Comportement d'ACL dans PBR sur le Nexus 7K contenant les informations L3 et L4

Contenu

[Introduction](#)
[Informations générales](#)
[Topologie](#)
[Cas de test 1 : Le trafic initié du routeur de RÉSEAU LOCAL vers le Pare-feu](#)
[Cas de test 2 : Le trafic initié par l'intermédiaire du fichier de renifleur du routeur de RÉSEAU LOCAL vers le Pare-feu avec l'UDP 500](#)

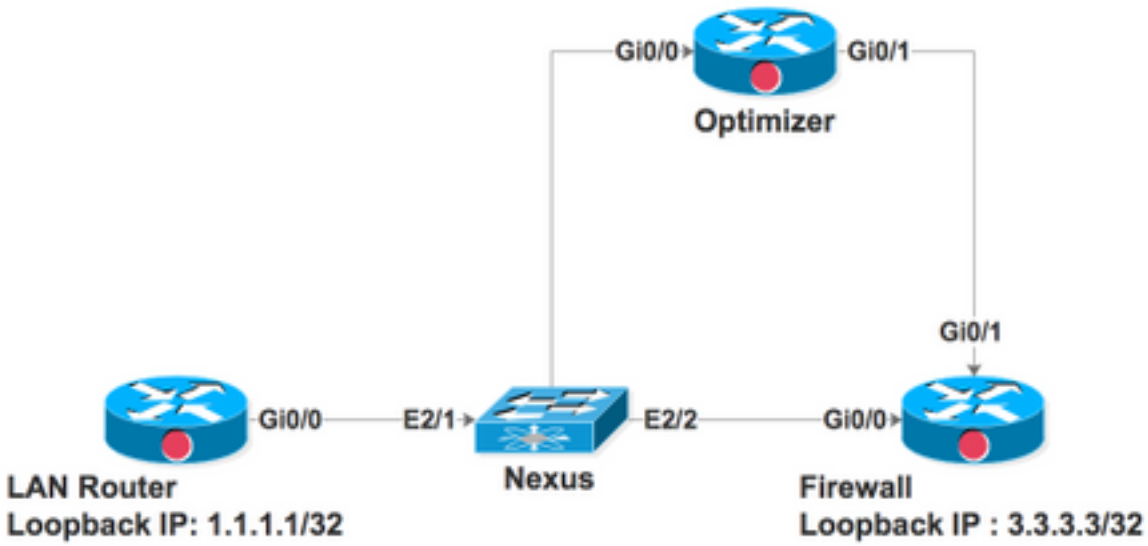
Introduction

Ce document décrit le comportement de la Gestion de réseau à base de règles sur des Commutateurs de Nexus quand vous filtrez basé sur la couche 3 (L3) et posez 4 les informations (L4).

Informations générales

Si vous ajoutez un ordre dans PBR afin d'apparier les informations L4 spécifiques, car une caractéristique N7K crée des entrées pour l'entrée de contrôle d'accès (as) et un fragment ACE est créé automatiquement qui apparie les informations L3 spécifiées dans l'ordre de correspondance. En cas de paquets fragmentés, le premier paquet connu sous le nom de fragment initial contient l'en-tête L4 et est apparié correctement dans la liste de contrôle d'accès (ACL). Cependant, les prochains fragments connus sous le nom de fragments non initiaux ne contiennent aucune informations L4 et ainsi si on permet la partie L3 des correspondances de rubrique de liste ACL, le fragment non initial. Le soin tellement plus grand devrait être pris, alors que le filtrage du trafic basé sur les informations L4, comme fragments non initiaux pourrait être incorrectement conduit faute d'informations L4.

Topologie



Le routeur de RÉSEAU LOCAL est connecté au Nexus sur l'interface E2.1, le VLAN 700. La condition requise est de réorienter le trafic qui apparie le Protocole SNMP (Simple Network Management Protocol), le Web etc. à l'optimiseur et tout l'autre le trafic directement afin de relier E2/2 vers le Pare-feu. PBR est configuré sur Switch Virtual Interface (SVI) Vlan700 sur le périphérique de Nexus. La configuration pour la même chose est fournie ici. Ordre 70 dans le route-map en avant tout l'autre le trafic au Pare-feu. Il y a une nouvelle condition que tout le trafic avec le port UDP 920x doit aller par l'intermédiaire de l'optimiseur, parce que cet ordre 50 est ajouté dans le route-map.

Voyez ici comment PBR répond les paquets fragmentés et Non-fragmentés qui frappent dans l'ordre 50 et apparient les informations L3 et L4.

Voici la configuration sur l'interface Vlan700 de Nexus pour réorienter le trafic qui est livré sur 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
```

Une fois que le routage basé par stratégie est configuré sur le SVI, le Nexus crée une entrée dans le matériel pour la même chose. Permet maintenant de regarder le matériel programmant pour le PBR sur le module 2 du Nexus :

```
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]
```

Vous voyez qu'en plus de l'entrée de liste d'accès qui apparie l'**UDP 0.0.0.0/0 0.0.0.0/0 eq 9201,** il y a une autre entrée qui apparie l'**UDP de** fragments **0.0.0.0/0 0.0.0.0/0 fragment** mais qui l'entrée n'a aucune informations de port UDP. Cette entrée est équivalente à tout autre qui apparie le paquet UDP, ainsi les paquets pour d'autres ports UDP obtiennent également apparié dans cet ordre généré par le matériel.

Cas de test 1 : Le trafic initié du routeur de RÉSEAU LOCAL vers le Pare-feu

- Le paquet qui atteint le Nexus non-a été fragmenté et par conséquent le trafic a été apparié comme prévu dans PBR.
- Il a été réorienté correctement au Pare-feu et peut être vu dedans met au point pour s'exécuter sur le Pare-feu.

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
```

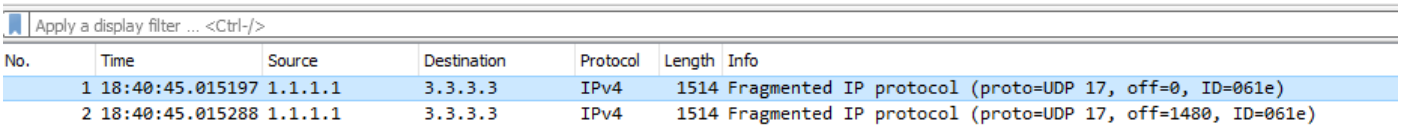
UDP packet -port 9201

```
*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
```

Cas de test 2 : Le trafic initié par l'intermédiaire du fichier de renifleur du routeur de RÉSEAU LOCAL vers le Pare-feu avec l'UDP 500

Le trafic avec deux fragments dans le fichier de renifleur généré ici :



1. Fragments initiaux avec le route-map :

- Le premier fragment avec le **décalage = 0** est connu comme fragment initial et il contient l'en-tête d'UDP dans le paquet.
- Pendant que le trafic est pour l'UDP 500, il obtient dans l'ordre appariés 70 de permettre l'**IP tout.**

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
```

- Ainsi le tout premier paquet qui a le à la fois les informations des couches 3 et 4 est conduit

correctement.

2. Paquets de fragments non initiaux avec le route-map :

- Le deuxième fragment avec le **≠ excentré 0** est connu comme fragment non initial et ne contient pas n'importe quelle en-tête d'UDP. C'est purement paquet IP avec l'UDP de type de protocole (17).
- Car il n'y a aucune informations de la couche 4, elle apparie dans l'ordre 70 : **IP autorisation-conduit 0.0.0.0/0 0.0.0.0/0.**
- Cependant, dans l'ordre 50, il y a une liste d'accès que les correspondances trafiquent pour le port UDP 920x. Le matériel crée automatiquement une entrée pour permettre les fragments d'UDP qui apparient les informations spécifiées de la couche 3.
- Par conséquent, chaque paquet fragmenté pour toutes informations de la couche 3 avec le protocole UDP qui est apparié dans l'ordre 50.

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

- De cette façon, là est un fragment qui est conduit correctement et des autres conduits par l'intermédiaire de l'ordre faux.
- Le deuxième fragment est modifié afin de faire le **décalage = 0**, et il est apparié dans l'ordre 70 comme prévu.
- C'est un comportement prévu toutes les fois que les fragments de la couche 4 sont reçus.
- L'intention de créer une entrée supplémentaire pour permettre des fragments est de permettre les fragments non initiaux reçus sans informations de la couche 4.
- Au cas où, le trafic n'était pour l'UDP 9201 et il y avait aucune entrée pour permettre des fragments. Alors le deuxième fragment aurait apparié dans l'ordre 70 pour permettre à l'**IP tout** et par conséquent pour être conduit incorrectement.

```
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
```

- Un autre ordre 45 est créé afin de permettre le trafic pour l'UDP 500 et observer que chacun des deux les fragments sont appariés dans l'ordre 45.
- Le fragment initial a apparié en raison des informations d'en-tête d'UDP et la non-initiale appariée dans les fragments rayent pour l'ordre 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
```

Liste d'accès pour l'ordre 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. Permet maintenant de voir comment le mot clé de fragments se comporte avec l'ACL et le route-map

- L'ordre 5 est appliqué pour permettre n'importe quel port UDP aléatoire 56 sur l'ACL de port.

```
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
```

- A initié un flux de trafic avec le paquet fragmenté de non-initiale et l'a observé apparier dans l'ordre 5. quoique le paquet soit pour l'UDP 500, il apparie dans l'ordre 5 afin de permettre l'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]
```

- Les fragments sont refusés sur l'ACL de port et on l'observe qu'aucun paquet n'est apparié dans l'ACL pour la non-initiale pendant que le paquet obtient réellement apparié dans l'**UDP d'**entrée **tous les n'importe quels fragments** automatiquement créés par la plate-forme.

```
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]
```

- A refusé les fragments dans l'ACL problématique dans PBR, toutefois ce contournement n'a pas fonctionné et des paquets sont encore vus pour s'assortir dans l'ordre 50 et 70. C'est dû au comportement de programmation de la liste d'accès et du route-map.

```
                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]
```

- Des sorties quand les fragments refusent est appliquées sur l'ACL de port et l'ACL PBR :

```
[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]
```

Il y a plusieurs moyens possibles de surmonter ce problème ou limite de paquets fragmentés avec les informations L4 :

- Le route-map peut être tordu afin de permettre les informations L3 spécifiques pour les ports UDP particuliers.

En configuration en cours, si la source L3 et les informations sur la destination est mentionnée alors le paquet de non-initiale est conduit a basé sur ces informations spécifiques. Cependant c'est utile seulement quand il n'y a aucun autre ordre avant qu'il apparie les mêmes informations 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]
```

- Le chemin de la source à la destination peut être vérifié afin de vérifier le MTU de sorte que le paquet n'obtienne pas fragmenté.
- Le contournement d'appliquer un autre ordre permet à l'UDP au-dessus de l'ordre problématique pour fonctionner, cependant, le comportement correspond expliqué plus tôt où l'ordre 45 était appliqué

```
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
```

Liste d'accès pour l'ordre 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
```

Liste d'accès 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
```

Bogue de documentation : Bogue de documentation CSCve05428 N7K || ACL dans PBR qui contient les informations L3 et L4.