Cisco IOS를 통한 정책 라우팅 및 ESP 및 ISAKMP 패킷에 미치는 영향

목차

소개

이 문서에서는 Cisco IOS®를 사용할 때 ESP(Encapsulating Security Payload) 및 ISAKMP(Internet Security Association and Key Management Protocol) 패킷에 적용되는 PBR(Policy Based Routing) 및 로컬 PBR의 영향에 대해 설명합니다.

기고자: Cisco TAC 엔지니어 Michal Garcarz

사전 요구 사항

요구 사항

Cisco에서는 이러한 주제에 대한 기본적인 지식을 얻을 것을 권장합니다.

- Cisco IOS
- Cisco IOS의 VPN 컨피그레이션

사용되는 구성 요소

이 문서의 정보는 Cisco IOS 버전 15.x를 기반으로 합니다.

이 문서의 정보는 특정 랩 환경의 디바이스를 토대로 작성되었습니다.이 문서에 사용된 모든 디바이스는 초기화된(기본) 컨피그레이션으로 시작되었습니다.현재 네트워크가 작동 중인 경우, 모든 명령어의 잠재적인 영향을 미리 숙지하시기 바랍니다.

배경 정보

IPsec 터널 설정 전에 라우터는 ISAKMP 교환을 시작합니다.이러한 패킷은 라우터에서 생성되므로 패킷은 로컬에서 생성된 트래픽으로 처리되며 로컬 PBR 결정이 적용됩니다.또한 라우터(EIGRP(Enhanced Interior Gateway Routing Protocol), NHRP(Next Hop Resolution Protocol), BGP(Border Gateway Protocol) 또는 ICMP(Internet Control Message Protocol) ping)에서 생성된 패킷은 로컬에서 생성된 트래픽으로 간주되며 로컬 PBR 결정이 적용됩니다.

라우터에서 전달되고 통과 트래픽이라고 하는 터널을 통해 전송되는 트래픽은 로컬에서 생성된 트래픽으로 간주되지 않으며, 원하는 라우팅 정책을 라우터의 인그레스 인터페이스에 적용해야 합니다.

이 트래픽이 터널을 통과하는 트래픽에 미치는 영향은 로컬에서 생성된 트래픽이 PBR을 따르지만 통과 트래픽은 그렇지 않다는 것입니다.이 문서에서는 이러한 행동의 차이에 대한 결과를 설명합니다.

ESP가 캡슐화되어야 하는 트랜짓 트래픽의 경우 PBR이 ESP 캡슐화 전후에 패킷의 이그레스 인터페이스를 결정하므로 라우팅 엔트리를 포함할 필요가 없습니다.ESP가 캡슐화되어야 하는 로컬에서 생성된 트래픽의 경우, 로컬 PBR은 캡슐화하기 전에 패킷에 대해서만 이그레스 인터페이스를 결정하고 라우팅은 캡슐화된 후 패킷에 대한 이그레스 인터페이스를 결정하므로 라우팅 엔트리를 포함해야 합니다.

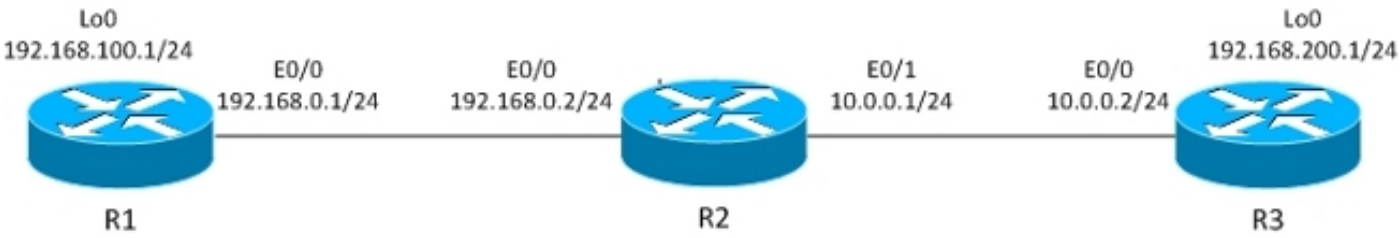
이 문서에는 ISP 링크가 두 개인 라우터 하나가 사용되는 일반적인 컨피그레이션 예가 포함되어 있습니다.인터넷에 액세스하기 위해 하나의 링크가 사용되고, 두 번째 링크는 VPN에 사용됩니다.링크 장애가 발생할 경우 트래픽은 다른 ISP(Internet Service Provider) 링크로 다시 라우팅됩니다.또한 위험이 있습니다.

PBR은 CEF(Cisco Express Forwarding)에서 수행되는 반면 로컬 PBR은 프로세스 스위칭입니다.

라우터에서 로컬로 생성된 트래픽

이 섹션에서는 라우터(R)1에서 시작된 트래픽의 동작에 대해 설명합니다. 이 트래픽은 R1에서 캡슐화된 ESP입니다.

토폴로지



IPsec LAN-to-LAN 터널은 R1과 R3 사이에 구축됩니다.

흥미로운 트래픽은 R1 Lo0(192.168.100.1)과 R3 Lo0(192.168.200.1) 사이입니다.

R3 라우터에는 R2에 대한 기본 경로가 있습니다.

R1에는 라우팅 항목이 없으며, 직접 연결된 네트워크만 있습니다.

구성

R1에는 모든 트래픽에 대한 로컬 PBR이 있습니다.

```
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
```

디버깅

R1에서 로컬로 생성된 모든 트래픽은 UP이면 R2로 전송됩니다.

터널을 가동할 때 무엇이 발생하는지 확인하려면 라우터 자체에서 흥미로운 트래픽을 전송합니다.

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

> **주의:**debug **ip packet** 명령은 많은 양의 디버그를 생성할 수 있으며 CPU 사용량에 큰 영향을 미칩니다.주의해서 사용하십시오.

이 디버그를 사용하면 디버그에 의해 처리되는 트래픽의 양을 제한하기 위해 access-list를 사용할 수도 있습니다.debug **ip packet** 명령은 프로세스 스위치드 트래픽만 표시합니다.

R1의 디버그는 다음과 같습니다.

```
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
```

다음과 같은 상황이 발생합니다.

흥미로운 트래픽(192.168.100.1 > 192.168.200.1)은 로컬 PBR과 일치하며 이그레스 인터페이스(E0/0)가 결정됩니다. 이 작업은 암호화 코드를 트리거하여 ISAKMP를 시작합니다.또한 이 패킷은 이그레스 인터페이스(E0/0)를 결정하는 로컬 PBR에 의해 정책 라우팅됩니다. ISAKMP 트래픽이 전송되고 터널이 협상됩니다.

다시 ping하면 어떻게 됩니까?

```
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)
```

다음과 같은 상황이 발생합니다.

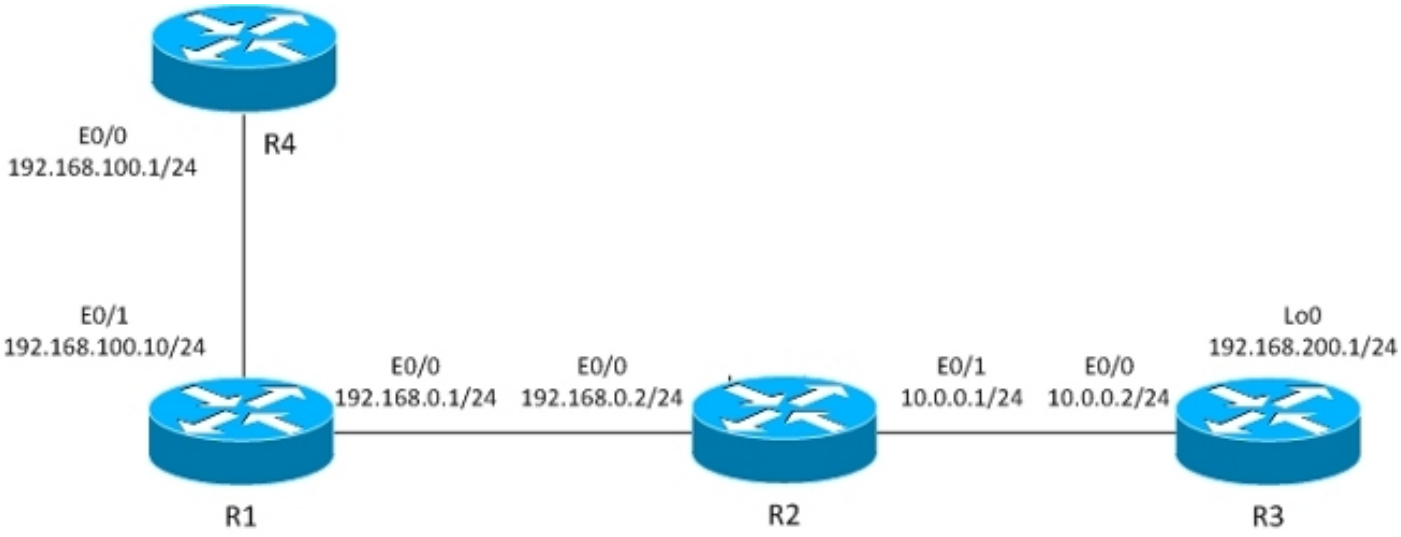
로컬에서 생성된 흥미로운 트래픽 192.168.100.1 > 192.168.200.1은 로컬로 라우팅되며 이그레스 인터페이스가 결정됩니다(E0/0). 패킷은 E0/0의 IPsec 출력 기능에 의해 소비되고 캡슐화됩니다.이그레스 인터페이스를 확인하기 위해 캡슐화된 패킷(192.168.0.1~10.0.0.2)이 라우팅을 위해 확인되지만 R1의 라우팅 테이블에는 아무 것도 없으므로 캡슐화가 실패합니다.

이 시나리오에서는 터널이 UP이지만 트래픽은 전송되지 않습니다. ESP 캡슐화 후 Cisco IOS가 이그레스 인터페이스를 확인하기 위해 라우팅 테이블을 검사하기 때문입니다.

라우터를 통한 전송 트래픽

이 섹션에서는 라우터를 통해 들어오는 통과 트래픽의 동작, 즉 해당 라우터에 의해 캡슐화된 ESP에 대해 설명합니다.

토폴로지



L2L 터널은 R1과 R3 사이에 구축됩니다.

흥미로운 트래픽은 R4(192.168.100.1)과 R3 lo0(192.168.200.1) 사이입니다.

R3 라우터에는 R2에 대한 기본 경로가 있습니다.

R4 라우터에는 R1에 대한 기본 경로가 있습니다.

R1에는 라우팅이 없습니다.

구성

라우터가 암호화를 위해 패킷을 수신하는 경우(로컬에서 생성된 트래픽 대신 트랜짓 트래픽)를 표시하기 위해 이전 토폴로지가 수정됩니다.

현재 R4에서 수신되는 흥미로운 트래픽은 R1에서 정책 라우팅되고(E0/1의 PBR에 의해) 모든 트래픽에 대한 로컬 정책 라우팅도 있습니다.

```
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
```

디버깅

R1에서 터널을 시작할 때(R4에서 흥미로운 트래픽을 수신한 후) 어떤 일이 발생하는지 확인하려면 다음을 입력합니다.

```
R1#debug ip packet

R4#ping 192.168.200.1
```

R1의 디버그는 다음과 같습니다.

```
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
```

다음과 같은 상황이 발생합니다.

흥미로운 트래픽은 E0/0에서 PBR에 도달하고 암호화 코드를 트리거하여 ISAKMP 패킷을 전송합니다.해당 ISAKMP 패킷은 로컬에서 정책 라우팅되며 이그레스 인터페이스는 로컬 PBR에 의해 결정됩니다.터널이 구축됩니다.

다음은 R4에서 192.168.200.1에 대한 ping입니다.

```
R4#ping 192.168.200.1
```

R1의 디버그는 다음과 같습니다.

```
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
```
다음과 같은 상황이 발생합니다.

흥미로운 트래픽은 E0/0에서 PBR에 도달하며, 해당 PBR은 이그레스 인터페이스(E0/0)를 결정합니다. E0/0에서 패킷은 IPSec에서 소비되고 캡슐화됩니다.캡슐화된 패킷이 동일한 PBR 규칙에 대해 확인되고 이그레스 인터페이스가 결정되면 패킷이 올바르게 전송되고 수신됩니다.

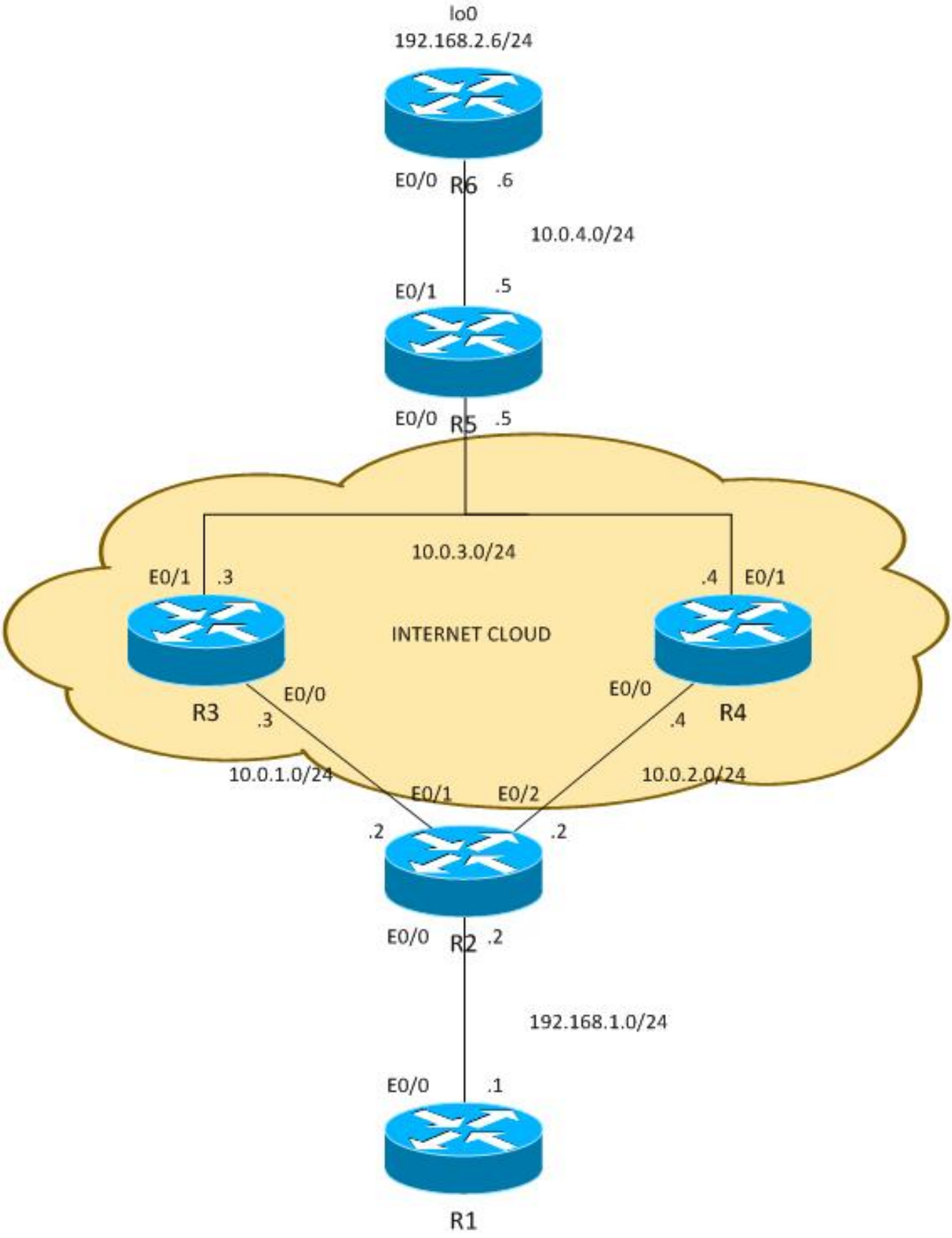
동작 차이점 요약

로컬에서 생성된 트래픽의 경우 ISAKMP(캡슐화되지 않은 트래픽)에 대한 이그레스 인터페이스는 로컬 PBR에 의해 결정됩니다.로컬로 생성된 트래픽의 경우 ESP(post-encapsulated traffic)에 대한 이그레스 인터페이스는 라우팅 테이블에 의해 결정됩니다(로컬 PBR은 선택되지 않음). 통과 트래픽의 경우 ESP(post-encapsulated traffic)에 대한 이그레스 인터페이스는 인터페이스 PBR(두 번, 캡슐화 전후)에 의해 결정됩니다.

컨피그레이션 예

다음은 VPN을 사용하는 PBR 및 로컬 PBR과 관련된 문제를 보여주는 실제 컨피그레이션 예입니다.R2(CE)에는 2개의 ISP 링크가 있습니다.R6 라우터에는 CE 및 ISP 링크가 하나씩 있습니다.R2에서 R3으로 연결되는 첫 번째 링크는 R2의 기본 경로로 사용됩니다. R4로 연결되는 두 번째 링크는 R6로 연결되는 VPN 트래픽에만 사용됩니다. ISP 링크 장애가 발생하면 트래픽이 다른 링크로 다시 라우팅됩니다.

토폴로지



구성

192.168.1.0/24에서 192.168.2.0/24 사이의 트래픽은 보호됩니다.ISP에서 고객에게 할당한 공용 주소로 간주되는 10.0.0.0/8 주소를 광고하기 위해 인터넷 클라우드에서 OSPF(Open Shortest Path

First)가 사용됩니다.실제 환경에서는 OSPF 대신 BGP가 사용됩니다.

R2 및 R6의 컨피그레이션은 암호화 맵을 기반으로 합니다.R2에서 PBR은 E0/0에서 VPN 트래픽이 UP인 경우 R4로 전송하기 위해 사용됩니다.

```
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
```

여기에서는 로컬 PBR이 필요하지 않습니다.인터페이스 PBR은 흥미로운 트래픽을 10.0.2.4으로 라우팅합니다. 이렇게 하면 R3을 통해 원격 피어 포인트에 라우팅하는 경우에도 올바른 인터페이스(R4로 링크)에서 ISAKMP를 시작하도록 암호화 코드가 트리거됩니다.

R6에서는 VPN에 대해 두 개의 피어가 사용됩니다.

```
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
```

R2는 R3 및 R4를 ping하기 위해 IP SLA(서비스 수준 계약)를 사용합니다. 기본 경로는 R3입니다. R3가 실패할 경우 R4를 선택합니다.

```
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
```

또한 R2는 모든 내부 사용자에게 인터넷 액세스를 허용합니다.ISP에서 R3로 연결되는 경우 이중화를 구현하려면 경로 맵이 필요합니다.R3이 UP이고 기본 경로가 R3을 가리키는 경우 E0/1 인터페이스에 대한 트래픽 내부의 PAT(Port Address Translations), R3이 다운되고 R4가 기본 경로로 사용되는 경우 PAT to interface E0/2)

```
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
```

ISAKMP와 마찬가지로 VPN 트래픽도 변환에서 제외해야 합니다.ISAKMP 트래픽이 변환에서 제외되지 않으면 R3로 이동하는 외부 인터페이스에 PAT됩니다.

```
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
```

테스트 중

이 컨피그레이션에서는 완벽한 이중화가 제공됩니다.VPN은 R4 링크를 사용하고 나머지 트래픽은 R3으로 라우팅됩니다. R4 장애가 발생한 경우 VPN 트래픽은 R3 링크로 설정됩니다(PBR의 경로 맵이 일치하지 않고 기본 라우팅이 사용됨).

ISP에서 R4로 향하는 트래픽이 중단되기 전에 R6은 피어 10.0.2.2에서 오는 트래픽을 확인합니다.

```
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
```

R2가 VPN 트래픽에 ISP에서 R3을 사용한 후 R6는 피어 10.0.1.2의 트래픽을 확인합니다.

```
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
```

반대 시나리오에서는 R3에 대한 링크가 중단되더라도 모든 것이 정상적으로 작동합니다.VPN 트래픽은 여전히 R4에 대한 링크를 사용합니다. NAT(Network Address Translation)는 외부 주소를 할당하기 위해 192.168.1.0/24에서 PAT로 수행됩니다.R3가 중단되기 전에 다음 번들로 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
```

R3가 다운된 후에도 R4에 대한 링크를 사용하는 새로운 변환(10.0.2.2)과 함께 기존 번역이 계속 존재합니다.

```
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
```

위험 요소

모든 것이 잘 된다면, 그 함정들은 어디에 있습니까?자세한 내용은

로컬로 생성된 트래픽

다음은 R2 자체에서 VPN 트래픽을 시작해야 하는 시나리오입니다.이 시나리오에서는 R2가 R4를 통해 ISAKMP 트래픽을 전송하고 터널이 UP되도록 하려면 R2에서 로컬 PBR을 구성해야 합니다 .그러나 이그레스 인터페이스는 라우팅 테이블을 사용하여 결정되며, 기본값은 R3을 가리키며, 해당 패킷은 VPN을 위한 전송에 사용되는 R4 대신 R3로 전송됩니다.이를 확인하려면 다음을 입력합니다.

```
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
```

이 예에서 로컬로 생성된 ICMP(Internet Control Message Protocol)는 R4를 통해 강제로 생성됩니다. 이 기능이 없으면 192.168.1.2에서 192.168.2.5으로 로컬로 생성된 트래픽은 라우팅 테이블을 사용하여 처리되고 터널은 R3로 설정됩니다.

이 구성을 적용한 후에는 어떻게 됩니까?192.168.1.2~192.168.2.5의 ICMP 패킷은 R4로 이동하고 R4에 대한 링크를 사용하여 터널이 시작됩니다. 터널은 다음과 같이 설정됩니다.

```
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
```

모든 것이 올바르게 작동하는 것 같습니다.트래픽은 올바른 링크 E0/2를 사용하여 R4로 전송됩니다. R6도 R4의 링크 IP 주소인 10.2.2.2에서 트래픽을 수신함을 보여줍니다.

```
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
```

그러나 실제로 여기는 **ESP 패킷에 대한 비대칭 라우팅이** 있습니다.ESP 패킷은 소스로 10.0.2.2과 함께 전송되지만 R3에 대한 링크에 배치됩니다. 암호화된 응답은 R4를 통해 반환됩니다. 이는 R3 및 R4에서 카운터를 확인하여 확인할 수 있습니다.

100개의 패킷을 전송하기 전에 E0/0의 R3 카운터:

```
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
```

100개의 패킷을 전송한 후에도 동일한 카운터가 있습니다.

```
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
```

수신 패킷 수는 100(R2로 연결되는 링크에서) 증가했지만 나가는 패킷은 2개만 증가했으므로 R3는 암호화된 ICMP 에코만 확인합니다.

100개의 패킷을 전송하기 전에 R4에서 응답을 확인합니다.

```
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
```

100개의 패킷을 전송한 후:

```
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
```

R2로 전송된 패킷 수는 102(암호화된 ICMP 응답)로 증가했으며, 수신된 패킷은 0씩 증가했습니다. 따라서 R4는 암호화된 ICMP 응답만 확인합니다.물론 패킷 캡처가 이를 확인합니다.

왜 이런 일이 발생할까요?그 답은 기사의 첫 부분에 있다.

다음은 이러한 ICMP 패킷의 흐름입니다.

1. 로컬 PBR로 인해 192.168.1.2에서 192.168.2.6으로 ICMP가 E0/2(R4로 연결)에 배치됩니다.
2. ISAKMP 세션은 10.0.2.2으로 구축되며 E0/2 링크를 예상대로 연결합니다.
3. 캡슐화 후 ICMP 패킷의 경우 라우터는 R3를 가리키는 라우팅 테이블을 사용하여 수행되는 이그레스 인터페이스를 확인해야 합니다. 따라서 소스 10.0.2.2(R4로 링크)가 있는 암호화된 패킷이 R3를 통해 전송되는 것입니다.
4. R6은 ISAKMP 세션과 일치하는 10.0.2.2에서 ESP 패킷을 수신하고, 패킷을 해독하고, ESP 응답을 10.0.2.2으로 전송합니다.
5. 라우팅 때문에 R5는 R4를 통해 10.0.2.2에 응답을 다시 전송합니다.
6. R2는 이를 수신하고 해독하며 패킷이 수락됩니다.

따라서 로컬에서 생성된 트래픽에 특히 주의해야 합니다.

많은 네트워크에서 uRPF(Unicast Reverse Path Forwarding)가 사용되고 R3의 E0/0에서 10.0.2.2에서 소싱된 트래픽이 삭제될 수 있습니다. 이 경우 ping이 작동하지 않습니다.

이 문제에 대한 해결책이 있습니까?라우터가 로컬에서 생성된 트래픽을 전송 트래픽으로 처리하도록 할 수 있습니다.이를 위해 로컬 PBR은 트래픽을 전송 트래픽처럼 라우팅되는 가짜 루프백 인터페이스로 전달해야 합니다.

이는 권장되지 않습니다.

> **참고:**PBR과 함께 NAT를 사용할 때는 특히 주의해야 합니다(PAT 액세스 목록의 ISKMP 트래픽에 대한 이전 섹션 참조).

PBR이 없는 컨피그레이션 예

또 다른 해결 방법이 있습니다.이전 예와 동일한 토폴로지를 사용하면 PBR 또는 로컬 PBR을 사용하지 않고도 모든 요구 사항을 충족할 수 있습니다.이 시나리오에서는 라우팅만 사용됩니다.R2에 하나 이상의 라우팅 항목만 추가되고 모든 PBR/로컬 PBR 컨피그레이션이 제거됩니다.

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

총 R2에는 다음과 같은 라우팅 구성이 있습니다.

```
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
```

첫 번째 라우팅 항목은 R3에 대한 링크가 UP인 경우 R3에 대한 기본 라우팅입니다.두 번째 라우팅 항목은 R3에 대한 링크가 다운된 경우 R4에 대한 백업 기본 경로입니다.세 번째 항목은 R4 링크 상태에 따라 원격 VPN 네트워크로 가는 트래픽이 전송되는 방법을 결정합니다(R4 링크가 UP인 경우 원격 VPN 네트워크로 가는 트래픽은 R4를 통해 전송됩니다). 이 컨피그레이션에서는 정책 라우팅이 필요하지 않습니다.

결점은 무엇입니까?더 이상 PBR을 사용하여 세분화된 제어가 없습니다.소스 주소를 확인할 수 없습니다.이 경우 소스와 상관없이 192.168.2.0/24에 대한 모든 트래픽은 UP일 때 R4로 전송됩니다.이전 예에서는 PBR과 특정 소스에 의해 제어되었습니다.192.168.1.0/24이 선택됩니다.

어떤 시나리오에서 이 솔루션이 너무 간단합니까?여러 LAN 네트워크(R2 뒷면)의 경우 이러한 네트워크 중 일부가 안전한 방식으로(암호화) 또는 다른 안전하지 않은 방식으로 192.168.2.0/24에 도달해야 하는 경우(암호화되지 않음) 안전하지 않은 네트워크의 트래픽은 여전히 R2의 E0/2 인터페이스에 있으며 crypto-map에 도달하지 않습니다.따라서 R4로 연결되는 링크를 통해 암호화되지 않은 상태로 전송됩니다. 이때 기본 요구 사항은 암호화된 트래픽에만 R4를 사용하는 것이었습니다.

이러한 시나리오와 요구 사항은 드물기 때문에 이 솔루션이 자주 사용됩니다.

요약

VPN 및 NAT와 함께 PBR 및 로컬 PBR 기능을 사용하는 것은 복잡할 수 있으며 패킷 플로우에 대한 심층적인 이해가 필요합니다.

여기에 제시된 것과 같은 시나리오에서는 두 개의 개별 라우터를 사용하는 것이 좋습니다. 각 라우터에는 ISP 링크가 하나씩 있습니다.ISP 장애가 발생할 경우 트래픽을 쉽게 재라우팅할 수 있습니다.PBR은 필요하지 않으며 전체 설계가 훨씬 간단합니다.

PBR을 사용할 필요는 없지만 대신 고정 부동 라우팅을 사용하는 보안 침해 솔루션도 있습니다.

다음을 확인합니다.

현재 이 구성에 대해 사용 가능한 확인 절차가 없습니다.

문제 해결

현재 이 컨피그레이션에 사용할 수 있는 특정 문제 해결 정보가 없습니다.

관련 정보

- 기술 지원 및 문서 – Cisco Systems
- Cisco IOS 15.3 M&T- Cisco Systems