Comportamiento de ACL en PBR en Nexus 7K que contiene información de L3 y L4

Contenido

Introducción

Este documento describe el comportamiento del routing basado en políticas (PBR) en switches Nexus cuando se filtra en función de la información de capa 3 (L3) y capa 4 (L4).

Antecedentes

Si agrega una secuencia en PBR para que coincida con información específica de L4, como una función N7K crea entradas para Access Control Entry (ACE) y se crea automáticamente una ACE de fragmento que coincide con la información de L3 especificada en la secuencia de coincidencia. En el caso de los paquetes fragmentados, el primer paquete conocido como fragmento inicial contiene el encabezado L4 y coincide correctamente en la Lista de control de acceso (ACL). Sin embargo, los fragmentos siguientes conocidos como fragmentos no iniciales no contienen ninguna información L4 y, por lo tanto, si la parte L3 de la entrada ACL coincide, se permite el fragmento no inicial. Por lo tanto, se debe tener sumo cuidado, mientras se filtra el tráfico basado en la información L4, ya que los fragmentos no iniciales podrían enrutarse erróneamente en ausencia de información L4.

Topología

El router LAN está conectado a Nexus en la interfaz E2.1, Vlan 700. El requisito es redirigir el tráfico que coincide con el protocolo simple de administración de red (SNMP), la Web, etc. al Optimizer y a todo el resto del tráfico directamente para interconectar E2/2 hacia el firewall. PBR se configura en la interfaz virtual del switch (SVI) Vlan700 en el dispositivo Nexus. Aquí se proporciona la configuración para la misma. La secuencia 70 en el route-map reenvía el resto del tráfico al firewall. Existe un nuevo requisito de que todo el tráfico con el puerto UDP 920x debe ir a través de Optimizer, ya que esta Secuencia 50 se agrega en el route-map.

Vea aquí cómo PBR responde a los paquetes fragmentados y no fragmentados que llegan en la secuencia 50 y coinciden con la información de L3 y L4.

Esta es la configuración en la interfaz Nexus Vlan700 para redirigir el tráfico que viene en 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
```

Una vez que el ruteo basado en políticas se configura en SVI, Nexus crea una entrada en hardware para el mismo. Veamos ahora la programación de hardware para el PBR en el módulo 2 de 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]
```

Usted ve que además de Access List Entry que coincide con **udp 0.0.0.0/0 0.0.0.0/0 eq 9201,** hay otra entrada que coincide con los fragmentos **udp 0.0.0.0/0 0.0.0.0/0 fragment** pero esa entrada no tiene ninguna información de puerto UDP. Esta entrada es equivalente a cualquier otra que coincida con el paquete UDP, por lo que los paquetes para otros puertos UDP también se igualan

en esta secuencia generada por el hardware.

Caso de prueba 1: Tráfico iniciado desde el router LAN hacia el firewall

- El paquete que llega al Nexus no estaba fragmentado y, por lo tanto, el tráfico coincidió como se esperaba en PBR.
- Se redirigió correctamente al firewall y se puede ver en depuraciones ejecutadas en el 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
```

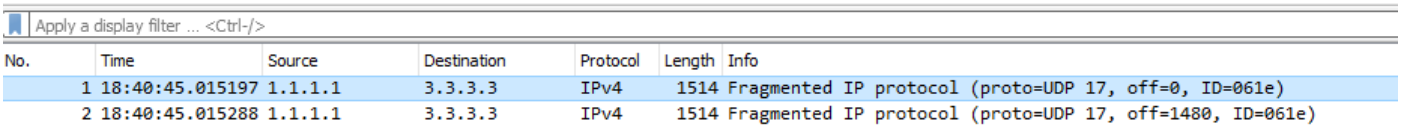
UDP packet -port 9201

```
*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 prueba 2: Tráfico iniciado a través del archivo del sabueso desde el router LAN hacia el firewall con UDP 500

Tráfico con dos fragmentos en el archivo del sabueso generado aquí:



1. Fragmentos iniciales con Route-Map:

- El primer fragmento con **Desplazamiento = 0** se conoce como fragmento inicial y contiene el encabezado UDP en el paquete.
- Como el tráfico es para UDP 500, se corresponde en la secuencia 70 para permitir **ip any**

any.

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

- El primer paquete que tiene información de Capa 3 y Capa 4 se rutea correctamente.

2. Paquetes de Fragmentos no Iniciales con Route-Map:

- El segundo fragmento con **Offset › 0** se conoce como fragmento no inicial y no contiene ningún encabezado UDP. Es un paquete puramente IP con tipo de protocolo UDP (17).
- Como no hay información de Capa 4, coincide en la secuencia 70 : **permit-routing ip 0.0.0.0/0 0.0.0.0/0.**
- Sin embargo, en la secuencia 50, hay una Lista de Acceso que coincide con el tráfico para el puerto UDP 920x. El hardware crea automáticamente una entrada para permitir los fragmentos UDP que coinciden con la información de Capa 3 especificada.
- Por lo tanto, cada paquete fragmentado para cualquier información de Capa 3 con protocolo UDP que coincide en la secuencia 50.

```
            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]
>>>>>>>>>>>>>>>>>>>>
```

- De esta manera, hay un fragmento que se rutea correctamente y otro ruteado a través de una secuencia incorrecta.
- El segundo fragmento se modifica para hacer el **desplazamiento = 0**, y se corresponde en la Secuencia 70 como se esperaba.
- Este es un comportamiento esperado cada vez que se reciben los fragmentos de la Capa 4.
- La intención de crear una entrada adicional para permitir fragmentos es permitir los fragmentos no iniciales recibidos sin información de Capa 4.
- En el caso, el tráfico era para UDP 9201 y no había entrada para permitir fragmentos. Entonces el segundo fragmento habría coincidido en la Secuencia 70 para permitir **ip any any** y por lo tanto se rutearía erróneamente.

```
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
```

- Se crea otra secuencia 45 para permitir el tráfico para UDP 500 y observar que ambos fragmentos se corresponden en la secuencia 45.
- El fragmento inicial coincidió debido a la información de encabezado UDP y a la coincidencia no inicial en la línea de fragmentos para la secuencia 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 acceso para la secuencia 45:

```
Nexus# sh ip access-lists udptraffic

IP access list udptraffic

permit udp any any eq isakmp
```

3. Ahora veamos cómo se comporta la palabra clave fragments con ACL y Route-Map

- La secuencia 5 se aplica para permitir cualquier puerto UDP 56 aleatorio en la ACL del puerto.

```
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]
```

- Inició un flujo de tráfico con un paquete no inicial fragmentado y observó que coincidía en la secuencia 5. Aunque el paquete es para UDP 500, coincide en la secuencia 5 para 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]
```

- Los fragmentos se niegan en la ACL del puerto y se observa que no hay paquetes coincidentes en la ACL para los no iniciales, ya que el paquete realmente se corresponde en

la entrada **udp cualquier fragmento** creado automáticamente por la 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]
```

- Denegados los fragmentos en la ACL problemática en PBR, sin embargo esta solución alternativa no funcionó y se sigue viendo que los paquetes coinciden en la secuencia 50 y 70. Esto se debe al comportamiento de programación de la lista de acceso y del mapa de ruta.

```
          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]
```

- Salidas cuando se aplica la negación de fragmentos en la ACL de puerto y en la ACL de 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]
```

Hay varias maneras posibles de superar este problema o limitación de paquetes fragmentados con información de L4:

- El mapa de ruta se puede ajustar para permitir información específica de L3 para puertos UDP específicos.

En la configuración actual, si se menciona la información de origen y destino de L3, el paquete no inicial se rutea según esa información específica. Sin embargo, esto sólo es útil cuando no hay otra secuencia antes de que coincida con la misma información de L3.

```
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
```

- El trayecto de origen a destino se puede verificar para verificar la MTU de modo que el paquete no se fragmente.
- La solución alternativa de aplicar otra secuencia permite que el UDP por encima de la secuencia problemática funcione, sin embargo, el comportamiento es el mismo que se explicó anteriormente cuando se aplicó la secuencia 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 acceso para la secuencia 45:

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

Lista de acceso IP udptraffic:

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

Error Doc: Error de CSCve05428 N7K Doc || ACL en PBR que contiene información L3 y L4.