# 包含第3層和第4層資訊的Nexus 7K上PBR中的ACL行為

## 目錄



## 簡介

本文說明當您根據第3層(L3)和第4層(L4)資訊進行篩選時，Nexus交換器上的原則型路由(PBR)的行為。

## 背景資訊

如果在PBR中新增序列以匹配特定的L4資訊，則由於功能N7K會為訪問控制條目(ACE)建立條目，並且會自動建立與匹配序列中指定的L3資訊匹配的片段ACE。在分段封包的情況下，第一個封包（稱為初始片段）包含L4標頭，並在存取控制清單(ACL)中正確相符。 但是，稱為非初始片段的下一個片段不包含任何L4資訊，因此如果ACL專案的L3部分相符，則允許非初始片段。因此在根據L4資訊過濾流量時，應高度小心，因為如果沒有L4資訊，非初始片段可能會錯誤路由。

## 拓撲

LAN路由器連線到介面E2.1(Vlan 700)上的Nexus。要求是將匹配簡單網路管理協定(SNMP)、Web等的流量重定向到Optimizer和所有其他流量，以便將E2/2介面直接指向防火牆。在Nexus裝置上的交換機虛擬介面(SVI)Vlan700上配置PBR。此處提供了相同內容的配置。路由對映中的序列70將所有其他流量轉發到防火牆。有一個新要求，即具有UDP埠920x的所有流量都需要通過最佳化程式，因為此序列50會新增到路由對映中。

在此處瞭解PBR如何響應在序列50中命中並匹配L3和L4資訊的分段和非分段資料包。

以下是Nexus介面Vlan700上的配置，用於重定向來自E2/1的流量：

```
interface Vlan700

 no shutdown
```

```
 mtu 9000
 vrf member ABC
 no ip redirects
 ip address 10.11.25.25/28
 ip policy route-map In_to_Out


Nexus# show route-map In_to_Out
route-map In_to_Out, permit, sequence 3
 Match clauses:
   ip address (access-lists): Toolbar
 Set clauses:
   ip next-hop 10.3.22.13
route-map In_to_Out, permit, sequence 5
 Match clauses:
   ip address (access-lists): Internet
 Set clauses:
   ip next-hop 10.11.25.19
route-map In_to_Out, permit, sequence 7
 Match clauses:
   ip address (access-lists): Web
 Set clauses:
   ip next-hop 10.11.25.19
route-map In_to_Out, permit, sequence 10
 Match clauses:
   ip address (access-lists): In_to_Out_Internet
 Set clauses:
   ip next-hop 10.11.25.23
route-map In_to_Out, permit, sequence 30
 Match clauses:
   ip address (access-lists): In_to_Out_www
 Set clauses:
   ip next-hop 10.11.25.23
```

```
route-map In_to_Out, permit, sequence 35

 Match clauses:

   ip address (access-lists): In_to_Out_https

 Set clauses:

   ip next-hop 10.11.25.23

route-map In_to_Out, permit, sequence 40

 Match clauses:

   ip address (access-lists): In_to_Out_8080

 Set clauses:

   ip next-hop 10.11.25.23

route-map In_to_Out, permit, sequence 50

 Match clauses:

   ip address (access-lists): UDP_Traffic

 Set clauses:

   ip next-hop 10.11.25.23 >>>>>>>>>>>>>>>>>>> Towards Optimizer

route-map In_to_Out, permit, sequence 70

 Match clauses:

   ip address (access-lists): To_Firewall

 Set clauses:

    ip next-hop . 10.22.45.63 >>>>>>>>>>>>>>>> Towards Firewall



Nexus# show ip access-lists UDP_Traffic

IP access list UDP_Traffic

10 permit udp any any eq 9201
20 permit udp any any eq 9202

30 permit udp any any eq 9203



Nexus# sh ip access-lists To_Firewall

IP access list To_Firewall

       10 permit ip any any
```

在SVI上配置基於策略的路由後，Nexus會在硬體中為此路由建立條目。現在讓我們來看看Nexus模組2上的PBR的硬體程式設計：

```
Nexus# show system internal access-list vlan 700 input entries detail module 2

Flags: F - Fragment entry  E - Port Expansion

      D - DSCP Expansion  M - ACL Expansion

      T - Cross Feature Merge Expansion

INSTANCE 0x0

---------------

 Tcam 1 resource usage:

 ----------------------

 Label_b = 0x201

  Bank 0

  ------

    IPv4 Class

      Policies: PBR(GGSN_Toolbar)

      Netflow profile: 0

      Netflow deny profile: 0

      Entries:

        [Index] Entry [Stats]

        ---------------------

 [0019:000f:000f] prec 1 permit-routed ip 0.0.0.0/0 224.0.0.0/4   [0]

 [002d:0024:0024] prec 1 redirect(0x5d)-routed tcp 1.1.22.80/28 0.0.0.0/0 eq 80 flow-label 80
[0]

 [002e:0025:0025] prec 1 redirect(0x5d)-routed tcp 1.1.22.80/28 0.0.0.0/0 fragment   [0]

 [002f:0026:0026] prec 1 redirect(0x5d)-routed tcp 1.1.22.80/28 0.0.0.0/0 eq 8080 flow-label
8080  [0]

 [0030:0027:0027] prec 1 redirect(0x5d)-routed tcp 1.1.22.80/28 0.0.0.0/0 fragment   [0]

 [0031:0028:0028] prec 1 redirect(0x5d)-routed tcp 1.1.22.48/28 0.0.0.0/0 eq 80 flow-label 80
[0]

 [0032:0029:0029] prec 1 redirect(0x5d)-routed tcp 1.1.22.48/28 0.0.0.0/0 fragment   [0]

 [0033:002a:002a] prec 1 redirect(0x5d)-routed tcp 1.1.22.48/28 0.0.0.0/0 eq 8080 flow-label
8080  [0]

 [0034:002b:002b] prec 1 redirect(0x5d)-routed tcp 1.1.22.48/28 0.0.0.0/0 fragment   [0]

 [0035:002c:002c] prec 1 permit-routed ip 1.1.22.24/29 0.0.0.0/0   [0]

 [0036:002d:002d] prec 1 permit-routed ip 1.1.22.32/28 0.0.0.0/0   [0]

 [0037:002e:002e] prec 1 permit-routed ip 1.1.22.64/28 0.0.0.0/0   [0]
```

```
 [0038:002f:002f] prec 1 permit-routed ip 1.1.22.80/28 0.0.0.0/0   [0]

 [003d:0033:0033] prec 1 permit-routed ip 1.1.22.96/28 0.0.0.0/0   [0]

 [003e:0034:0034] prec 1 permit-routed tcp 0.0.0.0/0 196.11.146.149/32 eq 25 flow-label 25  [0]

 [0059:004f:004f] prec 1 permit-routed tcp 0.0.0.0/0 196.11.146.149/32 fragment   [0]

 [005a:0050:0050] prec 1 redirect(0x5e)-routed ip 1.1.22.16/29 0.0.0.0/0   [0]

 [005b:0051:0051] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 eq 80 flow-label 80  [0]

 [005c:0052:0052] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

 [005d:0053:0053] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 eq 443 flow-label 443
[0]

 [005e:0054:0054] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

 [005f:0055:0055] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 eq 8080 flow-label 8080
[0]

 [0060:0056:0056] prec 1 redirect(0x5e)-routed tcp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

********************Sequence 50 is to match the traffic for UDP ports
9201/9202/9203**********************

 [0061:0057:0057] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9201 flow-label 9201
[0]

 [0062:0058:0058] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

 [0063:0059:0059] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9202 flow-label 9202
[0]

 [0064:005a:005a] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

 [0065:005b:005b] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9203 flow-label 9203
[0]

 [0066:005c:005c] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment   [0]

********************Sequence 70 is to send all other traffic to Firewall**********************

 [0067:005d:005d] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [23]

 [0068:005e:005e] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [0]
```

您會看到，除了與**udp 0.0.0.0/0 0.0.0.0/0 eq 9201匹配的訪問清單條目**外，還有另一個條目與片段**udp 0.0.0.0/0 0.0.0.0/0片段**匹配，但該條目沒有任何UDP埠資訊。此條目等效於與UDP資料包匹配的任何其他條目，因此其他UDP埠的資料包也會按照硬體生成的序列進行匹配。

# 測試案例1:從LAN路由器向防火牆發起的流量

- 到達Nexus的資料包是非分段的，因此流量與PBR中的預期流量匹配。
- 它已被正確重定向到防火牆，可在防火牆上運行的調試中看到。

**UDP packet -port 500**

*Mar 26 04:07:48.959: IP: s=1.1.1.1 (**GigabitEthernet0/0**), d=3.3.3.3, len 28, rcvd 4 -à **Traffic entering from Nexus interface**

*Mar 26 04:07:48.959: UDP src=500, dst=500

**TCP packet - port 80**

*Mar 26 04:07:48.671: IP: s=1.1.1.1 (**GigabitEthernet0/1**), d=3.3.3.3, len 40, rcvd 4 -à **Traffic entering from Optimizer interface**

*Mar 26 04:07:48.671: TCP src=1720, dst=80, seq=0, ack=0, win=0

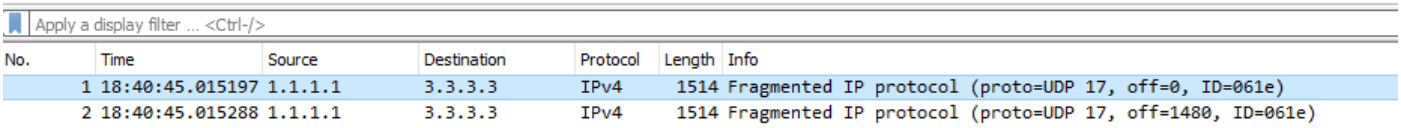
**UDP packet -port 9201**

*Mar 27 09:30:19.879: IP: s=1.1.1.1 (**GigabitEthernet0/1**), d=3.3.3.3, len 28, input feature à **Traffic entering from Optimizer interface**

*Mar 27 09:30:19.879: UDP src=6000, dst=9201, MCI Check(80), rtype 0, forus FALSE, sendself FALSE, mtu 0, fwdchk FALSE

# 測試案例2:通過UDP 500從LAN路由器到防火牆的監聽器檔案發起的流量

此處產生的監聽器檔案中有兩個片段的流量：




1. 使用Route-Map的初始片段：

- **Offset = 0**的第一個片段稱為初始片段，它包含資料包中的UDP報頭。
- 由於流量用於UDP 500，因此它在序列70中匹配以允許**ip any any。**

prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0 [23]

- 因此，同時包含第3層和第4層資訊的第一個資料包會正確路由。

2. 使用Route-Map的非初始分段資料包：

- **Offset ≠ 0**的第二個片段稱為非初始片段，不包含任何UDP報頭。它是協定型別為UDP(17)的純IP資料包。
- 由於沒有第4層資訊，因此它符合序列70 :**允許路由ip 0.0.0.0/0 0.0.0.0/0。**
- 但是在序列50中，存在匹配UDP埠920x的流量的訪問清單。硬體會自動建立一個條目，以允許與指定的第3層資訊匹配的UDP片段。
- 因此，每個分段的資料包都包含序列50中匹配的UDP協定的所有第3層資訊。

prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9201 flow-label 9201 [0]

```
              prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment [23]
>>>>>>>>>>>>>>>>>>>>
```

- 如此一來，就會有一個片段被正確路由，而另一個片段則透過錯誤的順序路由。
- 第二個片段的修改是為了使**offset = 0**，它在序列70中按預期匹配。
- 每當收到第4層片段時，都會發生此預期行為。
- 建立允許片段的額外條目的目的是允許接收到非初始片段，而不需要第4層資訊。
- 在這種情況下，流量用於UDP 9201，沒有允許片段的條目。如此一來，第二個片段就會在序列70中比對以允許**ip any any**，因此會錯誤路由。

```
Nexus# sh route-map In_to_Out pbr-statistics

route-map In_to_Out, permit, sequence 3

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 5

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 7

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 10

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 30

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 35

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 40

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 50 ----------> 2nd Fragment for UDP 500 is matched here

 Policy routing matches: 4397 packets

route-map In_to_Out, permit, sequence 70------------> 1st Fragment for UDP 500 is matched here

 Policy routing matches: 4397 packets
```

- 建立另一個序列45以允許UDP 500的流量，並觀察兩個片段在序列45中是否匹配。
- 由於UDP報頭資訊而匹配的初始片段與序列45的片段行中的非初始片段匹配。

```
Nexus# sh route-map In_to_Out pbr-statistics

route-map In_to_Out, permit, sequence 3

 Policy routing matches: 0 packets
```

```
route-map In_to_Out, permit, sequence 5

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 7

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 10

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 30

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 35

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 40

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 45----------> Both fragments matched here

 Policy routing matches: 213 packets

route-map In_to_Out, permit, sequence 50

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 70

 Policy routing matches: 0 packets


Default routing: 0 packets
```

**序列45的訪問清單：**

```
Nexus# sh ip access-lists udptraffic

IP access list udptraffic

permit udp any any eq isakmp
```

3. 現在，我們來看看fragments關鍵字如何與ACL和Route-Map一起使用

- 應用序列5可允許埠ACL上的任何隨機UDP埠56。

```
          Nexus# sh ip access-lists TEST_UDP


       IP access list TEST_UDP

       statistics per-entry
```

```
        5 permit udp any any eq 56 [match=0]

        10 permit udp any any eq isakmp [match=0]

        20 permit ip any any [match=0]
```

- 已啟動具有分段非初始資料包的流量流，並觀察到它在序列5中匹配。即使該資料包用於UDP 500，它也在序列5中匹配以允許UDP 56。

```
Nexus# sh ip access-lists TEST_UDP



IP access list TEST_UDP

       statistics per-entry

        5 permit udp any any eq 56 [match=56]

        10 permit udp any any eq isakmp [match=0]

        20 permit ip any any [match=0]
```

- 片段ACL會在連線埠ACL上遭到拒絕，而且會發現沒有封包在ACL中針對非初始值進行配對，因為封包確實會在專案**udp any any fragments**中透過平台自動建立相符。

```
NEXUS# sh ip access-lists TEST_UDP



IP access list TEST_UDP

       statistics per-entry

        fragments deny-all

       5 permit udp any any eq 56 [match=0]

        10 permit udp any any eq isakmp [match=0]

        20 permit ip any any [match=0]



 [0014:000a:000a] prec 3 permit udp 0.0.0.0/0 0.0.0.0/0 eq 56 flow-label 56  [0]-> Here we are
now not seeing any entry to allow UDP fragments

 [0015:000b:000b] prec 3 permit udp 0.0.0.0/0 0.0.0.0/0 eq 500 flow-label 500  [0]

 [0016:000c:000c] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0   [0]

 [0017:000d:000d] prec 3 deny ip 0.0.0.0/0 0.0.0.0/0 fragment   [100]>> Getting matched in
fragments deny statement

 [001e:0014:0014] prec 3 deny ip 0.0.0.0/0 0.0.0.0/0   [0]
```

- 已拒絕PBR中成問題ACL中的片段，但此解決方法無效，且序列50和70中的封包仍視為相符。這是由於存取清單和路由對映的程式化行為所致。

```
NEXUS# sh ip access-lists UDP_Traffic

IP access list UDP_Traffic

statistics per-entry

 fragments deny-all

10 permit udp any any eq 9201

 20 permit udp any any eq 9202

 30 permit udp any any eq 9203
```

```
[0061:0057:0057] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9201 flow-label 9201
[0]

[0062:0058:0058] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [8027]

 [0063:0059:0059] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9202 flow-label 9202
[0]

 [0064:005a:005a] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [0]

 [0065:005b:005b] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9203 flow-label 9203
[0]

 [0066:005c:005c] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [0]

  [0067:005d:005d] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [8027]

 [0068:005e:005e] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [0]
```

- 在連線埠ACL和PBR ACL上套用片段deny時的輸出：

```
[0061:0057:0057] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9201 flow-label 9201
[0]

[0062:0058:0058] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [8027] ---
> Once the fragments are denied in port CAL, we observed non-initial packets to be getting
dropped (See the mismatch in number of packets between UDP and IP counter)

 [0063:0059:0059] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9202 flow-label 9202
[0]

 [0064:005a:005a] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [0]

 [0065:005b:005b] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 eq 9203 flow-label 9203
[0]

 [0066:005c:005c] prec 1 redirect(0x5e)-routed udp 0.0.0.0/0 0.0.0.0/0 fragment    [0]

  [0067:005d:005d] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [8214]

 [0068:005e:005e] prec 1 permit-routed ip 0.0.0.0/0 0.0.0.0/0   [0]


VDC-1 Ethernet2/1 :
```

```
 ====================

INSTANCE 0x0

---------------

 Tcam 0 resource usage:

 ----------------------

 Label_a = 0x200

  Bank 0

  ------

    IPv4 Class

      Policies: PACL(TEST_UDP)

       Netflow profile: 0

      Netflow deny profile: 0

      Entries:

         [Index] Entry [Stats]

        ---------------------

 [0014:000a:000a] prec 3 permit udp 0.0.0.0/0 0.0.0.0/0 eq 56 flow-label 56  [8027]

 [0015:000b:000b] prec 3 permit udp 0.0.0.0/0 0.0.0.0/0 eq 500 flow-label 500  [8214]

 [0016:000c:000c] prec 3 permit ip 0.0.0.0/0 0.0.0.0/0   [0]

 [0017:000d:000d] prec 3 deny ip 0.0.0.0/0 0.0.0.0/0 fragment   [100]

 [001e:0014:0014] prec 3 deny ip 0.0.0.0/0 0.0.0.0/0   [0]
```

有幾種可能的方法可以克服具有L4資訊的分段封包的問題或限制：

- 可以調整路由對映，以允許特定UDP埠的特定L3資訊。

在當前配置中，如果提及L3源和目標資訊，則非初始資料包將根據該特定資訊路由。但是，僅當與相同的L3資訊匹配之前沒有其他序列時，該命令才有用。

```
Nexus# show ip access-lists UDP_Traffic

IP access list UDP_Traffic

10 permit udp host 1.1.1.1 host 3.3.3.3 eq 9201
20 permit udp any any eq 9202

30 permit udp any any eq 9203
```

- 可以驗證從來源到目的地的路徑，以便檢查MTU，使封包不會分段。
- 應用另一個序列的變通方法允許有問題的序列上方的UDP工作，但是，該行為與應用序列45時

所解釋的相同

```
Nexus# sh route-map In_to_Out pbr-statistics

route-map In_to_Out, permit, sequence 3

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 5

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 7

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 10

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 30

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 35

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 40

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 45----------> Both fragments matched here

 Policy routing matches: 213 packets

route-map In_to_Out, permit, sequence 50

 Policy routing matches: 0 packets

route-map In_to_Out, permit, sequence 70

 Policy routing matches: 0 packets
```

序列45的訪問清單：

```
Nexus# sh ip access-lists udptraffic
```

IP存取清單更新流量：

```
permit udp any any eq isakmp
```

文檔錯誤：CSCve05428 N7K文檔漏洞 || PBR中同時包含L3和L4資訊的ACL。