Verhalten der ACL im PBR auf dem Nexus 7K, das L3- und L4-Informationen enthält

Inhalt

Einführung

In diesem Dokument wird das Verhalten von Policy-Based Routing (PBR) auf Nexus-Switches beschrieben, wenn Sie nach Informationen zu Layer 3 (L3) und Layer 4 (L4) filtern.

Hintergrundinformationen

Wenn Sie eine Sequenz in PBR hinzufügen, um bestimmte L4-Informationen abzugleichen, erstellt die Funktion N7K Einträge für die Zugriffssteuerungseingabe (ACEs), und ein Fragment-ACE wird automatisch erstellt, das den in der Abgleichsequenz angegebenen L3-Informationen entspricht. Bei fragmentierten Paketen enthält das erste als initiales Fragment bekannte Paket den L4-Header und wird in der Zugriffssteuerungsliste (ACL) korrekt zugeordnet. Die nächsten Fragmente, die als nicht initiale Fragmente bezeichnet werden, enthalten jedoch keine L4-Informationen. Wenn der L3-Teil des ACL-Eintrags also übereinstimmt, ist das nicht initiale Fragment zulässig. Daher sollte bei der Filterung des Datenverkehrs auf der Grundlage von L4-Informationen äußerste Vorsicht geboten werden, da die nicht initialen Fragmente bei Fehlen von L4-Informationen möglicherweise falsch geroutet werden.

Topologie

Der LAN-Router ist über die Schnittstelle E2.1, VLAN 700 mit dem Nexus verbunden. Es ist erforderlich, den Datenverkehr, der mit Simple Network Management Protocol (SNMP), Web usw. übereinstimmt, direkt an Optimizer und den anderen Datenverkehr umzuleiten, um E2/2 mit der Firewall zu verbinden. PBR wird auf dem Switch Virtual Interface (SVI) VLAN700 auf dem Nexus-Gerät konfiguriert. Eine Konfiguration für diese Konfiguration finden Sie hier. Die Sequenz 70 in der Route-Map leitet den gesamten anderen Datenverkehr an die Firewall weiter. Es gibt eine neue Anforderung, dass der gesamte Datenverkehr mit dem UDP-Port 920x über Optimizer erfolgen muss, da diese Sequenz 50 in der route-map hinzugefügt wird.

Hier sehen Sie, wie PBR auf fragmentierte und nicht fragmentierte Pakete reagiert, die in der Folge 50 eintreffen und L3- und L4-Informationen entsprechen.

Die folgende Konfiguration für die Nexus-Schnittstelle Vlan700 dient zur Umleitung des Datenverkehrs, der über E2/1 übertragen wird:

```
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
```

Wenn das richtlinienbasierte Routing auf SVI konfiguriert ist, erstellt Nexus einen Hardwareeintrag für dasselbe. Sehen wir uns jetzt die Hardwareprogrammierung für PBR auf Modul 2 von Nexus an:

```
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]
```

Sie sehen, dass neben dem Access List Entry, der **udp 0.0.0.0/0 0.0.0.0/0 eq 9201** entspricht, ein weiterer Eintrag vorhanden ist, der dem **Fragment 0.0.0.0/0 0.0.0.0/0** entspricht, aber über diesen Eintrag keine UDP-Portinformationen verfügt. Dieser Eintrag entspricht allen anderen, die mit dem

UDP-Paket übereinstimmen. Daher werden die Pakete für andere UDP-Ports in dieser von der Hardware generierten Sequenz abgeglichen.

Testfall 1: Von LAN-Router an Firewall initiierter Datenverkehr

- Das Paket, das den Nexus erreicht, war nicht fragmentiert und der Datenverkehr entsprach daher den Erwartungen des PBR.
- Sie wurde ordnungsgemäß an die Firewall umgeleitet und kann beim Debuggen auf der Firewall angezeigt werden.

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
```

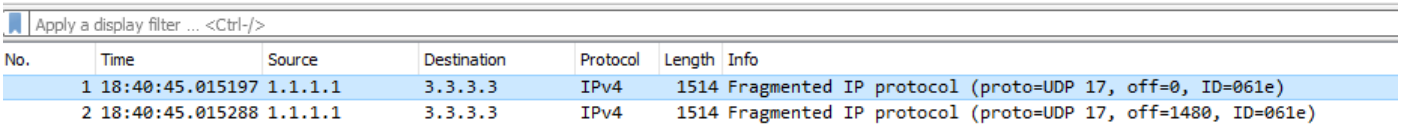
UDP packet -port 9201

```
*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
```

Testfall 2: Über Sniffer-Datei initiierter Datenverkehr vom LAN-Router zur Firewall mit UDP 500

Datenverkehr mit zwei Fragmenten in der hier generierten Sniffer-Datei:



1. Erste Fragmente mit Route Map:

- Das erste Fragment mit **Offset = 0** wird als initiales Fragment bezeichnet und enthält den UDP-Header im Paket.
- Da der Datenverkehr für UDP 500 erfolgt, wird er in Sequenz 70 abgeglichen, um **ip any any**

any any zuzulassen.

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

- Das erste Paket mit Layer-3- und Layer-4-Informationen wird also ordnungsgemäß geroutet.

2. Pakete ohne anfängliche Fragmente mit Route Map:

- Das zweite Fragment mit **Offset ≠ 0** wird als nicht initiales Fragment bezeichnet und enthält keinen UDP-Header. Es handelt sich um ein reines IP-Paket mit Protokolltyp UDP (17).
- Da es keine Layer-4-Informationen gibt, wird sie in der Sequenz 70 übereinstimmen: **permit-routed ip 0.0.0.0/0 0.0.0.0/0.**
- In der Sequenz 50 gibt es jedoch eine Zugriffsliste, die den Datenverkehr für den UDP-Port 920x abgleicht. Die Hardware erstellt automatisch einen Eintrag, um die UDP-Fragmente zuzulassen, die den angegebenen Layer-3-Informationen entsprechen.
- Daher ist jedes fragmentierte Paket für alle Layer-3-Informationen mit UDP-Protokoll, das in der Sequenz 50 zugeordnet ist, fragmentiert.

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

- Auf diese Weise wird ein Fragment ordnungsgemäß geroutet und ein anderes über falsche Sequenz.
- Das zweite Fragment wird geändert, um **Offset = 0** zu erzeugen, und es wird wie erwartet in Sequence 70 zugeordnet.
- Dies ist ein erwartetes Verhalten, wenn die Layer-4-Fragmente empfangen werden.
- Die Absicht, einen zusätzlichen Eintrag zu erstellen, um Fragmente zuzulassen, besteht darin, die nicht initialen Fragmente zu erlauben, die ohne Layer-4-Informationen empfangen werden.
- Für den Fall, dass der Datenverkehr für UDP 9201 erfolgte und es keinen Eintrag gab, um Fragmente zuzulassen. Anschließend hätte das zweite Fragment in Sequence 70 zugeordnet werden können, um **ip beliebige Frames** zuzulassen und somit falsch geroutet zu werden.

```
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
```

- Eine weitere Sequenz 45 wird erstellt, um den Datenverkehr für UDP 500 zuzulassen und zu beobachten, dass beide Fragmente in der Sequenz 45 zugeordnet sind.
- Das ursprüngliche Fragment wurde aufgrund von UDP-Headerinformationen und der nicht initialen Übereinstimmung in der Fragmentzeile für die Sequenz 45 zugeordnet.

```
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
```

Zugriffsliste für Sequenz 45:

```
Nexus# sh ip access-lists udptraffic

IP access list udptraffic

permit udp any any eq isakmp
```

3. Sehen wir uns nun an, wie sich das fragments-Schlüsselwort mit ACL und Route-Map verhält.

- Die Sequenz 5 wird angewendet, um jeden beliebigen UDP-Port 56 auf der Port-ACL zuzulassen.

```
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]
```

- Initiierte einen Datenverkehrsstrom mit fragmentiertem, nicht initialem Paket und stellte fest, dass das Paket in Sequenz 5 übereinstimmt. Obwohl das Paket für UDP 500 gilt, stimmt es mit der Sequenz 5 überein, um UDP 56 zuzulassen.

```
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]
```

- Die Fragmente werden auf der Port-ACL abgelehnt, und es wird beobachtet, dass in der ACL

keine Pakete für nicht initiale Pakete zugeordnet werden, da das Paket im Eintrag **udp** tatsächlich **alle Fragmente** zugeordnet wird, die automatisch von der Plattform erstellt werden.

```
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]
```

- Die Fragmente in der problematischen ACL im PBR wurden abgelehnt, diese Problemumgehung funktionierte jedoch nicht, und es wird immer noch festgestellt, dass die Pakete in den Sequenzen 50 und 70 übereinstimmen. Dies ist auf das Programmierverhalten von Zugriffslisten und Route-Map zurückzuführen.

```
          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]
```

- Gibt aus, wenn Fragmente denk auf Port-ACL und PBR-ACL angewendet werden:

```
[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]
```

Es gibt mehrere Möglichkeiten, dieses Problem zu beheben oder die Beschränkung fragmentierter Pakete mit L4-Informationen zu beschränken:

- Die Route Map kann angepasst werden, um bestimmte L3-Informationen für bestimmte UDP-Ports zuzulassen.

Wenn in der aktuellen Konfiguration L3-Quell- und -Zielinformationen erwähnt werden, wird das nicht initiale Paket basierend auf diesen spezifischen Informationen geroutet. Dies ist jedoch nur dann nützlich, wenn keine andere Sequenz vorhanden ist, bevor sie mit den gleichen L3-Informationen übereinstimmt.

```
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
```

- Der Pfad von der Quelle zum Ziel kann überprüft werden, um die MTU so zu überprüfen, dass das Paket nicht fragmentiert wird.
- Die Problemumgehung beim Anwenden einer anderen Sequenz ermöglicht die Funktionsfähigkeit von UDP über der problematischen Sequenz, das Verhalten ist jedoch identisch mit dem, was bereits zuvor erläutert wurde, als Sequenznummer 45 angewendet wurde

```
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
```

Zugriffsliste für Sequence 45:

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

IP-Zugriffslistenaktualisierungs-Datenverkehr:

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

Doc-Bug: CSCve05428 N7K Doc-Fehler | | ACL im PBR, die L3- und L4-Informationen enthält.