Cisco IOS策略路由及其對ESP和ISAKMP資料包的影響

目錄

簡介

本檔案介紹使用Cisco IOS®時，原則型路由(PBR)和本地PBR應用於封裝安全負載(ESP)和網際網路安全關聯和金鑰管理通訊協定(ISAKMP)封包時的效果。

作者：Michal Garcarz，思科TAC工程師。

必要條件

需求

思科建議您瞭解以下主題的基本知識：

- Cisco IOS
- Cisco IOS上的VPN配置

採用元件

本檔案中的資訊是根據Cisco IOS版本15.x。

本文中的資訊是根據特定實驗室環境內的裝置所建立。文中使用到的所有裝置皆從已清除（預設）的組態來啟動。如果您的網路正在作用，請確保您已瞭解任何指令可能造成的影響。

背景資訊

建立IPsec隧道之前，路由器會發起ISAKMP交換。由於這些資料包是由路由器生成的，因此這些資料包將被視為本地生成的流量，並應用任何本地PBR決策。此外，由路由器(增強型內部閘道路由通訊協定(EIGRP)、下一個躍點解析通訊協定(NHRP)、邊界閘道通訊協定(BGP)或網際網路控制訊息通訊協定(ICMP)Ping)產生的任何封包也會視為本地產生的流量，並套用本地PBR決策。

由路由器轉發並通過隧道傳送的流量（稱為中轉流量）不視為本地生成的流量，任何所需的路由策略都必須應用到路由器的輸入介面。

這對於通過隧道的流量具有的影響是，本地生成的流量遵循PBR，但傳輸流量不遵循。本文解釋這種行為差異的後果。

對於需要ESP封裝的傳輸流量，不需要任何路由條目，因為PBR在ESP封裝之前和之後確定資料包的出口介面。對於需要進行ESP封裝的本地生成的流量，有必要設定路由條目，因為本地PBR僅確定封裝前資料包的出口介面，而路由確定封裝後資料包的出口介面。

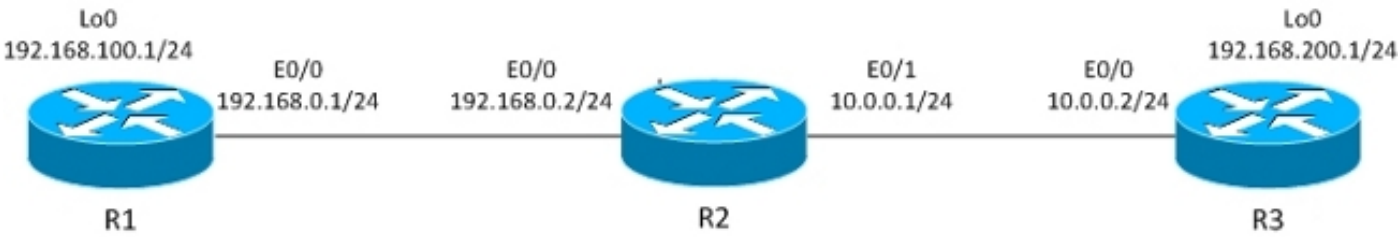
本文檔包含使用具有兩個ISP鏈路的路由器的典型配置示例。一個鏈路用於訪問Internet，另一個鏈路用於VPN。如果出現任何鏈路故障，流量會使用不同的網際網路服務提供商(ISP)鏈路重新路由。同時也存在一些缺陷。

請注意，PBR是在Cisco Express Forwarding(CEF)中執行的，而本地PBR是進程交換的。

路由器本地生成的流量

本節介紹從路由器(R)1發起的流量的行為。該流量是由R1封裝的ESP。

拓撲



IPsec LAN到LAN隧道在R1和R3之間構建。

相關流量在R1 Lo0(192.168.100.1)和R3 Lo0(192.168.200.1)之間。

R3路由器具有通往R2的預設路由。

R1沒有路由條目，只有直連網路。

組態

R1具有所有流量的本地PBR:

```
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
```

調試

當R1為UP狀態時，它會將本地生成的所有流量傳送到R2。

若要確認開啟通道時發生的情況，請從路由器本身傳送相關流量：

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

> **注意**：debug ip packet指令可能會產生大量偵錯，且對CPU使用率有巨大影響。請謹慎使用。

此偵錯也允許使用存取清單，以限制偵錯處理的流量量。debug ip packet命令僅顯示進行進程交換的流量。

以下是R1上的調試：

```
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
```

以下是發生的情況：

感興趣的流量(192.168.100.1 > 192.168.200.1)與本地PBR匹配，並且確定輸出介面(E0/0)。 此操作將觸發加密代碼以啟動ISAKMP。該資料包還由本地PBR進行策略路由，用於確定輸出介面(E0/0)。 傳送ISAKMP流量，並協商隧道

再次執行ping操作時會發生什麼情況？

```
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)
```

以下是發生的情況：

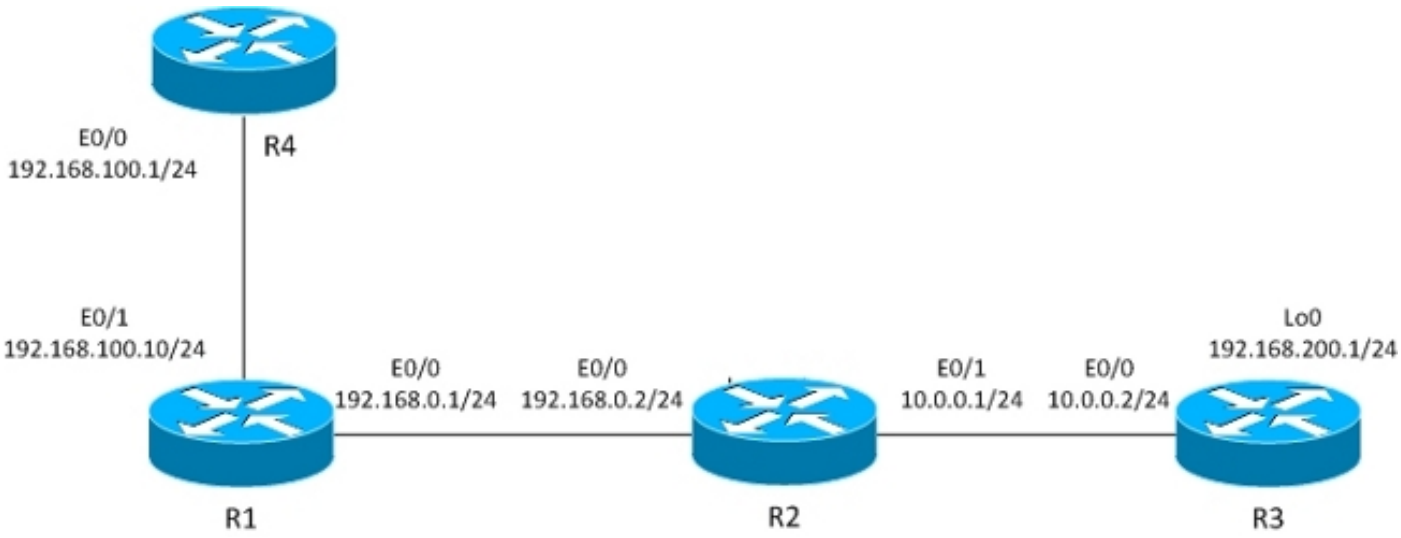
本地生成的關注流量192.168.100.1 > 192.168.200.1是本地策略路由流量，並且輸出介面被確定(E0/0)。 資料包由E0/0上的IPsec輸出功能佔用並封裝。檢查封裝的包（從192.168.0.1到10.0.0.2）的路由以確定出口介面，但R1的路由表中沒有任何內容，因此封裝失敗。

在此案例中，通道為UP，但流量不會傳送，因為在ESP封裝後，Cisco IOS會檢查路由表以確定輸出介面。

通過路由器傳輸流量

本節介紹通過路由器傳輸流量（路由器封裝的ESP）的行為。

拓撲



L2L隧道在R1和R3之間構建。

相關流量在R4(192.168.100.1)和R3 lo0(192.168.200.1)之間。

R3路由器具有通往R2的預設路由。

R4路由器具有通往R1的預設路由。

R1沒有路由。

組態

修改先前的拓撲是為了在路由器收到要加密的資料包（傳輸流量而不是本地生成的流量）時顯示流。

現在，從R4收到的相關流量在R1上通過策略路由（在E0/1上通過PBR），並且所有流量也有本地策略路由：

```
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
```

調試

若要確認在R1上開啟通道時發生的情況（在接收到來自R4的相關流量後），請輸入：

```
R1#debug ip packet

R4#ping 192.168.200.1
```

以下是R1上的調試：

```
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
以下是發生的情況：

相關流量在E0/0上命中PBR，並觸發加密代碼以傳送ISAKMP資料包。該ISAKMP資料包在本地策略路由，輸出介面由本地PBR確定。隧道已建立。

以下是從R4對192.168.200.1執行的更多ping:

```
R4#ping 192.168.200.1
```
以下是R1上的調試：

```
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
```
以下是發生的情況：

相關流量會命中E0/0上的PBR，而PBR會決定輸出介面(E0/0)。 在E0/0上，資料包由IPSec佔用並封裝。根據同一PBR規則檢查封裝的資料包並確定輸出介面後，將正確傳送和接收資料包。

行為差異摘要

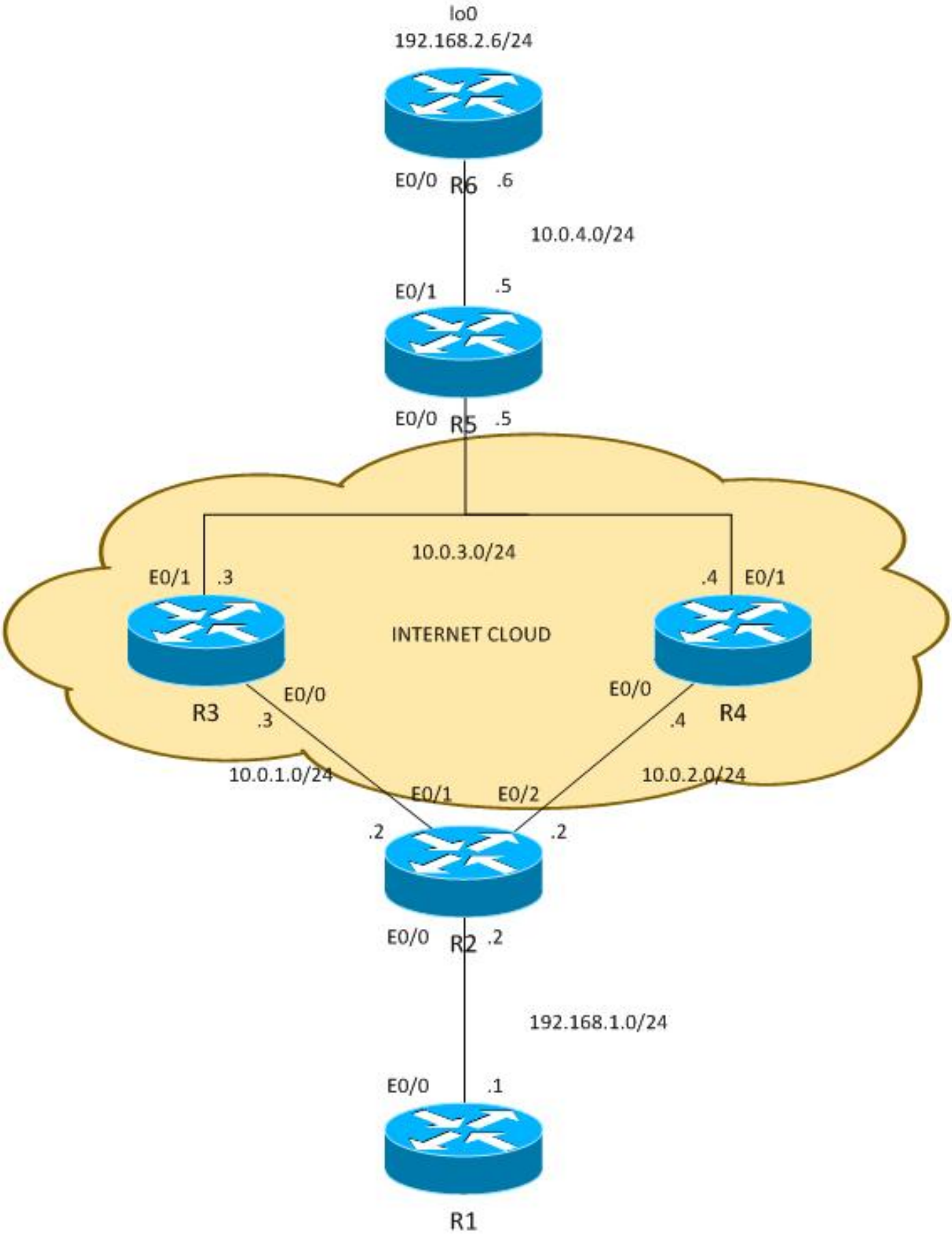
對於本地生成的流量，非封裝流量的輸出介面(ISAKMP)由本地PBR確定。對於本地生成的流量，後

封裝流量(ESP)的出口介面由路由表確定（未檢查本地PBR）。 對於傳輸流量，後封裝流量(ESP)的出口介面由介面PBR（封裝前和封裝後兩次）確定。

組態範例

這是一個實際配置示例，其中介紹了使用PBR時可能遇到的問題，以及使用VPN時本地PBR可能遇到的問題。R2(CE)有兩個ISP鏈路。R6路由器也有CE和一個ISP鏈路。從R2到R3的第一條鏈路用作R2的預設路由。到R4的第二條鏈路僅用於到R6的VPN流量。在任何ISP鏈路發生故障時，流量都會重新路由到另一條鏈路。

拓撲



組態

192.168.1.0/24和192.168.2.0/24之間的流量會受到保護。Internet雲中使用開放最短路徑優先(OSPF)來通告10.0.0.0/8地址，這些地址被視為由ISP分配給客戶的公有地址。在現實世界中，使用

BGP而不是OSPF。

R2和R6上的配置基於加密對映。在R2上，在E0/0上使用PBR，以便在處於UP狀態時將VPN流量定向到R4:

```
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
```

此處您可以看到不需要本地PBR。介面PBR將相關流量路由到10.0.2.4。即使路由是通過R3到達遠端對等點，也會觸發加密代碼，從正確的介面（指向R4的鏈路）發起ISAKMP。

在R6上，使用VPN的兩個對等點：

```
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使用IP服務級別協定(SLA)來ping R3和R4。預設路由是R3。如果R3出現故障，它選擇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
```

此外，R2允許所有內部使用者訪問Internet。為了在ISP與R3之間發生故障時實現冗餘，路由對映是必需的。它將流量內的埠地址轉換(PAT)分配到不同的輸出介面（當R3為UP且預設路由指向R3時，PAT到E0/1介面；當R3為down狀態且R4用作預設路由時，PAT到介面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
```

VPN流量需要從ISAKMP轉換中排除。如果未將ISAKMP流量排除在轉換之外，則會將其通過PAT傳送到指向R3的外部介面：

```
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
```

測試

透過此組態，系統會有完全備援。VPN使用R4鏈路，其餘流量通過R3路由。如果R4出現故障，則使用R3鏈路建立VPN流量（PBR的路由對映不匹配，使用預設路由）。

在ISP到R4關閉之前，R6會看到來自對等體10.0.2.2的流量：

```
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
```

在R2使用ISP到R3的VPN流量後，R6會看到來自對等10.0.1.2的流量：

```
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
```

相反，當到R3的鏈路斷開時，一切仍然正常。VPN流量仍使用到R4的鏈路。對192.168.1.0/24執行PAT網路地址轉換(NAT)，以便分配外部地址。在R3關閉之前，會轉換到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
```

在R3關閉後，仍有舊轉換和新轉換（到10.0.2.2）使用指向R4的連結：

```
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
```

陷阱

如果一切順利，那麼陷阱在哪裡？細節上都有。

本地產生的流量

以下場景需要從R2自身發起VPN流量。此方案要求您在R2上配置本地PBR，以強制R2通過R4傳送ISAKMP流量並使隧道啟動。但是，輸出介面是使用路由表確定的，預設路由表指向R3，資料包傳送到R3，而不是R4，後者用於VPN的傳輸。若要驗證這一點，請輸入：

```
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
```

在本例中，本地產生的網際網路控制訊息通訊協定(ICMP)是透過R4強制產生的。如果沒有該通訊協定，則使用路由表處理本地從192.168.1.2到192.168.2.5產生的流量，並與R3建立通道。

應用此配置後會發生什麼情況？從192.168.1.2到192.168.2.5的ICMP資料包將傳送到R4，並用指向R4的鏈路發起隧道。隧道設定：

```
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
```

一切似乎都能正常運作。流量通過正確的鏈路E0/2傳送到R4。即使R6也顯示從10.2.2.2（R4的鏈路IP地址）接收流量：

```
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
```

但實際上，這裡存在**ESP資料包的非對稱**路由。ESP資料包以10.0.2.2作為源傳送，但被置於通往R3的鏈路上。加密響應通過R4返回。這可以通過檢查R3和R4上的計數器來驗證：

傳送100個資料包之前的E0/0的R3計數器：

```
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
```

和相同的計數器，在傳送100個資料包之後：

```
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
```

傳入資料包的數量增加了100（在通向R2的鏈路上），但傳出資料包僅增加了2。因此，R3隻能看到加密的ICMP回應。

在傳送100個資料包之前，會在R4上看到響應：

```
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
```

傳送100個資料包之後：

```
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
```

向R2傳送的資料包數增加了102（加密的ICMP應答），而收到的資料包數增加了0。因此，R4隻能看到加密的ICMP應答。當然，資料包捕獲可以確認這一點。

為什麼會發生這種情況？答案在文章的第一部分。

以下是這些ICMP封包的流量：

1. 由於本地PBR，從192.168.1.2到192.168.2.6的ICMP被置於E0/2（指向R4的鏈路）上。
2. ISAKMP會話是使用10.0.2.2構建的，並按預期置於E0/2鏈路上。
3. 對於封裝後的ICMP資料包，路由器需要確定輸出介面，這通過使用指向R3的路由表來完成。這就是通過R3傳送源為10.0.2.2（指向R4的鏈路）的加密資料包的原因。
4. R6收到來自10.0.2.2（與ISAKMP會話一致）的ESP資料包，對資料包進行解密，然後將ESP響應傳送到10.0.2.2。
5. 由於路由問題，R5會通過R4將響應傳送回10.0.2.2。
6. R2收到並解密，資料包被接受。

因此，對本地生成的流量格外謹慎非常重要。

在很多網路中，會使用單點傳送反向路徑轉送(uRPF)，而且來源為10.0.2.2的流量可能會在R3的E0/0上捨棄。在這種情況下，ping無法運作。

此問題是否有任何解決方案？可以強制路由器將本地生成的流量視為傳輸流量。為此，本地PBR需要將流量定向到偽環回介面，該介面將像傳輸流量一樣路由到該介面。

不建議這樣做。

> **附註**：將NAT與PBR一起使用時，務必特別小心（請參閱上一節有關PAT訪問清單中的ISKMP流量的部分）。

不帶PBR的配置示例

還有一個解決方案是妥協。使用與前一個示例相同的拓撲，可以不使用PBR或本地PBR來滿足所有要求。在此案例中，只使用路由。在R2上只新增了一個路由條目，並且所有PBR/本地PBR配置都將被刪除：

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

總而言之，R2具有以下路由配置：

```
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
```

通向R3的鏈路為UP時，第一個路由條目是通向R3的預設路由。第二個路由條目是通向R3的鏈路關閉時通向R4的備用預設路由。第三個條目決定向遠端VPN網路傳送流量的方式，這取決於R4鏈路狀態（如果R4鏈路為UP鏈路，則通過R4傳送流向遠端VPN網路的流量）。 透過此設定，不需要原則路由。

缺點是什麼？不再使用PBR進行粒度控制。無法確定源地址。在這種情況下，無論來源如何，通向192.168.2.0/24的所有流量都會在UP時傳送到R4。在上一個示例中，由PBR和特定源控制：已選擇192.168.1.0/24。

對於哪種情況，此解決方案過於簡單？適用於多個LAN網路（在R2之後）。 當其中一些網路需要以安全方式（加密）和其他不安全方式（未加密）到達192.168.2.0/24時，來自不安全網路的流量仍會置於R2的E0/2介面上，並且不會命中crypto-map。因此，它通過到R4的鏈路以未加密的方式傳送（主要要求是僅對加密流量使用R4）。

這種情形及其要求非常罕見，因此經常使用此解決方案。

摘要

將PBR和本地PBR功能與VPN和NAT一起使用可能很複雜，需要深入瞭解資料包流。

對於此處所示的情況，建議使用兩台單獨的路由器 — 每台路由器都包含一個ISP鏈路。在ISP發生故障的情況下，流量很容易重新路由。無需PBR，整體設計更簡單。

還有一個折衷解決方案，它不要求使用PBR，而是使用靜態浮動路由。

驗證

目前沒有適用於此組態的驗證程序。

疑難排解

目前尚無適用於此組態的具體疑難排解資訊。

相關資訊

- 技術支援與文件 - Cisco Systems
- Cisco IOS 15.3 M&T — 思科系統