для PBR не совпадает, и маршрутизация по умолчанию используется).

Прежде чем интернет-провайдер к R4 не работает, R6 видит трафик от узла 10.0.2.2:

```
R6#show crypto session
Crypto session current status

Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 10.0.2.2 port 500
 IKEv1 SA: local 10.0.4.6/500 remote 10.0.2.2/500 Active
 IPSEC FLOW: permit ip 192.168.2.0/255.255.255.0 192.168.1.0/255.255.255.0
       Active SAs: 2, origin: crypto map
```

После того, как R2 использует интернет-провайдера для R3 для трафика VPN, R6 видит трафик от узла 10.0.1.2:

```
R6#show crypto session
```

```
Crypto session current status

Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 10.0.1.2 port 500
 IKEv1 SA: local 10.0.4.6/500 remote 10.0.1.2/500 Active
 IPSEC FLOW: permit ip 192.168.2.0/255.255.255.0 192.168.1.0/255.255.255.0
       Active SAs: 2, origin: crypto map
```

Для противоположного сценария, когда ссылка на R3 выключается, все все еще хорошо работает. Трафик VPN все еще использует ссылку на R4. Технология NAT выполнена для 192.168.1.0/24 к PAT для адаптации внешнего адреса. Прежде чем R3 выключается, существует трансляция на 10.0.1.2:

```
R2#show ip nat translations
Pro Inside global     Inside local      Outside local     Outside global
icmp 10.0.1.2:1       192.168.1.1:1     10.0.4.6:1        10.0.4.6:1
```

После того, как R3 выключается, существует все еще старая трансляция наряду с новой трансляцией (к 10.0.2.2), который использует ссылку к R4:

```
R2#show ip nat translations
Pro Inside global     Inside local      Outside local     Outside global
icmp 10.0.2.2:0       192.168.1.1:0     10.0.4.6:0        10.0.4.6:0
icmp 10.0.1.2:1       192.168.1.1:1     10.0.4.6:1        10.0.4.6:1
```

Ловушки

Если все хорошо работает, где ловушки? Они находятся в подробных данных.

Трафик, генерируемый локально

Вот сценарий, который должен инициировать трафик VPN от самого R2. Этот сценарий требует, чтобы вы настроили локальный PBR на R2, чтобы вынудить R2 передать трафик ISAKMP через R4 и заставить туннель восстанавливать работоспособность. Но исходящий интерфейс определен с использованием таблиц маршрутизации с по умолчанию, указывающим на R3, и тот пакет передан к R3 вместо R4, который используется для транзита для VPN. Чтобы проверить, что, войдите:

```
ip access-list extended isakmp
 permit udp any any eq isakmp
 permit udp any eq isakmp any
 permit icmp any any

route-map LOCAL-PBR permit 10
 match ip address isakmp
 set ip next-hop verify-availability 10.0.2.4 1 track 20

ip local policy route-map LOCAL-PBR
```

В данном примере Протокол ICMP, который генерируется локально, вызван через R4. Без этого трафик, генерируемый локально от 192.168.1.2 до 192.168.2.5, обработан с использованием таблиц маршрутизации, и туннель установлен с R3.

Что происходит после применения этой конфигурации? Пакет ICMP от 192.168.1.2 до 192.168.2.5 помещен к R4, и туннель инициируется со ссылкой на R4. Туннель установлен:

```
R2#ping 192.168.2.6 source e0/0 repeat 10
Type escape sequence to abort.
Sending 10, 100-byte ICMP Echos to 192.168.2.6, timeout is 2 seconds:
Packet sent with a source address of 192.168.1.2
.!!!!!!!!!
Success rate is 90 percent (9/10), round-trip min/avg/max = 4/4/5 ms

R2#show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Ethernet0/1
Session status: DOWN
Peer: 10.0.4.6 port 500 fvrf: (none) ivrf: (none)
     Desc: (none)
     Phase1_id: (none)
 IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.2.0/255.255.255.0
       Active SAs: 0, origin: crypto map
       Inbound: #pkts dec"ed 0 drop 0 life (KB/Sec) 0/0
       Outbound: #pkts enc"ed 0 drop 0 life (KB/Sec) 0/0

Interface: Ethernet0/2
Uptime: 00:00:06
Session status: UP-ACTIVE
Peer: 10.0.4.6 port 500 fvrf: (none) ivrf: (none)
     Phase1_id: 10.0.4.6
     Desc: (none)
 IKEv1 SA: local 10.0.2.2/500 remote 10.0.4.6/500 Active
         Capabilities:(none) connid:1009 lifetime:23:59:53
 IKEv1 SA: local 10.0.2.2/500 remote 10.0.4.6/500 Inactive
         Capabilities:(none) connid:1008 lifetime:0
 IPSEC FLOW: permit ip 192.168.1.0/255.255.255.0 192.168.2.0/255.255.255.0
       Active SAs: 2, origin: crypto map
       Inbound: #pkts dec"ed 9 drop 0 life (KB/Sec) 4298956/3593
       Outbound: #pkts enc"ed 9 drop 0 life (KB/Sec) 4298956/3593
```

Все, кажется, работает правильно. Трафик передается с корректным E0/2 ссылки к R4. Даже R6 показывает, что трафик получен от 10.2.2.2, который является IP-адресом ссылки R4:

```
R6#show crypto session detail
Crypto session current status

Code: C - IKE Configuration mode, D - Dead Peer Detection
K - Keepalives, N - NAT-traversal, T - cTCP encapsulation
X - IKE Extended Authentication, F - IKE Fragmentation

Interface: Ethernet0/0
Uptime: 14:50:38
Session status: UP-ACTIVE
Peer: 10.0.2.2 port 500 fvrf: (none) ivrf: (none)
     Phase1_id: 10.0.2.2
     Desc: (none)
 IKEv1 SA: local 10.0.4.6/500 remote 10.0.2.2/500 Active
         Capabilities:(none) connid:1009 lifetime:23:57:13
 IPSEC FLOW: permit ip 192.168.2.0/255.255.255.0 192.168.1.0/255.255.255.0
       Active SAs: 2, origin: crypto map
       Inbound: #pkts dec"ed 1034 drop 0 life (KB/Sec) 4360587/3433
       Outbound: #pkts enc"ed 1029 drop 0 life (KB/Sec) 4360587/3433
```

Но фактически, существует **асимметричная маршрутизация для пакетов ESP** здесь. Пакеты

ESP передаются с 10.0.2.2 как источник, но помещены на ссылку к R3. Зашифрованный ответ возвращен через R4. Это может быть проверено путем проверки счетчиков на R3 и R4:

Счетчики R3 E0/0 прежде, чем передать 100 пакетов:

```
R3#show int e0/0 | i pack
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
    739 packets input, 145041 bytes, 0 no buffer
    0 input packets with dribble condition detected
    1918 packets output, 243709 bytes, 0 underruns
```

И те же счетчики, после передачи 100 пакетов:

```
R3#show int e0/0 | i pack
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
    839 packets input, 163241 bytes, 0 no buffer
    0 input packets with dribble condition detected
    1920 packets output, 243859 bytes, 0 underruns
```

Количество входящих пакетов, увеличенных на 100 (на ссылке к R2), но исходящие пакеты, увеличилось только на 2. Таким образом, R3 только видит зашифрованное эхо - запрос ICMP.

Ответ замечен на R4, прежде, чем передать 100 пакетов:

```
R4#show int e0/0 | i packet
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 1000 bits/sec, 1 packets/sec
    793 packets input, 150793 bytes, 0 no buffer
    0 input packets with dribble condition detected
    1751 packets output, 209111 bytes, 0 underruns
```

После передачи 100 пакетов:

```
R4#show int e0/0 | i packet
 5 minute input rate 0 bits/sec, 0 packets/sec
 5 minute output rate 0 bits/sec, 0 packets/sec
    793 packets input, 150793 bytes, 0 no buffer
    0 input packets with dribble condition detected
    1853 packets output, 227461 bytes, 0 underruns
```

Количество пакетов передало к R2, увеличенному на 102 (зашифрованный Ответ ICMP), в то время как полученные пакеты увеличились на 0. Таким образом, R4 только видит зашифрованный Ответ ICMP. Конечно, захват пакета подтверждает это.

Почему это происходит? Ответ находится в первой части статьи.

Вот поток тех пакетов ICMP:

1. ICMP от 192.168.1.2 до 192.168.2.6 помещен на E0/2 (ссылка к R4) из-за локального PBR.
2. Сеанс ISAKMP создан с 10.0.2.2 и поставивший ссылка E0/2 как ожидалось.
3. Для пакетов ICMP после инкапсуляции маршрутизатор должен определить исходящий интерфейс, который сделан с использованием таблиц маршрутизации та точка к R3. Это - то, почему зашифрованный пакет с источником 10.0.2.2 (ссылка к R4) передается через R3.
4. R6 получает пакет ESP от 10.0.2.2, который совместим с сеансом ISAKMP, дешифрует

пакет и передает ответ ESP на 10.0.2.2.
5. Из-за маршрутизации R5 передает ответ обратно на 10.0.2.2 через R4.
6. R2 получает его и дешифрует, и пакет принят.

Это - то, почему важно быть дополнительно осторожный с локально формируемым трафиком.

Во многих сетях используется Одноадресная пересылка по обратному пути (uRPF), и трафик получен от 10.0.2.2, мог быть отброшен на E0/0 R3. В этом случае эхо-запрос не работает.

Есть ли какое-либо решение для этой проблемы? Возможно вынудить маршрутизатор рассматривать локально формируемый трафик как транзитный трафик. Для этого локальный PBR должен направить трафик к поддельному интерфейсу обратной связи, от которого он маршрутизируется как транзитный трафик.

Это не рекомендуется.

> **Примечание:** Важно быть дополнительно осторожный при использовании NAT наряду с PBR (обратитесь к предыдущему разделу о трафике ISKMP в access-list PAT).

Пример конфигурации без PBR

Существует также другое решение, которое является компромиссом. С той же топологией как предыдущий пример возможно удовлетворить все требования без использования PBR или локального PBR. Для этого сценария только используется маршрутизация. Только еще одна запись маршрутизации прибавляется R2, и все конфигурации PBR PBR / локальные конфигурации PBR удалены:

```
ip route 192.168.2.0 255.255.255.0 10.0.2.4 track 20
```

Всего, R2 имеет эту настройку маршрутизации:

```
ip route 0.0.0.0 0.0.0.0 10.0.1.3 track 10
ip route 0.0.0.0 0.0.0.0 10.0.2.4 100
ip route 192.168.2.0 255.255.255.0 10.0.2.4 track 20
```

Когда ссылка на R3 подключена UP, первая запись маршрутизации является маршрутизацией по умолчанию к R3. Когда ссылка на R3 не работает, вторая запись маршрутизации является резервным маршрутом по умолчанию к R4. Третья запись решает, какой путь трафик к удаленной сети VPN передается, в зависимости от состояния канала R4 (если ссылка R4 подключена UP, трафик к удаленной сети VPN передается через R4). С этой конфигурацией нет никакой потребности в маршрутизации в соответствии с политикой.

Каков недостаток? Нет никакого гранулированного контроля с помощью PBR больше. Не возможно определить адрес источника. В этом случае весь трафик к 192.168.2.0/24 передается к R4, когда это подключено UP, независимо от источника. В предыдущем примере, который управлялся PBR и определенным источником: 192.168.1.0/24 выбран.

Для которого сценария это решение слишком просто? Для сетей нескольких сегментов LAN (позади R2). Когда некоторые из тех сетей должны достигнуть 192.168.2.0/24 безопасным способом (зашифрованные) и другие опасные (дешифрованные) пути, трафик от ненадежных сетей все еще помещен на интерфейс E0/2 R2 и не поражает криптокарту.

Таким образом, это передается дешифрованное через ссылку на R4 (и основное требование должно было использовать R4 только для зашифрованного потока данных).

Этот тип сценария и его требования редки, который является, почему это решение используется справедливо часто.

Сводка

Использование PBR и локальных функций PBR наряду с VPN и NAT могло бы быть сложным и требует глубокого понимания потока пакетов.

Для сценариев, таких как представленные здесь, рекомендуется использовать два отдельных маршрутизатора - каждый маршрутизатор с одним каналом поставщика. В случае сбоя интернет-провайдера трафик может быть перенаправлен легко. Нет никакой потребности в PBR, и общая схема намного более проста.

Существует также компромиссное решение, которое не требует использования PBR, но использует статическое плавание, направляющее вместо этого.

Проверка

В настоящее время для этой конфигурации нет процедуры проверки.

Устранение неполадок

Для этой конфигурации в настоящее время нет сведений об устранении проблем.

Дополнительные сведения

- Cisco Systems – техническая поддержка и документация
- Cisco IOS 15.3 Cisco Systems M&T-