了解802.1x DACL、每用户ACL、过滤器ID和设备跟踪行为

目录

- 简介
- 设备跟踪理论
- 设备跟踪配置
- 设备跟踪测试
  - 12.2.33版中的调试，DHCP监听更新了IP设备跟踪
  - 探测功能和ARP监听
  - 版本12.2.55的IP设备跟踪 — 隐藏命令
  - 版本12.2.55的IP设备跟踪 — 静态IP示例
  - 版本15.x的IP设备跟踪
  - Cisco IOS-XE®的IP设备跟踪
  - 802.1x的IP设备跟踪和12.2.55版的DACL
  - 802.1x的IP设备跟踪和15.x版的DACL
    - 特定ACL条目
    - 控制方向
  - 802.1x的IP设备跟踪和版本15.x的每用户ACL
    - 与DACL比较时的差异
  - 802.1x的IP设备跟踪和15.x版的Filter-ID ACL
  - IP设备跟踪 — 默认值和最佳实践
- 版本15.x的接口ACL重写
- 用于802.1x的默认ACL
- 打开模式
- 当接口ACL为必填项时
- 4500/6500上的DACL
- 802.1x的MAC地址状态
- 故障排除
- 相关信息

简介

本文档介绍IP设备跟踪功能、添加和删除主机的触发器以及设备跟踪对802.1x DACL的影响。

设备跟踪理论

本文档介绍 IP 设备跟踪功能的工作方式，包括用于添加和移除主机的触发器。

此外，还解释了设备跟踪对802.1x可下载访问控制列表(DACL)的影响。

行为在版本和平台之间更改。

本文档的第二部分重点介绍由身份验证、授权和记帐(AAA)服务器返回并应用于802.1x会话的访问控制列表(ACL)。

DACL、Per-User ACL和Filter-ID ACL之间进行了对比。

此外，还讨论了ACL重写和默认ACL的一些注意事项。

设备跟踪在以下情况下添加条目：

- 它通过DHCP监听获取新条目。
- 它通过地址解析协议(ARP)请求（从ARP数据包读取发送方MAC地址和发送方IP地址）获取新条目。

此功能有时称为ARP检测，但它与动态ARP检测(DAI)不同。

该功能默认启用，不能禁用。它也称为ARP监听，但在启用“debug arp snooping”后，调试不会显示它。

ARP监听默认启用，不能禁用或控制。

设备跟踪会在ARP请求没有响应时删除条目（默认每30秒为设备跟踪表中的每台主机发送一次探测）。

设备跟踪配置

```
ip dhcp excluded-address 192.168.0.1 192.168.0.240
ip dhcp pool POOL
   network 192.168.0.0 255.255.255.0
!
ip dhcp snooping vlan 1
ip dhcp snooping
ip device tracking
!
interface Vlan1
  ip address 192.168.0.2 255.255.255.0
ip route 0.0.0.0 0.0.0.0 10.48.66.1
!
interface FastEthernet0/1
  description PC
```

设备跟踪测试

<#root>

BSNS-3560-1#

show ip dhcp binding

```
IP address       Client-ID/              Lease expiration        Type
                 Hardware address
192.168.0.241    0100.5056.994e.a1       Mar 02 1993 02:31 AM    Automatic

BSNS-3560-1#
```

show ip device tracking all

```
IP Device Tracking = Enabled
---------------------------------------------------------------
  IP Address     MAC Address       Interface          STATE
---------------------------------------------------------------
192.168.0.241   0050.5699.4ea1 FastEthernet0/1        ACTIVE
```

12.2.33版中的调试，DHCP监听更新了IP设备跟踪

DHCP监听填充绑定表：

<#root>

```
BSNS-3560-1#
```

show debugging

```
DHCP Snooping packet debugging is on
DHCP Snooping event debugging is on
DHCP server packet debugging is on.
DHCP server event debugging is on.
track:
  IP device-tracking redundancy events debugging is on
  IP device-tracking cache entry Creation debugging is on
  IP device-tracking cache entry Destroy debugging is on
  IP device-tracking cache events debugging is on

02:30:57: DHCP_SNOOPING: checking expired snoop binding entries
02:31:12: DHCPSNOOP(hlfm_set_if_input): Setting if_input to Fa0/1 for pak.  Was Vl1
02:31:12: DHCPSNOOP(hlfm_set_if_input): Setting if_input to Vl1 for pak.  Was Fa0/1
02:31:12: DHCPSNOOP(hlfm_set_if_input): Setting if_input to Fa0/1 for pak.  Was Vl1
02:31:12:
```

DHCP_SNOOPING: received new DHCP packet from input interface

```
 (FastEthernet0/1)
02:31:12:
```

DHCP_SNOOPING: process new DHCP packet, message type: DHCPREQUEST, input interface: Fa0/1, MAC da: 001f.27e6.cfc0, MAC sa: 0050.5699.4ea1, IP da: 192.168.0.2, IP sa: 192.168.0.241, DHCP ciaddr:

```
 192.168.0.241, DHCP yiaddr: 0.0.0.0,
 DHCP siaddr: 0.0.0.0, DHCP giaddr: 0.0.0.0, DHCP chaddr: 0050.5699.4ea1
02:31:12:
```

DHCP_SNOOPING: add relay information option

```
.
02:31:12: DHCP_SNOOPING_SW: Encoding opt82 CID in vlan-mod-port format
```

```
02:31:12: DHCP_SNOOPING_SW: Encoding opt82 RID in MAC address format
02:31:12: DHCP_SNOOPING: binary dump of relay info option, length: 20 data&colon;
0x52 0x12 0x1 0x6 0x0 0x4 0x0 0x1 0x1 0x3 0x2 0x8 0x0 0x6 0x0 0x1F 0x27 0xE6 0xCF 0x80
02:31:12: DHCP_SNOOPING_SW: bridge packet get invalid mat entry: 001F.27E6.CFC0,
 packet is flooded to ingress VLAN: (1)
02:31:12: DHCP_SNOOPING_SW: bridge packet send packet to cpu port: Vlan1.
02:31:12:

DHCPD: DHCPREQUEST received from client 0100.5056.994e.a1

.
02:31:12:

DHCPD: Sending DHCPACK to client 0100.5056.994e.a1 (192.168.0.241)

.
02:31:12: DHCPD: unicasting BOOTREPLY to client 0050.5699.4ea1 (192.168.0.241).
02:31:12: DHCP_SNOOPING: received new DHCP packet from input interface (Vlan1)
02:31:12:

DHCP_SNOOPING: process new DHCP packet, message type: DHCPACK

, input interface:
 Vl1, MAC da: 0050.5699.4ea1, MAC sa: 001f.27e6.cfc0, IP da: 192.168.0.241,
 IP sa: 192.168.0.2, DHCP ciaddr: 192.168.0.241, DHCP yiaddr: 192.168.0.241,
 DHCP siaddr: 0.0.0.0, DHCP giaddr: 0.0.0.0, DHCP chaddr: 0050.5699.4ea1
02:31:12:

DHCP_SNOOPING: add binding on port FastEthernet0/1

.
02:31:12: DHCP_SNOOPING: added entry to table (index 189)
02:31:12: DHCP_SNOOPING: dump binding entry: Mac=00:50:56:99:4E:A1 Ip=192.168.0.241
 Lease=86400     ld Type=dhcp-snooping Vlan=1 If=FastEthernet0/1
```

将DHCP绑定添加到数据库后，它会触发设备跟踪通知：

<#root>

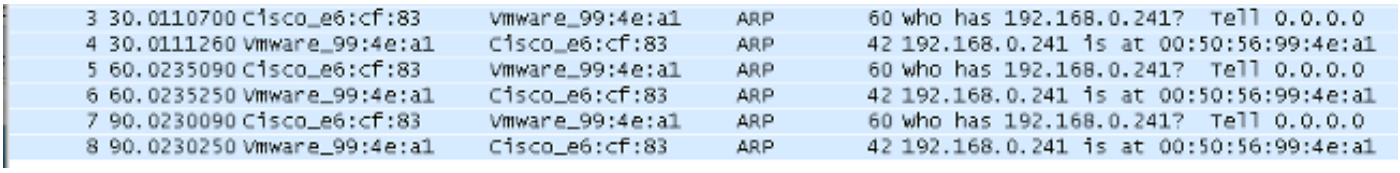
```
02:31:12:

sw_host_track-ev:host_track_notification: Add event for host 0050.5699.4ea1,
 192.168.0.241 on interface FastEthernet0/1


02:31:12: sw_host_track-ev:Async Add event for host 0050.5699.4ea1, 192.168.0.241
 on interface FastEthernet0/1
02:31:12: sw_host_track-ev:MSG = 2
02:31:12: DHCP_SNOOPING_SW no entry found for 0050.5699.4ea1 0.0.0.1 FastEthernet0/1
02:31:12:

DHCP_SNOOPING_SW host tracking not found for update add dynamic
 (192.168.0.241, 0.0.0.0, 0050.5699.4ea1) vlan 1


02:31:12: DHCP_SNOOPING: direct forward dhcp reply to output port: FastEthernet0/1.
02:31:12:

sw_host_track-ev:Add event: 0050.5699.4ea1, 192.168.0.241, FastEthernet0/1


02:31:12: sw_host_track-obj_create:0050.5699.4ea1(192.168.0.241) Cache entry created
```

```
02:31:12:

sw_host_track-ev:Activating host 0050.5699.4ea1, 192.168.0.241 on
 interface FastEthernet0/1


02:31:12: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
```

默认情况下，每30秒发送一次ARP探测：

<#root>

```
02:41:12: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
02:41:12: sw_host_track-ev:0050.5699.4ea1:

Send Host probe (0)


02:41:12: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
02:41:42: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
02:41:42: sw_host_track-ev:0050.5699.4ea1:

Send Host probe (1)


02:41:42: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
02:42:12: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
02:42:12: sw_host_track-ev:0050.5699.4ea1:

Send Host probe (2)


02:42:12: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
02:42:42: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
02:42:42:

sw_host_track-obj_destroy:0050.5699.4ea1(192.168.0.241): Cache entry deleted


02:42:42: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
```



3	30.0110700	Cisco_e6:cf:83	Vmware_99:4e:a1	ARP	60 who has 192.168.0.241? Tell 0.0.0.0
4	30.0111260	Vmware_99:4e:a1	Cisco_e6:cf:83	ARP	42 192.168.0.241 is at 00:50:56:99:4e:a1
5	60.0235090	Cisco_e6:cf:83	Vmware_99:4e:a1	ARP	60 who has 192.168.0.241? Tell 0.0.0.0
6	60.0235250	Vmware_99:4e:a1	Cisco_e6:cf:83	ARP	42 192.168.0.241 is at 00:50:56:99:4e:a1
7	90.0230090	Cisco_e6:cf:83	Vmware_99:4e:a1	ARP	60 who has 192.168.0.241? Tell 0.0.0.0
8	90.0230250	Vmware_99:4e:a1	Cisco_e6:cf:83	ARP	42 192.168.0.241 is at 00:50:56:99:4e:a1

从设备跟踪表中删除条目后，相应的DHCP绑定条目仍存在：

<#root>

```
BSNS-3560-1#

show ip device tracking all


IP Device Tracking = Enabled
```

```
------------------------------------------------------------
  IP Address     MAC Address       Interface          STATE
------------------------------------------------------------

BSNS-3560-1#
```

show ip dhcp binding

```
IP address       Client-ID/              Lease expiration        Type
                 Hardware address
192.168.0.241    0100.5056.994e.a1       Mar 02 1993 03:06 AM    Automatic
```

当您有ARP响应时，存在问题，但设备跟踪条目仍然会被删除。

该Bug似乎出现在版本12.2.33中，未出现在版本12.2.55或15.x软件中。

另外，在使用L2端口（接入端口）和L3端口（无交换机端口）时，也存在一些差异。

探测功能和ARP监听

使用ARP监听功能跟踪设备：

<#root>

```
BSNS-3560-1#
```

show debugging

```
ARP:
  ARP packet debugging is on
Arp Snoop:
  Arp Snooping debugging is on

03:43:36: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
03:43:36: sw_host_track-ev:0050.5699.4ea1: Send Host probe (0)
03:43:36:
```

IP ARP: sent req src 0.0.0.0 001f.27e6.cf83,

dst 192.168.0.241 0050.5699.4ea1 FastEthernet0/1

```
03:43:36: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
03:43:36: IP ARP: rcvd rep src 192.168.0.241 0050.5699.4ea1, dst 0.0.0.0 Vlan1
```

版本12.2.55的IP设备跟踪 — 隐藏命令

对于此处的版本12.2，请使用隐藏命令将其激活：

```
<#root>

BSNS-3560-1#

show ip device tracking all


IP Device Tracking = Enabled
IP Device Tracking Probe Count = 2
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
------------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface                STATE
------------------------------------------------------------------------
192.168.0.244   0050.5699.4ea1  55   FastEthernet0/1          ACTIVE

Total number interfaces enabled: 1
Enabled interfaces:

  Fa0/1



BSNS-3560-1#

ip device tracking interface fa0/48


BSNS-3560-1#

show ip device tracking all


IP Device Tracking = Enabled
IP Device Tracking Probe Count = 2
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
------------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface                STATE
------------------------------------------------------------------------
10.48.67.87     000c.2978.825d  1006 FastEthernet0/48         ACTIVE
10.48.67.31     020a.dada.dada  1006 FastEthernet0/48         ACTIVE
10.48.66.245    acf2.c5ed.8171  1006 FastEthernet0/48         ACTIVE
192.168.0.244   0050.5699.4ea1  55   FastEthernet0/1          ACTIVE
10.48.66.193    000c.2997.4ca1  1006 FastEthernet0/48         ACTIVE
10.48.66.186    0050.5699.3431  1006 FastEthernet0/48         ACTIVE

Total number interfaces enabled: 2
Enabled interfaces:

  Fa0/1, Fa0/48
```

版本12.2.55的IP设备跟踪 — 静态IP示例

在本示例中，PC已配置了静态IP地址。调试表明，在获得ARP响应(MSG=2)后，设备跟踪条目会更新。

```
<#root>

01:03:16: sw_host_track-ev:0050.5699.4ea1 Stopping cache timer
```

```
01:03:16: sw_host_track-ev:0050.5699.4ea1: Send Host probe (0)
01:03:16: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
01:03:16: sw_host_track-ev:host_track_notification: Add event for host 0050.5699.4ea1,
 192.168.0.241 on interface FastEthernet0/1, vlan 1
01:03:16: sw_host_track-ev:Async Add event for host 0050.5699.4ea1, 192.168.0.241
 on interface FastEthernet0/1
01:03:16: sw_host_track-ev:
```

MSG = 2

```
01:03:16: sw_host_track-ev:Add event: 0050.5699.4ea1, 192.168.0.241, FastEthernet0/1
01:03:16: sw_host_track-ev:
```

0050.5699.4ea1: Cache entry refreshed

```
01:03:16: sw_host_track-ev:Activating host 0050.5699.4ea1, 192.168.0.241 on
 interface FastEthernet0/1
01:03:16: sw_host_track-ev:0050.5699.4ea1 Starting cache timer: 30 seconds
```

因此，来自PC的每个ARP请求都会更新设备跟踪表（来自ARP数据包的发送方MAC地址和发送方IP地址）。

版本15.x的IP设备跟踪

请务必记住，LAN Lite版本不支持某些功能，例如用于802.1x的DACL（请注意 — Cisco Feature Navigator并不总是显示正确的信息）。

可以执行版本12.2中的隐藏命令，但无效。在软件版本15.x中，IP设备跟踪(IPDT)默认仅对启用了802.1x的接口启用：

<#root>

```
bsns-3750-5#
```

show ip device tracking all

```
IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface                STATE
---------------------------------------------------------------------
192.168.10.12   0007.5032.6941  100  GigabitEthernet1/0/1     ACTIVE
192.168.2.200   000c.29d7.0617  1    GigabitEthernet1/0/1     ACTIVE

Total number interfaces enabled: 2
Enabled interfaces:
```

Gi1/0/1, Gi1/0/2

```
bsns-3750-5#
```

```
show run int g1/0/3

Building configuration...

Current configuration : 38 bytes
!
interface GigabitEthernet1/0/3

bsns-3750-5(config)#
int g1/0/3

bsns-3750-5(config-if)#
switchport mode access

bsns-3750-5(config-if)#
authentication port-control auto

bsns-3750-5(config-if)#
do show ip device tracking all

IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
-----------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface                STATE
-----------------------------------------------------------------------
192.168.10.12   0007.5032.6941  100  GigabitEthernet1/0/1     ACTIVE
192.168.2.200   000c.29d7.0617  1    GigabitEthernet1/0/1     ACTIVE

Total number interfaces enabled: 3
Enabled interfaces:
  Gi1/0/1, Gi1/0/2,
Gi1/0/3
```

从端口删除802.1x配置后，也会从该端口删除IPDT。

端口状态可能是“DOWN”，因此必须具有“switchport mode access”和“authentication port-control auto”才能在该端口上激活IP设备跟踪。

最大接口设备限制设置为10:

```
<#root>

bsns-3750-5(config-if)#
ip device tracking maximum

 ?
  <1-10>  Maximum devices
```

Cisco IOS-XE®的IP设备跟踪

同样，与Cisco IOS版本15.x相比，Cisco IOS-XE 3.3上的行为也发生了变化。

版本12.2中的隐藏命令已过时，但现在返回以下错误：

```
<#root>

3850-1#

no ip device tracking int g1/0/48

% Command accepted but obsolete, unreleased or unsupported; see documentation.
```

在Cisco IOS-XE中，会为所有接口（甚至未配置802.1x的接口）激活设备跟踪：

```
<#root>

3850-1#

show ip device tracking all

Global IP Device Tracking for clients = Enabled
Global IP Device Tracking Probe Count = 3
Global IP Device Tracking Probe Interval = 30
Global IP Device Tracking Probe Delay Interval = 0
---------------------------------------------------------------------------------------
  IP Address    MAC Address   Vlan  Interface           Probe-Timeout
   State    Source
---------------------------------------------------------------------------------------
10.48.39.29     000c.29bd.3cfa 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.28     0016.9dca.e4a7 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.76.117    0021.a0ff.5540 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.21     00c0.9f87.7471 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.16     0050.5699.1093 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.76.191.247   0024.9769.58cf 20   GigabitEthernet1/0/48  30
   ACTIVE   ARP
192.168.99.4    d48c.b52f.4a1e 99   GigabitEthernet1/0/12  30
   INACTIVE ARP
10.48.39.13     000c.296e.8dbc 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.15     0050.5699.128d 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.9      0012.da20.8c00 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.8      6c20.560e.1b64 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.11     000c.29e9.db25 1    GigabitEthernet1/0/48  30
```

```
   ACTIVE   ARP
10.48.39.5      0014.f15f.f7ca 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.4      000c.2972.57bc 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.7      5475.d029.74cf 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.76.108    001c.58de.9340 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.1      0006.f62a.c4a3 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.3      0050.5699.1bee 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.76.84     0015.58c5.e8b7 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.56     0015.fa13.9a40 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.59     0050.5699.1bf4 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP
10.48.39.58     000c.2957.c7ad 1    GigabitEthernet1/0/48  30
   ACTIVE   ARP


Total number interfaces enabled: 57
Enabled interfaces:
  Gi1/0/1, Gi1/0/2, Gi1/0/3, Gi1/0/4, Gi1/0/5, Gi1/0/6, Gi1/0/7,
  Gi1/0/8, Gi1/0/9, Gi1/0/10, Gi1/0/11, Gi1/0/12, Gi1/0/13, Gi1/0/14,
  Gi1/0/15, Gi1/0/16, Gi1/0/17, Gi1/0/18, Gi1/0/19, Gi1/0/20, Gi1/0/21,
  Gi1/0/22, Gi1/0/23, Gi1/0/24, Gi1/0/25, Gi1/0/26, Gi1/0/27, Gi1/0/28,
  Gi1/0/29, Gi1/0/30, Gi1/0/31, Gi1/0/32, Gi1/0/33, Gi1/0/34, Gi1/0/35,
  Gi1/0/36, Gi1/0/37, Gi1/0/38, Gi1/0/39, Gi1/0/40, Gi1/0/41, Gi1/0/42,
  Gi1/0/43, Gi1/0/44, Gi1/0/45, Gi1/0/46, Gi1/0/47,

Gi1/0/48,

 Gi1/1/1,
  Gi1/1/2, Gi1/1/3, Gi1/1/4, Te1/1/1, Te1/1/2, Te1/1/3, Te1/1/4
3850-1#$

3850-1#sh run int

g1/0/48


Building configuration...

Current configuration : 39 bytes
!
interface GigabitEthernet1/0/48
end

3850-1(config-if)#

ip device tracking maximum

 ?
  <0-65535>  Maximum devices (0 means disabled)
```

此外，每个端口的最大条目数没有限制（0表示已禁用）。

802.1x的IP设备跟踪和12.2.55版的DACL

如果为802.1x配置了DACL，则使用设备跟踪条目来填充设备的IP地址。

此示例显示对静态配置的IP执行的设备跟踪：

```
<#root>

BSNS-3560-1#

show ip device tracking all


IP Device Tracking = Enabled
IP Device Tracking Probe Count = 2
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
---------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface                STATE
---------------------------------------------------------------------

192.168.0.244

  0050.5699.4ea1  2    FastEthernet0/1          ACTIVE

Total number interfaces enabled: 1
Enabled interfaces:
  Fa0/1
```

这是使用“permit icmp any any”DACL构建的802.1x会话：

```
<#root>

BSNS-3560-1#

sh authentication sessions  interface fa0/1


            Interface:  FastEthernet0/1
          MAC Address:  0050.5699.4ea1


 IP Address:  192.168.0.244


            User-Name:  cisco
               Status:  Authz Success
               Domain:  DATA
      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  single-host
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  2


    ACS ACL:  xACSACLx-IP-DACL-516c2694
```

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3042A900000008008900C5
      Acct Session ID:  0x0000000D
               Handle:  0x19000008

Runnable methods list:
       Method   State
       dot1x    Authc Success
```

<#root>

```
BSNS-3560-1#
```

show epm session summary

```
EPM Session Information
-----------------------
Total sessions seen so far : 1
Total active sessions      : 1

Interface               IP Address        MAC Address     Audit Session Id:
------------------------------------------------------------------------------
FastEthernet0/1         192.168.0.244     0050.5699.4ea1   0A3042A900000008008900C5
```

下面显示了一个应用的ACL:

<#root>

```
 BSNS-3560-1#
```

show ip access-lists

```
Extended IP access list Auth-Default-ACL
    10 permit udp any range bootps 65347 any range bootpc 65348
    20 permit udp any any range bootps 65347
    30 deny ip any any (8 matches)
```

Extended IP access list xACSACLx-IP-DACL-516c2694 (per-user)

```
    10 permit icmp any any (6 matches)
```

此外，fa0/1接口的ACL也相同：

<#root>

```
BSNS-3560-1#
```

show ip access-lists interface fa0/1

```
     permit icmp any any
```

即使默认值为dot1x ACL:

```
<#root>

BSNS-3560-1#

show ip interface fa0/1


FastEthernet0/1 is up, line protocol is up
  Inbound  access list is Auth-Default-ACL
```

ACL应使用“any”作为192.168.0.244。对于身份验证代理，此方式是类似的，但对于802.1x DACL src "any"，不会更改为检测到的PC IP。

对于身份验证代理，来自ACS的一个原始ACL将通过show ip access-list命令缓存并显示，并且使用show ip access-list interface fa0/1命令在接口上应用一个特定(Per-User with specific IP)ACL。但是，身份验证代理不使用设备IP跟踪。

如果未正确检测到IP地址，该怎么办？禁用设备跟踪后：

```
<#root>

BSNS-3560-1#

show authentication sessions interface fa0/1


            Interface:  FastEthernet0/1
          MAC Address:  0050.5699.4ea1


 IP Address:  Unknown


            User-Name:  cisco
               Status:  Authz Success
               Domain:  DATA
      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  single-host
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  2


ACS ACL:  xACSACLx-IP-DACL-516c2694


      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3042A9000000000000C775
```

```
      Acct Session ID:  0x00000001
               Handle:  0xB0000000

Runnable methods list:
       Method   State
       dot1x    Authc Success
```

因此，没有附加IP地址，但DACL仍然适用：

```
<#root>

BSNS-3560-1#

show ip access-lists


Extended IP access list Auth-Default-ACL
    10 permit udp any range bootps 65347 any range bootpc 65348
    20 permit udp any any range bootps 65347
    30 deny ip any any (4 matches)
Extended IP access list

 xACSACLx-IP-DACL-516c2694 (per-user)


    10 permit icmp any any
```

在此场景中，不需要对802.1x进行设备跟踪。唯一的区别是，事先知道客户端的IP地址可用于RADIUS访问请求。附加属性8后：

```
radius-server attribute 8 include-in-access-req
```

它存在于Access-Request中，在ACS上可以创建更精细的授权规则：

```
00:17:44: RADIUS(00000001): Send Access-Request to 10.48.66.185:1645 id 1645/27, len 257
00:17:44: RADIUS:  authenticator F8 17 06 CE C1 85 E8 E8 - CB 5B 57 96 6C 07 CE CA
00:17:44: RADIUS:  User-Name           [1]   7   "cisco"
00:17:44: RADIUS:  Service-Type        [6]   6   Framed                   [2]
00:17:44: RADIUS:  Framed-IP-Address   [8]   6   192.168.0.244
```

请记住，TrustSec还需要IP设备跟踪，以实现IP到SGT的绑定。

802.1x的IP设备跟踪和15.x版的DACL

在DACL中，版本15.x和版本12.2.55有何区别？在软件版本15.x中，它与auth-proxy的工作方式相同。

当输入show ip access-list命令（来自AAA的缓存响应）时可以看到通用ACL，但是在show ip access-list interface fa0/1命令之后，src "any"会被主机的源IP地址替换（通过IP设备跟踪获知）。

以下是一个电话和PC在一个端口(g1/0/1)、软件版本15.0.2SE2(3750X)上的示例：

<#root>

```
bsns-3750-5#sh authentication sessions interface g1/0/1


            Interface:  GigabitEthernet1/0/1
          MAC Address:

 0007.5032.6941


           IP Address:

192.168.10.12


            User-Name:  00-07-50-32-69-41
               Status:  Authz Success
               Domain:

 VOICE


      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:

100


              ACS ACL:

 xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2


      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  C0A80001000001012B680D23
      Acct Session ID:  0x0000017B
               Handle:  0x99000102

Runnable methods list:
       Method   State
       dot1x    Failed over


 mab

 Authc Success



----------------------------------------
            Interface:  GigabitEthernet1/0/1
          MAC Address:
```

```
 0050.5699.4ea1

           IP Address:
192.168.2.200

            User-Name:
cisco

               Status:  Authz Success
               Domain:
 DATA

      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:
 20

              ACS ACL:
xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2

      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  C0A80001000001BD336EC4D6
      Acct Session ID:  0x000002F9
               Handle:  0xF80001BE


Runnable methods list:
       Method   State

  dot1x    Authc Success

       mab      Not run
```

电话通过MAC身份验证绕行(MAB)进行身份验证，而PC使用dot1x。电话和PC使用相同的ACL:

```
<#root>

bsns-3750-5#
show ip access-lists xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2

Extended IP access list xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2 (
per-user
```

```
)
    10
permit ip any any
```

但是，在接口级别验证后，源地址已被设备的IP地址替换。

IP设备跟踪会触发更改，并且它可能随时发生（远远晚于身份验证会话和ACL下载）：

```
<#root>
bsns-3750-5#
show ip access-lists interface g1/0/1

     permit ip
host 192.168.2.200
 any (5 matches)
     permit ip
host 192.168.10.12
 any
```

两个MAC地址均标记为静态：

```
<#root>
bsns-3750-5#
sh mac address-table interface g1/0/1

          Mac Address Table
-------------------------------------------

Vlan    Mac Address       Type        Ports
----    -----------       --------    -----
  20    0050.5699.4ea1
 STATIC
      Gi1/0/1
 100    0007.5032.6941
 STATIC
      Gi1/0/1
```

特定ACL条目

DACL中的源“any”何时替换为主机IP地址？仅当同一端口上至少存在两个会话（两个Supplicant客户端）时。

如果只有一个会话，则无需替换源“any”。

当存在多个会话时，问题就会出现，而且并非所有会话的IP设备跟踪都知道主机的IP地址。在这种情况下，某些条目仍为“任意”。

某些平台上的行为有所不同。例如，在版本15.0(2)EX的2960X上，即使每个端口只有一个身份验证会话，ACL也始终是特定的。

但是，对于3560X和3750X版本15.0(2)SE，至少需要两个会话以使该ACL成为特定的。

控制方向

默认情况下，control-direction为both:

```
<#root>

bsns-3750-5(config)#

int g1/0/1


bsns-3750-5(config-if)#

authentication control-direction ?


  both  Control traffic in BOTH directions
  in    Control inbound traffic only

bsns-3750-5(config-if)#

authentication control-direction both
```

这意味着在对请求方进行身份验证之前，无法向该端口发送或从该端口发送流量。对于“in”模式，流量可能从端口发送到请求方，而不是从请求方发送到端口（可能对LAN唤醒功能有用）。

但是，交换机仅将ACL应用于“in”方向。使用哪种模式并不重要。

```
<#root>

bsns-3750-5#

sh ip access-lists interface g1/0/1 out


bsns-3750-5#

sh ip access-lists interface g1/0/1 in
```

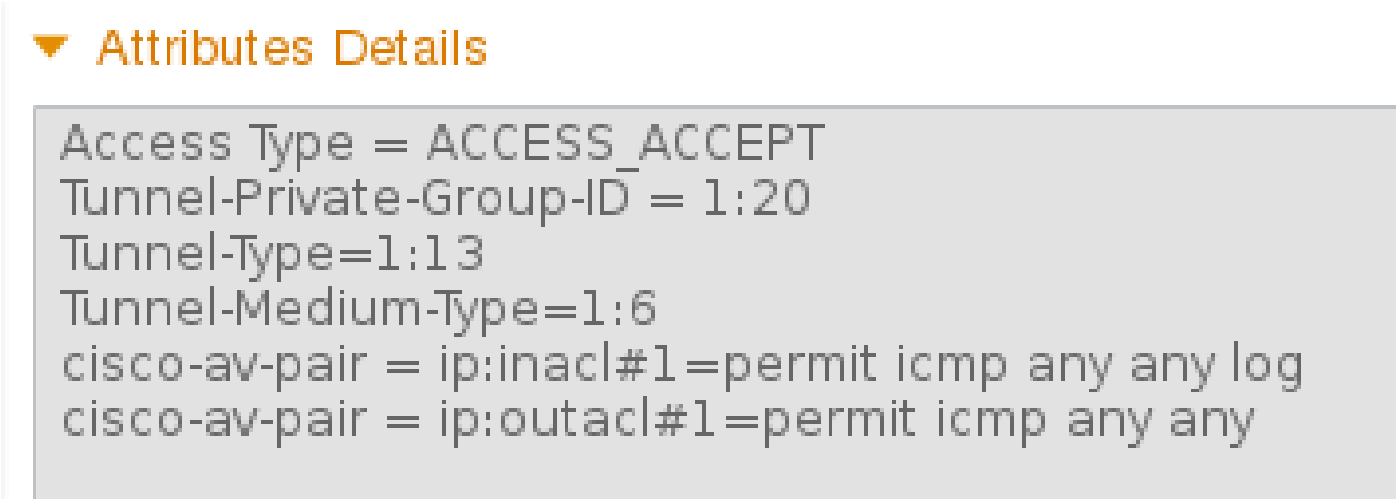
```
    permit ip host 192.168.2.200 any
    permit ip host 192.168.10.12 any
```

这基本上意味着，在身份验证之后，ACL将应用于流向端口的流量（方向），并且允许所有来自端口的流量（方向）。

802.1x的IP设备跟踪和版本15.x的每用户ACL

也可以使用在cisco-av-pair "ip:inacl"和"ip:outacl"中传递的每用户ACL。

此示例配置类似于之前的配置，但这次电话使用DACL，而PC使用每用户ACL。PC的ISE配置文件是：



电话仍然应用了DACL:

<#root>

bsns-3750-5#

show authentication sessions interface g1/0/1

```
            Interface:  GigabitEthernet1/0/1
          MAC Address:  0007.5032.6941
           IP Address:
```

192.168.10.12

```
            User-Name:  00-07-50-32-69-41
               Status:  Authz Success
               Domain:
```

VOICE

```
      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  multi-auth
     Oper control dir:  both
```

```
        Authorized By:  Authentication Server
          Vlan Policy:  100
              ACS ACL:
```

xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  C0A8000100000568431143D8
      Acct Session ID:  0x000006D2
               Handle:  0x84000569

Runnable methods list:
       Method   State
       dot1x    Failed over
       mab      Authc Success

bsns-3750-5#
```

sh ip access-lists xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2

```
Extended IP access list xACSACLx-IP-PERMIT_ALL_TRAFFIC-51134bb2 (per-user)
    10
```

permit ip any any

但是，同一端口上的PC使用每用户ACL:

<#root>

```
    Interface:  GigabitEthernet1/0/1
          MAC Address:  0050.5699.4ea1
           IP Address:
```

192.168.2.200

```
            User-Name:  cisco
               Status:  Authz Success
               Domain:
```

DATA

```
      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  20
```

Per-User ACL: permit icmp any any log

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  C0A800010000056743114008
      Acct Session ID:  0x000006D1
```

```
               Handle:  0x9D000568
```

要验证如何在gig1/0/1端口上合并它：

<#root>

```
bsns-3750-5#
```

show ip access-lists interface g1/0/1

```
     permit icmp host 192.168.2.200 any log
     permit ip host 192.168.10.12 any
```

第一个条目取自每用户ACL（注意log关键字），第二个条目取自DACL。

它们都是由特定IP地址的IP设备跟踪重写的。

可以使用debug epm all命令验证每用户ACL:

<#root>

```
Apr 12 02:30:13.489: EPM_SESS_EVENT:
```

IP Per-User ACE: permit icmp any any log received

```
Apr 12 02:30:13.489: EPM_SESS_EVENT:Recieved string
```

GigabitEthernet1/0/1#IP#7844C6C

```
Apr 12 02:30:13.489: EPM_SESS_EVENT:Add ACE [permit icmp any any log] to ACL
 [GigabitEthernet1/0/1#IP#7844C6C]
Apr 12 02:30:13.497: EPM_SESS_EVENT:Executed [ip access-list extended
 GigabitEthernet1/0/1#IP#7844C6C] command through parse_cmd. Result= 0
Apr 12 02:30:13.497: EPM_SESS_EVENT:Executed [permit icmp any any log]
 command through parse_cmd. Result= 0
Apr 12 02:30:13.497: EPM_SESS_EVENT:Executed [end] command through
 parse_cmd. Result= 0
Apr 12 02:30:13.497: EPM_SESS_EVENT:
```

Notifying PD regarding Policy (NAMED ACL)
application on the interface GigabitEthernet1/0/1

还可以通过show ip access-lists命令：

<#root>

```
bsns-3750-5#
```

show ip access-lists

```
Extended IP access list GigabitEthernet1/0/1#IP#7844C6C (per-user)
    10 permit icmp any any log
```

ip:outacl属性如何？版本15.x中完全省略了该功能。已收到该属性，但交换机不应用/处理该属性。

与DACL比较时的差异

如Cisco Bug ID CSCut25702中所述，每用户ACL的行为与DACL不同。

只包含一个条目(“permit ip any any”)和一个连接到端口的请求方的DACL可以在未启用IP设备跟踪的情况下正常工作。

“any”参数不会被替换，并且允许所有流量。但是，对于每用户ACL，必须启用IP设备跟踪。

如果它被禁用，并且只有“permit ip any any”条目和一个请求方，则所有流量都会被阻止。

802.1x的IP设备跟踪和15.x版的Filter-ID ACL

此外，还可以使用IETF属性filter-id [11]。AAA服务器返回ACL名称，该名称在交换机本地定义。
ISE配置文件可能如下所示：

请注意，您需要指定方向（输入或输出）。为此，需要手动添加属性：

然后，调试显示：

<#root>

debug epm all

```
Apr 12 23:41:05.170: EPM_SESS_EVENT:Filter-Id :
```

Filter-ACL received

```
Apr 12 23:41:05.170: EPM_SESS_EVENT:Notifying PD regarding Policy (NAMED ACL)
 application on the interface GigabitEthernet1/0/1
```

对于已通过身份验证的会话，也会显示该ACL:

<#root>

```
bsns-3750-5#
```

show authentication sessions interface g1/0/1

```
            Interface:  GigabitEthernet1/0/1
          MAC Address:  0050.5699.4ea1
           IP Address:  192.168.2.200
            User-Name:  cisco
               Status:  Authz Success
               Domain:  DATA
      Security Policy:  Should Secure
      Security Status:  Unsecure
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  20
```

Filter-Id: Filter-ACL

```
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  C0A800010000059E47B77481
      Acct Session ID:  0x00000733
               Handle:  0x5E00059F

Runnable methods list:
       Method   State
       dot1x
```

Authc Success

```
       mab      Not run
```

并且，当ACL绑定到接口时：

<#root>

bsns-3750-5#

show ip access-lists interface g1/0/1

```
     permit icmp host 192.168.2.200 any log
     permit tcp host 192.168.2.200 any log
```

请注意，此ACL可以与同一接口上的其他类型ACL合并。例如，在同一交换机端口上有另一个从ISE获取DACL的请求方："permit ip any any"，您可以看到：

<#root>

bsns-3750-5#

show ip access-lists interface g1/0/1

```
     permit icmp host 192.168.2.200 any log
     permit tcp host 192.168.2.200 any log
     permit ip host 192.168.10.12 any
```

请注意，IP设备跟踪会重写每个源（请求方）的源IP。

"out"过滤器列表如何处理？同样（作为每用户ACL），交换机不使用该值。

IP设备跟踪 — 默认值和最佳实践

对于早于15.2(1)E的版本，在可以使用任何IPDT功能之前，需要首先使用此CLI命令全局启用该功能：

<#root>

(config)#

ip device tracking

对于版本15.2(1)E及更高版本，不再需要ip device tracking命令。IPDT仅在依赖它的功能启用它时启用。

如果没有启用IPDT的功能，则禁用IPDT。"no ip device tracking"命令不起作用。特定功能具有启用/禁用IPDT的控制。

启用IPDT时，必须记住上的"重复IP地址"问题。有关详细信息，请参阅排除"重复IP地址0.0.0.0"错误消息故障。

建议在TRUNK端口上禁用IPDT:

```
<#root>

(config-if)#

no ip device tracking
```

在更高版本的Cisco IOS上，它是不同的命令：

```
<#root>

(config-if)#

 ip device tracking maximum 0
```

建议在接入端口上启用IPDT并延迟ARP探测以避免“重复IP地址”问题：

```
<#root>

(config-if)#

 ip device tracking probe delay 10
```

版本15.x的接口ACL重写

对于接口ACL，它在身份验证之前起作用：

```
<#root>

interface GigabitEthernet1/0/2
 description windows7
 switchport mode access

 ip access-group test1 in


 authentication order mab dot1x
 authentication port-control auto
 mab
 dot1x pae authenticator
end

bsns-3750-5#

show ip access-lists test1


Extended IP access list test1
    10 permit tcp any any log-input
```

但是，身份验证成功后，它将由AAA服务器返回的ACL重写（覆盖）（无论它是DACL、ip:inacl还是filterid都无关紧要）。

该ACL(test1)可以阻止流量（通常在开放模式下允许该流量），但在身份验证之后，不再重要。

即使没有从AAA服务器返回ACL，也会覆盖接口ACL并提供完全访问。

这有点误导，因为三重内容可寻址存储器(TCAM)表示ACL仍然绑定在接口级别。

以下是3750X版本15.2.2的一个示例：

<#root>

bsns-3750-6#

show platform acl portlabels interface g1/0/2

Port based ACL: (asic 1)

  Input Label: 5   Op Select Index: 255
    Interface(s): Gi1/0/2
    Access Group:

test1

, 4 VMRs
    Ip Portal: 0 VMRs
    IP Source Guard: 0 VMRs
    LPIP: 0 VMRs
    AUTH: 0 VMRs
    C3PLACL: 0 VMRs
    MAC Access Group: (none), 0 VMRs

该信息仅对接口级别有效，对会话级别无效。可以从以下内容推断出一些更多信息（表示复合ACL）：

<#root>

bsns-3750-6#

show ip access-lists interface g1/0/2

permit ip host 192.168.1.203 any

Extended IP access list

test1

```
 10 permit icmp host x.x.x.x host n.n.n.n
```

第一个条目创建为“permit ip any any”，DACL为成功身份验证返回（并且“any”被设备跟踪表中的条目替换）。

第二个条目是接口ACL的结果，应用于所有新的身份验证（在授权之前）。

遗憾的是，（同样取决于平台）两个ACL都是串联的。这发生在3750X上的15.2.2版上。

这意味着，对于授权会话，两者均适用。首先是DACL，然后是接口ACL。

因此，当您添加显式“deny ip any any”时，DACL不会考虑接口ACL。

DACL中通常没有明确的deny语句，然后应用接口ACL。

3750X上版本15.0.2的行为是相同的，但sh ip access-list interface命令不再显示接口ACL（但它仍然与接口ACL串联，除非DACL中存在明确的deny）。

用于802.1x的默认ACL

默认ACL有两种类型：

- auth-default-ACL-OPEN — 用于开放模式
- auth-default-ACL — 用于封闭访问

当端口处于未授权状态时，将同时使用auth-default-ACL和auth-default-ACL-OPEN。 默认情况下，使用封闭访问。

这意味着，在身份验证之前，除auth-default-ACL允许的流量外，所有流量都会被丢弃。

这样，DHCP流量在授权成功之前会得到允许。

会分配IP地址，并且可正确应用下载的DACL。

该ACL是自动创建的，在配置中找不到。

```
<#root>

bsns-3750-5#

sh run | i Auth-Default


bsns-3750-5#

sh ip access-lists Auth-Default-ACL

Extended IP access list

Auth-Default-ACL
```

```
    10 permit udp any range bootps 65347 any range bootpc 65348 (22 matches)
    20 permit udp any any range bootps 65347 (12 matches)
    30 deny ip any any
```

它是为第一次身份验证（在身份验证和授权阶段）动态创建的，并在删除最后一个会话后删除。

Auth-Default-ACL仅允许DHCP流量。在身份验证成功并下载新的DACL后，它将应用于该会话。

当模式更改为open auth-default-ACL-OPEN出现时，其使用方式与Auth-Default-ACL完全相同：

```
<#root>

bsns-3750-5(config)#int g1/0/2
bsns-3750-5(config-if)#authentication open

bsns-3750-5#

show ip access-lists

Extended IP access list

Auth-Default-ACL-OPEN

    10 permit ip any any
```

两个ACL都可以自定义，但配置中从未出现过。

```
<#root>

bsns-3750-5(config)#

ip access-list extended Auth-Default-ACL

bsns-3750-5(config-ext-nacl)#permit udp any any

bsns-3750-5#

sh ip access-lists

Extended IP access list Auth-Default-ACL
    10 permit udp any range bootps 65347 any range bootpc 65348 (22 matches)
    20 permit udp any any range bootps 65347 (16 matches)
    30 deny ip any any
    40 permit udp any any

bsns-3750-5#

sh run | i Auth-Def
```

```
bsns-3750-5#
```

打开模式

上一节描述了ACL的行为（包括默认情况下用于开放模式的ACL）。打开模式的行为是：

- 当会话处于未授权状态时，它允许所有流量（根据默认auth-default-ACL-OPEN）。
- 会话在身份验证/授权期间处于未授权状态(适用于加密设备型号E(PXE)引导方案)或在此过程失败之后处于未授权状态（适用于称为“低影响模式”的方案）。
- 当会话进入多个平台的授权状态时，会连接ACL并使用第一个DACL，然后是接口ACL。
- 对于多身份验证或多域，可能同时存在多个处于不同状态的会话（然后不同的ACL类型适用于每个会话）。

当接口ACL为必填项时

对于多个6500/4500平台，必须配置接口ACL才能正确应用DACL。

以下示例包含4500 sup2 12.2.53SG6，无接口ACL:

<#root>

```
brisk#
```

show run int g2/3

```
!
interface GigabitEthernet2/3
 switchport mode access
 switchport voice vlan 10
 authentication host-mode multi-auth
 authentication open
 authentication order mab dot1x
 authentication priority dot1x mab
 authentication port-control auto
 mab
```

然后，在主机通过身份验证后，下载DACL。未应用，授权失败。

<#root>

```
*Apr 25 04:38:05.239: RADIUS: Received from id 1645/19 10.48.66.74:1645,
```

Access-Accept,

```
 len 209
*Apr 25 04:38:05.239: RADIUS:  authenticator 35 8E 59 E4 D5 CF 8F 9A -
 EE 1C FC 5A 9F 67 99 B2
*Apr 25 04:38:05.239: RADIUS:  User-Name           [1]   41
```

```
  "

#ACSACL#-IP-PERMIT_ALL_TRAFFIC-51ef7db1

"
*Apr 25 04:38:05.239: RADIUS:  State               [24]  40
*Apr 25 04:38:05.239: RADIUS:   52 65 61 75 74 68 53 65 73 73 69 6F 6E 3A 30 61
  [ReauthSession:0a]
*Apr 25 04:38:05.239: RADIUS:   33 30 34 32 34 61 30 30 30 45 46 35 30 46 35 33
  [30424a000EF50F53]
*Apr 25 04:38:05.239: RADIUS:   35 41 36 36 39 33            [ 5A6693]
*Apr 25 04:38:05.239: RADIUS:  Class               [25]  54
*Apr 25 04:38:05.239: RADIUS:   43 41 43 53 3A 30 61 33 30 34 32 34 61 30 30 30
  [CACS:0a30424a000]
*Apr 25 04:38:05.239: RADIUS:   45 46 35 30 46 35 33 35 41 36 36 39 33 3A 69 73
  [EF50F535A6693:is]
*Apr 25 04:38:05.239: RADIUS:   65 32 2F 31 38 30 32 36 39 35 33 38 2F 31 32 38
  [e2/180269538/128]
*Apr 25 04:38:05.239: RADIUS:   36 35 35 33              [ 6553]
*Apr 25 04:38:05.239: RADIUS:  Message-Authenticato[80]  18
*Apr 25 04:38:05.239: RADIUS:   AF 47 E2 20 65 2F 59 39 72 9A 61 5C C5 8B ED F5
         [ G e/Y9ra\]
*Apr 25 04:38:05.239: RADIUS:  Vendor, Cisco       [26]  36
*Apr 25 04:38:05.239: RADIUS:   Cisco AVpair       [1]   30
  "

ip:inacl#1=permit ip any any

"
*Apr 25 04:38:05.239: RADIUS(00000000): Received from id 1645/19
*Apr 25 04:38:05.247:

EPM_SESS_ERR:Failed to apply ACL to interface


*Apr 25 04:38:05.247: EPM_API:In function epm_send_message_to_client
*Apr 25 04:38:05.247: EPM_SESS_EVENT:Sending response message to process
 AUTH POLICY Framework
*Apr 25 04:38:05.247: EPM_SESS_EVENT:Returning feature config
*Apr 25 04:38:05.247: EPM_API:In function epm_acl_feature_free
*Apr 25 04:38:05.247: EPM_API:In function epm_policy_aaa_response
*Apr 25 04:38:05.247: EPM_FSM_EVENT:Event epm_ip_wait_event state changed from
 policy-apply to ip-wait
*Apr 25 04:38:05.247: EPM_API:In function epm_session_action_ip_wait
*Apr 25 04:38:05.247: EPM_API:In function epm_send_ipwait_message_to_client
*Apr 25 04:38:05.247: EPM_SESS_ERR:NULL feature list for client ctx 1B2694B0
 for type DOT1X
*Apr 25 04:38:05.247:

%AUTHMGR-5-FAIL: Authorization failed for client
 (0007.5032.6941) on Interface Gi2/3
 AuditSessionID 0A3043450000000060012C050



brisk#

show authentication sessions



Interface  MAC Address     Method   Domain   Status         Session ID
Gi2/3      0007.5032.6941  mab      VOICE

Authz Failed
```

```
   0A3043450000000060012C050
```

添加接口ACL后：

```
<#root>

brisk#

show ip access-lists all


Extended IP access list all
    10 permit ip any any (63 matches)

brisk#sh run int g2/3
!
interface GigabitEthernet2/3
 switchport mode access
 switchport voice vlan 10

 ip access-group all in


 authentication host-mode multi-auth
 authentication open
 authentication order mab dot1x
 authentication priority dot1x mab
 authentication port-control auto
 mab
```

身份验证和授权成功，DACL应用正确：

```
<#root>

brisk#

show authentication sessions


Interface  MAC Address     Method   Domain   Status         Session ID
Gi2/3      0007.5032.6941  mab      VOICE

 Authz Success

  0A3043450000000800 1A2CE4
```

该行为不依赖于“身份验证打开”。为了接受DACL，您需要打开/关闭模式的接口ACL。

4500/6500上的DACL

在4500/6500上，DACL与acl_snoop DACL一起应用。此处显示了一个包含4500 sup2

12.2.53SG6（电话+ PC）的示例。语音(10)和数据(100)VLAN有单独的ACL:

```
<#root>

brisk#

show ip access-lists


Extended IP access list

acl_snoop_Gi2/3_10


    10 permit ip host

 192.168.2.200

 any
    20 deny ip any any
Extended IP access list

acl_snoop_Gi2/3_100


    10 permit ip host

192.168.10.12

 any
    20 deny ip any any
```

ACL是特定的，因为IPDT具有正确的条目：

```
<#root>

brisk#

show ip device tracking all


IP Device Tracking = Enabled
IP Device Tracking Probe Count = 3
IP Device Tracking Probe Interval = 30
IP Device Tracking Probe Delay Interval = 0
-----------------------------------------------------------------------
  IP Address     MAC Address   Vlan  Interface              STATE
-----------------------------------------------------------------------

192.168.10.12

  0007.5032.6941

100

  GigabitEthernet2/3     ACTIVE

192.168.2.200

  000c.29d7.0617
```

10

```
   GigabitEthernet2/3     ACTIVE
```

经过身份验证的会话确认地址：

<#root>

```
brisk#
```

show authentication sessions int g2/3

```
            Interface:  GigabitEthernet2/3
          MAC Address:  000c.29d7.0617
           IP Address:
```

192.168.2.200

```
            User-Name:  00-0C-29-D7-06-17
               Status:  Authz Success
               Domain:  VOICE
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  N/A
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3043450000003003258E0C
      Acct Session ID:  0x00000034
               Handle:  0x54000030

Runnable methods list:
       Method   State
       mab      Authc Success
       dot1x    Not run

---------------------------------------
            Interface:  GigabitEthernet2/3
          MAC Address:  0007.5032.6941
           IP Address:
```

192.168.10.12

```
            User-Name:  00-07-50-32-69-41
               Status:  Authz Success
               Domain:  DATA
       Oper host mode:  multi-auth
     Oper control dir:  both
        Authorized By:  Authentication Server
          Vlan Policy:  N/A
      Session timeout:  N/A
         Idle timeout:  N/A
    Common Session ID:  0A3043450000002E031D1DB8
      Acct Session ID:  0x00000032
               Handle:  0x4A00002E

Runnable methods list:
```

```
       Method   State
       mab      Authc Success
       dot1x    Not run
```

在此阶段，PC和电话都响应ICMP响应，但接口ACL仅显示：

<#root>

```
brisk#show ip access-lists interface g2/3
     permit ip host
```

192.168.10.12

```
 any
```

为什么？因为DACL只推送到电话(192.168.10.12)。对于PC，使用开放模式的接口ACL:

<#root>

```
interface GigabitEthernet2/3
 ip access-group all in
 authentication open

brisk#
```

show ip access-lists all

```
Extended IP access list all
    10 permit ip any any (73 matches)
```

总之，为PC和电话创建acl_snoop，但仅为电话返回DACL。因此，该ACL被视为绑定到接口。

802.1x的MAC地址状态

当802.1x身份验证启动时，MAC地址仍被视为DYNAMIC，但对该数据包的操作是DROP:

<#root>

```
bsns-3750-5#
```

show authentication sessions

```
Interface  MAC Address     Method   Domain   Status         Session ID
Gi1/0/1
```

0007.5032.6941

```
  dot1x    UNKNOWN
```

Running

C0A8000100000596479F4DCE

bsns-3750-5#

show mac address-table interface g1/0/1

Mac Address Table

Vlan	Mac Address	Type	Ports
100	**0007.5032.6941**	**DYNAMIC**	**Drop**

Total Mac Addresses for this criterion: 1

身份验证成功后，MAC地址变为静态地址，并且提供端口号：

<#root>

bsns-3750-5#

show authentication sessions

Interface	MAC Address	Method	Domain	Status	Session ID
Gi1/0/1	**0007.5032.6941**	mab	VOICE	**Authz Success**	C0A8000100000596479F4DCE

bsns-3750-5#

show mac address-table interface g1/0/1

Mac Address Table

Vlan	Mac Address	Type	Ports
100	**0007.5032.6941**	**STATIC**	**Gi1/0/1**

对于两个域(VOICE/DATA)的所有mab/dot1x会话都是如此。

故障排除

请记得阅读特定软件版本和平台的802.1x配置指南。

如果打开TAC案例，请提供以下命令的输出：

- show tech
- show authentication session interface <xx> detail
- show mac address-table interface <xx>

收集SPAN端口数据包捕获和以下调试也很有用：

- debug radius verbose
- debug epm all
- debug authentication all
- debug dot1x all
- debug authentication feature <yy> all
- debug aaa authentication
- debug aaa authorization

相关信息

- 802.1X身份验证服务配置指南，Cisco IOS XE版本3SE（Catalyst 3850交换机）
- Catalyst 3750-X和Catalyst 3560-X交换机软件配置指南，Cisco IOS版本15.2(1)E
- Catalyst 3750-X和3560-X软件配置指南，版本15.0(1)SE
- Catalyst 3560软件配置指南，版本12.2(52)SE
- 技术支持和文档 - Cisco Systems

关于此翻译
思科采用人工翻译与机器翻译相结合的方式将此文档翻译成不同语言，希望全球的用户都能通过各自的语言得到支持性的内容。

请注意：即使是最好的机器翻译，其准确度也不及专业翻译人员的水平。

Cisco Systems, Inc. 对于翻译的准确性不承担任何责任，并建议您总是参考英文原始文档（已提供链接）。