L3とL4の両方の情報を含むNexus 7K上のPBRでのACLの動作

内容

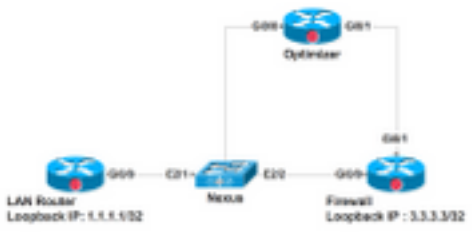
概要

このドキュメントでは、レイヤ3(L3)およびレイヤ4(L4)情報に基づいてフィルタリングを行う場合の、Nexusスイッチでのポリシーベースルーティング(PBR)の動作について説明します。

背景説明

特定のL4情報に一致するようにPBRにシーケンスを追加すると、機能N7Kがアクセスコントロールエントリ(ACE)のエントリを作成し、フラグメントアドレスACEが自動的に作成され、一致シーケンスで指定されたL3情報に一致します。フラグメント化されたパケットの場合、先頭フラグメントと呼ばれる最初のパケットにはL4ヘッダーが含まれ、アクセスコントロールリスト(ACL)で正しく照合されます。 ただし、非先頭フラグメントと呼ばれる次のフラグメントにはL4情報が含まれないため、ACLエントリのL3部分が一致すると、非先頭フラグメントが許可されます。したがって、L4情報がない場合に先頭以外のフラグメントが誤ってルーティングされる可能性があるため、L4情報に基づいてトラフィックをフィルタリングする際は、細心の注意を払う必要があります。

トポロジ



LANルータは、インターフェイスE2.1、Vlan 700のNexusに接続されています。要件は、Simple Network Management Protocol(SNMP)、Webなどに一致するトラフィックをオプティマイザおよびその他すべてのトラフィックに直接リダイレクトし、E2/2をファイアウォールに接続します。PBRは、Nexusデバイスのスイッチ仮想インターフェイス(SVI)Vlan700に設定されます。ここでは、同じ設定を行います。ルートマップのシーケンス70は、他のすべてのトラフィックをファイアウォールに転送します。このシーケンス50がルートマップに追加されるため、UDPポート920xのすべてのトラフィックがオプティマイザを経由する必要があるという新しい要件があります。

シーケンス50でヒットし、L3とL4の両方の情報に一致するフラグメント化パケットと非フラグメント化パケットにPBRが応答する方法を次に示します。

E2/1に着信するトラフィックをリダイレクトするためのNexusインターフェイスVlan700の設定を次に示します。

```
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
```

SVIでポリシーベースルーティングが設定されると、Nexusは同じエントリをハードウェアに作成します。次に、Nexusのモジュール2のPBRのハードウェアプログラミングについて説明します。

```
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]
```

udp 0.0.0.0/0 0.0.0.0/0 eq 9201に一致するアクセスリストエントリに加えて、フラグメントudp 0.0.0.0/0 0.0.0.0/0 fragmentに一致する別のエントリが存在しますが、そのエントリにはUDPポート情報はありません。このエントリは、UDPパケットに一致する他のエントリと同じであるため、他のUDPポートのパケットも、ハードウェアによって生成されるこのシーケンスで一致します

。

テストケース1:LANルータからファイアウォールへのトラフィック開始

- Nexusに到達するパケットはフラグメント化されていないため、PBRで期待どおりにトラフィックが一致しました。
- これはファイアウォールに正しくリダイレクトされ、ファイアウォールで実行されるデバッグで確認できます。

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

テストケース2:LANルータからファイアウォールに向かうスニファファイルを介してUDP 500で開始されるトラフィック

スニファファイルに2つのフラグメントが生成されたトラフィック：

Apply a display filter ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	18:40:45.015197	1.1.1.1	3.3.3.3	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=0, ID=061e)
2	18:40:45.015288	1.1.1.1	3.3.3.3	IPv4	1514	Fragmented IP protocol (proto=UDP 17, off=1480, ID=061e)

1. ルートマップを持つ先頭フラグメント：

- **Offset** = 0の最初のフラグメントは初期フラグメントと呼ばれ、パケット内にUDPヘッダーが含まれます。
- トラフィックはUDP 500用であるため、シーケンス70でip any anyを許可するように**照合さ**

れます。

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

- したがって、レイヤ3とレイヤ4の両方の情報を持つ最初のパケットが正しくルーティングされます。

2. Non-Initial Fragments packets with Route-Map:

- **Offset ≠** 0の2番目のフラグメントは、非先頭フラグメントと呼ばれ、UDPヘッダーを含みません。これは純粋にIPパケットであり、プロトコルタイプはUDP(17)です。
- レイヤ4の情報がないため、シーケンス70(100)で一致します。**permit-routed ip 0.0.0.0/0 0.0.0.0/0**
- ただし、シーケンス50には、UDPポート920xのトラフィックと一致するアクセスリストがあります。ハードウェアは、指定されたレイヤ3情報に一致するUDPフラグメントを許可するためのエントリを自動的に作成します。
- したがって、すべてのフラグメント化されたパケットは、UDPプロトコルを持つレイヤ3情報に対して、シーケンス50で一致します。

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

- このようにして、1つのフラグメントが正しくルーティングされ、別のフラグメントが間違ったシーケンスでルーティングされます。
- 2番目のフラグメントは**offset = 0**に変更さ**れ、**シーケンス70で想定どおりに一致します。
- これは、レイヤ4フラグメントが受信されるたびに予想される動作です。
- フラグメントを許可するための追加エントリを作成する目的は、レイヤ4情報なしで受信された非先頭フラグメントを許可することです。
- この場合、トラフィックはUDP 9201用であり、フラグメントを許可するエントリはありませんでした。次に、2番目のフラグメントはシーケンス70で一致し、**ip any any**を許可し、誤ってルーティングされます。

```
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
```

- UDP 500のトラフィックを許可し、両方のフラグメントがシーケンス45で一致していることを確認するために、別のシーケンス45が作成されます。
- 先頭フラグメントはUDPヘッダー情報に一致し、シーケンス45のフラグメント行で先頭以外が一致しました。

```
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
```

シーケンス45のアクセスリスト：

```
Nexus# sh ip access-lists udptraffic

IP access list udptraffic

permit udp any any eq isakmp
```

3. 次に、fragmentsキーワードがACLおよびルートマップでどのように動作するかを見てみましょう

- ポートACLでランダムなUDPポート56を許可するために、シーケンス5が適用されます。

```
          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]
```

- フラグメント化された非初期パケットを使用してトラフィックストリームを開始し、シーケンス5で一致していることが確認されました。パケットがUDP 500用であっても、UDP 56を許可するためにシーケンス5で一致します。

```
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]
```

- ポートACLではフラグメントが拒否され、非イニシャルのACLではパケットが一致しないことが確認されています。これは、パケットが**udpエントリで実際に一致し、プラットフォーム**

によって自動的に作成されるすべてのフラグメントが一致するからです。

```
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]
```

- PBRで問題のあるACLのフラグメントを拒否しましたが、この回避策は機能せず、パケットがシーケンス50と70の両方で一致することが確認されています。これは、アクセスリストとルートマップのプログラミング動作が原因です。

```
          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]
```

- フラグメント拒否がポートACLとPBR ACLの両方に適用された場合の出力：

```
[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]
```

L4情報を使用したフラグメント化パケットの問題または制限を解決するには、いくつかの方法があります。

- 特定のUDPポートに特定のL3情報を許可するために、ルートマップを微調整できます。

現在の設定では、L3の送信元および宛先情報が指定されている場合、非初期パケットはその特定の情報に基づいてルーティングされます。ただし、これは、同じL3情報に一致する前に他のシーケンスがない場合にのみ便利です。

```
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
```

- パケットがフラグメント化されないように、送信元から宛先へのパスをMTUを確認できます。
- 別のシーケンスを適用する回避策では、問題のあるシーケンスの上のUDPが動作しますが、この動作は、シーケンス45が適用された際の前述の動作と同じです

```
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
```

シーケンス45のアクセスリスト：

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

IPアクセスリストudptraffic:

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

Doc Bug:CSCve05428 N7K Docのバグ || L3とL4の両方の情報を含むPBRのACL。