Comportamento do ACL no PBR no nexo 7K que contém informação L3 e L4

Índice

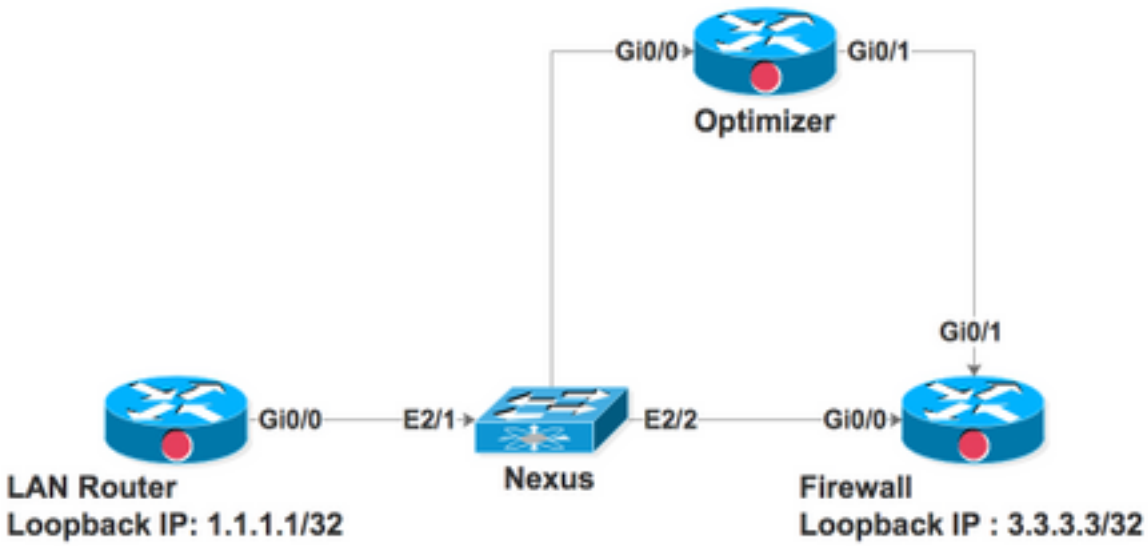
Introdução

Este documento descreve o comportamento do Policy-Based Routing (PBR) no Switches do nexo quando você filtra baseado na camada 3 (L3) e mergulha 4 a informação (L4).

Informações de Apoio

Se você adiciona uma sequência no PBR a fim combinar a informação L4 específica, porque uma característica N7K cria entradas para a entrada de controle de acesso (ACE) e um fragmento ACE é criado automaticamente que combine a informação L3 especificada na sequência do fósforo. Em caso dos pacotes fragmentados, o primeiro pacote conhecido como o fragmento inicial contém o encabeçamento L4 e é combinado corretamente no Access Control List (ACL). Contudo, os fragmentos seguintes conhecidos como fragmentos não iniciais não contêm nenhuma informação L4 e assim se a parcela L3 da entrada ACL combina, o fragmento não inicial é permitido. Tão máximo deve ser tomado, quando filtrar o tráfego baseado na informação L4, como os fragmentos não iniciais pôde errada ser distribuída na ausência da informação L4.

Topologia



O roteador de LAN é conectado ao nexo na relação E2.1, Vlan 700. A exigência é reorientar o tráfego que combina o Simple Network Management Protocol (SNMP), a Web etc. ao Optimizer e o todo tráfego restante diretamente a fim conectar E2/2 para o Firewall. O PBR é configurado na interface virtual do interruptor (SVI) Vlan700 no dispositivo do nexo. A configuração para o mesmos é fornecida aqui. Sequência 70 no mapa de rotas para a frente todo tráfego restante ao Firewall. Há uma exigência nova que todo o tráfego com porta 920x UDP precise de ir através do Optimizer, porque estes 50 pés da sequência são adicionados no mapa de rotas.

Veja aqui como o PBR responde os pacotes fragmentados e NON-fragmentados que batem em ordem 50 pés e combinam a informação L3 e L4.

Está aqui a configuração na relação Vlan700 do nexo para reorientar o tráfego que vem em E2/1:

```
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
```

Uma vez que o roteamento baseado política é configurado no SVI, o nexo cria uma entrada no hardware para o mesmos. Deixa agora o olhar no hardware que programa para o PBR no módulo 2 dos nexos:

```
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]
```

Você vê que além do que a entrada de lista de acesso que combina **UDP 0.0.0.0/0 0.0.0.0/0 eq 9201,** há uma outra entrada que combinem o **UDP 0.0.0.0/0 dos** fragmentos **0.0.0.0/0 fragmentos** mas que a entrada não tem nenhuma informação de porta UDP. Esta entrada é equivalente a qualquer outro que combinar o pacote de UDP, assim que os pacotes para outras portas UDP igualmente obtêm combinados nesta sequência gerada pelo hardware.

Caso de teste 1: Tráfego iniciado do roteador de LAN para o Firewall

- O pacote que alcança o nexo NON-foi fragmentado e daqui o tráfego combinou como esperado no PBR.
- Foi reorientado corretamente ao Firewall e pode ser visto dentro debuga para ser executado no Firewall.

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
```

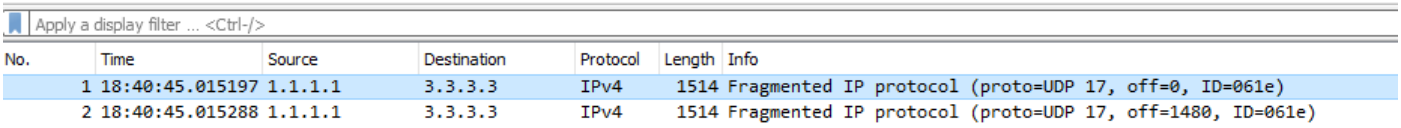
UDP packet -port 9201

*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

Caso de teste 2: Tráfego iniciado através do arquivo de rastreador do roteador de LAN para o Firewall com UDP 500

Tráfego com dois fragmentos no arquivo de rastreador gerado aqui:



1. Fragmentos iniciais com mapa de rotas:

- O primeiro fragmento com **offset = 0** é sabido como o fragmento inicial e contém o cabeçalho de UDP no pacote.
- Enquanto o tráfego é para UDP 500, consegue em ordem 70 combinados permitir **o IP algum.**

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

- O primeiro pacote que tem a informação de ambas camadas 3 e 4 é distribuído assim muito corretamente.

2. Pacotes dos fragmentos não iniciais com mapa de rotas:

- O segundo fragmento com **≠ 0 do offset** é sabido como o fragmento não inicial e não contém

nenhum cabeçalho de UDP. É puramente pacote IP com tipo de protocolo UDP (17).

- Porque não há nenhuma informação da camada 4, combina em ordem 70: **IP licença-roteado 0.0.0.0/0 0.0.0.0/0.**
- Contudo, em ordem 50 pés, há uma lista de acessos que os fósforos trafiquem para a porta 920x UDP. O hardware cria automaticamente uma entrada para permitir os fragmentos UDP que combinam a informação especificada da camada 3.
- Consequentemente, cada pacote fragmentado para alguma informação da camada 3 com protocolo UDP que é em ordem 50 pés combinados.

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
```

- Esta maneira, lá é um fragmento que é distribuído corretamente e outro distribuído através da sequência errada.
- O segundo fragmento é alterado a fim fazer o **offset = 0**, e é combinado em ordem 70 como esperado.
- Este é um comportamento esperado sempre que os fragmentos da camada 4 são recebidos.
- A intenção de criar uma entrada extra para permitir fragmentos é permitir os fragmentos não iniciais recebidos sem informação da camada 4.
- Caso que, o tráfego era para UDP 9201 e não havia nenhuma entrada para permitir fragmentos. Então o segundo fragmento combinaria em ordem 70 para permitir **o IP algum** e daqui ser distribuído errada.

```
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
```

- Uma outra sequência 45 é criada a fim permitir o tráfego para UDP 500 e observar que ambos os fragmentos estão combinados em ordem 45.
- O fragmento inicial combinou devido à informação de cabeçalho de UDP e NON-iniciais combinados nos fragmentos alinham para a sequência 45.

```
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
```

Lista de acessos para a sequência 45:

```
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


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
```

3. Deixa agora para ver como a palavra-chave dos fragmentos se comporta com ACL e mapa de rotas

- A sequência 5 é aplicada para permitir toda a porta aleatória 56 UDP na porta ACL.

```
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
```

- Iniciou um fluxo de tráfego com o pacote NON-inicial fragmentado e observado lhe para combinar em ordem 5. mesmo que o pacote seja para UDP 500, combina em ordem 5 a fim permitir 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]
```

- Os fragmentos são negados na porta ACL e observa-se que nenhum pacote está combinado no ACL para por maisinicial que o pacote obtenha realmente combinado no **UDP da** entrada **todos os quaisquer fragmentos** criados automaticamente pela plataforma.

```
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]
```

- Negou os fragmentos no ACL problemático no PBR, porém esta ação alternativa não trabalhou e os pacotes são vistos ainda para combinar em em ambos os 50 pés e 70 da sequência. Isto é devido ao comportamento de programação da lista de acessos e do mapa de rotas.

```
                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]
```

- As saídas quando os fragmentos negam são aplicadas na porta ACL e no PBR ACL:

```
[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]
```

Há diversos caminhos possíveis superar esta problema ou limitação dos pacotes fragmentados com informação L4:

- O mapa de rotas pode ser tweaked a fim permitir a informação L3 específica para portas particulares UDP.

Na configuração atual, se a informação de origem e de destino L3 é mencionada então o pacote NON-inicial é distribuído com base nessa informação específica. Contudo isto é útil somente quando não há nenhuma outra sequência antes que combine a mesma informação L3.

```
[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]
```

- O trajeto da fonte ao destino pode ser verificado a fim verificar o MTU de modo que o pacote não obtenha fragmentado.
- A ação alternativa de aplicar uma outra sequência permite que o UDP acima da sequência problemática trabalhe, contudo, o comportamento é mesmo que explicado mais cedo em que a sequência 45 era aplicada

```
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
```

Lista de acessos para a sequência 45:

```
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
```

Lista de acesso IP udptraffic:

```
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
```

Erro Doc: Erro CSCve05428 N7K Doc || ACL no PBR que contém a informação L3 e L4.