# Understand 802.1x DACL, Per-User ACL, Filter-ID, and Device Tracking Behavior

## Contents

## Introduction

This document describes the IP device tracking feature, the triggers to add and remove a host, and the impact of device tracking on the 802.1x DACL.

## Device Tracking Theory

This document describes how the IP device tracking feature works, which includes what the triggers are to add and remove a host.

Also, the impact of device tracking on the 802.1x Downloadable Access Control List (DACL) is explained.

The behavior changes between versions and platforms.

The second part of the document focuses on the Access Control List (ACL) returned by the Authentication, Authorization, and Accounting (AAA) server and applied to the 802.1x session.

A comparison between the DACL, Per-User ACL and Filter-ID ACL is presented.

Also, some caveats in regards to the ACL rewrite and default ACL are discussed.

Device tracking adds an entry when:

- it learns the new entry via DHCP snooping.
- it learns the new entry via an Address Resolution Protocol (ARP) request (reads the sender MAC address and the sender IP address from the ARP packet).

That functionality is sometimes called ARP inspection, but it is not the same as Dynamic ARP Inspection (DAI).

That feature is enabled by default and cannot be disabled. It is also called ARP snooping, but debugs do not show it after "debug arp snooping" is enabled.

ARP snooping is enabled by default and cannot be disabled or controlled.

Device tracking removes an entry when there is no response for an ARP request (sending probe for each host in the device tracking table, by default every 30 seconds).

## Device Tracking Configuration

```
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
```

## Device Tracking Tests

<#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
```

## Debugs From Version 12.2.33, IP Device Tracking Updated by DHCP Snooping

DHCP snooping populates the binding table:

<#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
```

After the DHCP binding is added to the database, it triggers the notification for device tracking:

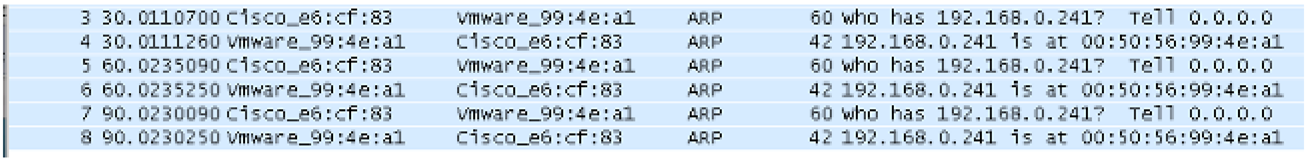
<#root>

```
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
```

ARP probes are sent by default every 30 seconds:

<#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
```

After the entry is removed from the device tracking table, the corresponding DHCP binding entry is still there:

<#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
```

There is the issue when you have an ARP response, but the device tracking entry is removed anyway.

That bug appears to be in Version 12.2.33 and has not appeared in Version 12.2.55 or 15.x software.

Also there are some differences when handling with the L2 port (access-port) and L3 port (no switchport).

## Probe and ARP Snooping

Device tracking with the ARP snooping feature:

<#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
```

## IP Device Tracking for Version 12.2.55 - Hidden Command

For Version 12.2 there, use a hidden command in order to activate it:

<#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**

## IP Device Tracking for Version 12.2.55 - Static IP Example

In this example, the PC has been configured with a static IP address. Debugs show that after you get an ARP response (MSG=2), the device tracking entry is updated.

<#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
```

So every ARP request from the PC updates the device tracking table (the sender MAC address and sender IP address from the ARP packet).

## IP Device Tracking for Version 15.x

It is important to remember that some of the features such as DACL for 802.1x are not supported in the LAN Lite version (beware - Cisco Feature Navigator does not always show the correct information).

The hidden command from Version 12.2 can be executed, but has no effect. In the Software Version 15.x, IP device tracking (IPDT) by default is only enabled for the interfaces which have 802.1x enabled:

<#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
```

After removal of 802.1x configuration from the port, IPDT is also removed from that port.

The port status is possibly be "DOWN", so it is necessary to have "switchport mode access" and "authenticaion port-control auto" in order to have IP device tracking activated on that port.

The maximum interface device limit is set to 10:

<#root>

```
bsns-3750-5(config-if)#

ip device tracking maximum

 ?
  <1-10>  Maximum devices
```

## IP Device Tracking for Cisco IOS-XE®

Again, the behavior on Cisco IOS-XE 3.3 has changed when compared to Cisco IOS Version 15.x.

The hidden command from Version 12.2 is obsolete, but now this error is returned:

<#root>

```
3850-1#

no ip device tracking int g1/0/48


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

In Cisco IOS-XE, device tracking is activated for all the interfaces (even the ones which do not have 802.1x configured):

<#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)
```

Also, there are no limits for maximum entries per port (0 means disabled).

## IP Device Tracking with 802.1x and DACL for Version 12.2.55

If 802.1x is configured with DACL, the device tracking entry is used in order to fill the IP address of device.

This example shows device tracking working for a statically configured IP:

<#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
```

This is an 802.1x session built with "permit icmp any any" DACL:

<#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   0A3042A90000000008008900C5
```

This shows an applied 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)
```

Also, the ACL on the fa0/1 interface is the same:

<#root>

```
BSNS-3560-1#

show ip access-lists interface fa0/1


     permit icmp any any
```

Even though the default is 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
```

It is expected for the ACL to use "any" as **192.168.0.244**. That works like this for auth proxy, but for 802.1x DACL src "any" is not changed to the detected IP of the PC.

For auth proxy, one original ACL from the ACS is cached and shown with the **show ip access-list** command and a specific (Per-User with specific IP) ACL is applied on the interface with the **show ip access-list**

**interface fa0/1** command. However, auth-proxy does not use device IP tracking.

What if the IP address is not detected correctly? After device tracking is disabled:

<#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
```

So no IP address is attached then, but the DACL is still applied:

<#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
```

In this scenario, device tracking for 802.1x is not required. The only difference is that knowing the IP address of the client upfront can be used for a RADIUS access-request. After attribute 8 is attached:

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

It exists in Access-Request and on ACS it is possible to create more granular authorization rules:

```
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
```

Keep in mind that TrustSec also needs IP device tracking for IP to SGT bindings.

## IP Device Tracking with 802.1x and DACL for Version 15.x

What is the difference between Version 15.x and Version 12.2.55 in DACL? In software Version15.x, it works the same as for auth-proxy.

The generic ACL can be seen when the **show ip access-list** command is entered (cached response from AAA), but after the **show ip access-list interface fa0/1** command, the src "any" is replaced by the source IP address of the host (known via IP device tracking).

This is the example for a phone and PC on one port (g1/0/1), software Version 15.0.2SE2 on 3750X:

<#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
```

The phone is authenticated via MAC Authentication Bypass (MAB), while the PC uses dot1x. Both the phone and PC use the same 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
```

However, when verified on the interface level the source has been replaced by the IP address of the device.

IP device tracking triggers that change and it can occur at any time (much later than the authentication session and download of the ACL):

<#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
```

Both MAC addresses are marked as static:

<#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
```

### Specific ACL Entry

When is the source "any" in the DACL replaced with the host IP address? Only when there are at least two sessions on the same port (two supplicants).

There is no need to replace the source "any" when there is only one session.

The problems appear when there are multiple sessions, and for not all of them IP device tracking knows the IP address of the host. In that scenario it is still "any" for some entries.

That behavior is different on some platforms. For example, on the 2960X with Version 15.0(2)EX the ACL is always specific even when there is just one authentication session per port.

However, for the 3560X and 3750X Version 15.0(2)SE, you need to have at least two sessions to make that ACL specific.

### Control-Direction

By default, the control-direction is type 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
```

That means that before the supplicant is authenticated, traffic cannot be sent to or from the port. For "in" mode, the traffic could have been sent from port to supplicant, but not from supplicant to the port (could be useful  for the WAKE on LAN feature).

Still, the switch applies the ACL just on the "in" direction. It does not matter which mode is used.

<#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
```

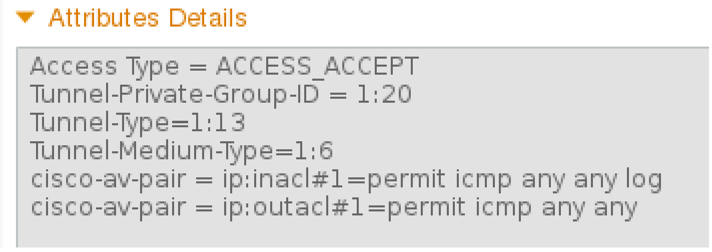
```
     permit ip host 192.168.2.200 any
     permit ip host 192.168.10.12 any
```

That basically means that after authentication the ACL is applied for traffic to the port (in direction) and all traffic is permitted from the port (out direction).

## IP Device Tracking with 802.1x and Per-User ACL for Version 15.x

It is also possible to use a Per-User ACL which is passed in cisco-av-pair "ip:inacl" and "ip:outacl".

This example configuration is similar to a previous configuration, but this time the phone uses DACL and the PC uses Per-User ACL. The ISE profile for the PC is:



```
Access Type = ACCESS_ACCEPT
Tunnel-Private-Group-ID = 1:20
Tunnel-Type=1:13
Tunnel-Medium-Type=1:6
cisco-av-pair = ip:inacl#1=permit icmp any any log
cisco-av-pair = ip:outacl#1=permit icmp any any
```

The phone still has the DACL applied:

<#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
```

However, the PC on the same port uses the Per-User 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:  C0A8000100000567431l400B
      Acct Session ID:  0x000006D1
               Handle:  0x9D000568
```

In order to verify how that is merged on the gig1/0/1 port:

<#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
```

The first entry has been taken from the Per-User ACL (notice the log keyword) and the second entry is taken from the DACL.

Both of them are rewritten by IP device tracking for the specific IP address.

Per-User ACL could be verified with the **debug epm all** command:

<#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
```

And also via the **show ip access-lists** command:

<#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
```

What about the ip:outacl attribute? It is completely omitted in Version 15.x. The attribute has been received, but the switch does not apply/process that attribute.

### Difference when Compared to the DACL

As noted in Cisco bug ID CSCut25702, the Per-User ACL behaves differently than DACL.

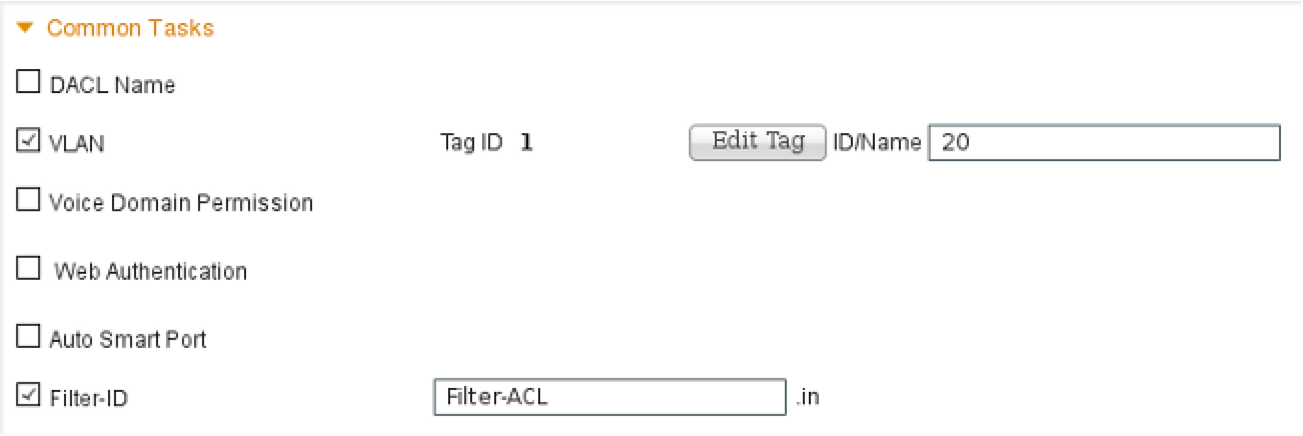
DACL with just one entry ("permit ip any any") and one supplicant connected to a port can work correctly without IP device tracking enabled.

The "any" argument is not substituted and all traffic is permitted. However, for the Per-User ACL it is mandatory to have IP device tracking enabled.

If it is disabled and has just the "permit ip any any" entry and one supplicant, then all traffic is blocked.

## IP Device Tracking with 802.1x and Filter-ID ACL for Version 15.x

Also, the IETF attribute filter-id [11] can be used. The AAA server returns the ACL name, which is defined locally on the switch. The ISE profile could look like this:



Common Tasks

☐ DACL Name

☑ VLAN Tag ID 1 Edit Tag ID/Name 20

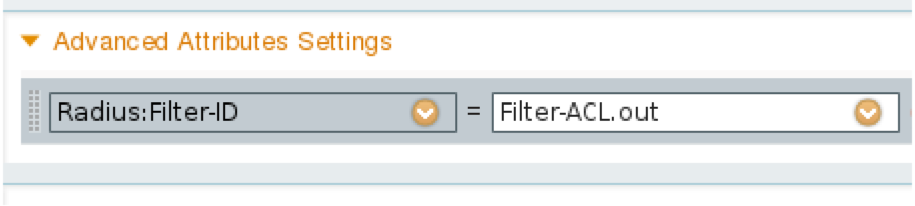
☐ Voice Domain Permission

☐ Web Authentication

☐ Auto Smart Port

☑ Filter-ID Filter-ACL .in

Note that you need to specifiy the direction (in or out). For that it is necessary to add the attribute manually:



Advanced Attributes Settings

Radius:Filter-ID = Filter-ACL.out

Then the debug shows:

<#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
```

That ACL is also shown for the authenticated session:

<#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
```

And, as the ACL is binded to the interface:

<#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
```

Note that this ACL can be merged with other types of ACLs on the same interface. For example, having on the same switch port another supplicant which gets DACL from ISE: "permit ip any any" you could see:

<#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
```

Note that the IP device tracking rewrites the source IP for each source (supplicant).

What about the "out" filter list? Again (as Per-User ACL), it is not used by the switch.

### IP Device Tracking - Defaults and Best Practices

For releases earlier than 15.2(1)E, before any feature can use IPDT it needs to be enabled globally first with this CLI command:

<#root>

```
(config)#

ip device tracking
```

For releases 15.2(1)E and later, the **ip device tracking** command is not needed any more. IPDT is enabled only if a feature that relies on it enables it.

If no feature enables IPDT, IPDT is disabled. The "no ip device tracking" command has no effect. The specific feature has the control to enable/disable IPDT.

When you enable IPDT, you have to remember about the "Duplicate IP Address" issue on . See Troubleshoot "Duplicate IP Address 0.0.0.0" Error Messages for more information.

It is recommended to disable IPDT on a trunk port:

<#root>

```
(config-if)#

no ip device tracking
```

On the later Cisco IOS, it is a different command:

<#root>

```
(config-if)#

 ip device tracking maximum 0
```

It is recommended to enable IPDT on the access port and delay ARP probes in order to avoid the "Duplicate IP Address" issue:

<#root>

```
(config-if)#

 ip device tracking probe delay 10
```

## Interface ACL Rewrite for Version 15.x

For the interface ACL, it works before authentication:

<#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
```

However, after authentication succeeds it is rewritten (override) by the ACL returned from the AAA server (it does not matter if it is DACL, ip:inacl, or filterid).

That ACL (test1) can block the traffic (which would normally be permitted on open mode), but after authentication, does not matter anymore.

Even when no ACL is returned from the AAA server, the interface ACL is overwritten and full access is provided.

That is a bit misleading since Ternary Content Addressable Memory (TCAM) indicates that the ACL is still binded on the interface level.

Here is an example from Version 15.2.2 on 3750X:

<#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
```

That information is valid only for the interface level, not for the session level. Some more information (presents a compounded ACL) can be deduced from:

<#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
```

The first entry is created as "permit ip any any" DACL is returned for successful authentication (and "any" is replaced by an entry from the device tracking table).

The second entry is the result of the interface ACL and is applied for all new authentications (before authorization).

Unfortunately, (again platform dependent) both ACLs are concatenated. That happens on Version 15.2.2 on 3750X.

That means that for authorized session, both of them are applied. First the DACL and second the interface ACL.

That is why when you add explicit "deny ip any any", the DACL does not take into consideration the interface ACL.

Usually there is no explicit deny in the DACL and then the interface ACL is applied after that.

The behavior for Version 15.0.2 on 3750X is the same, but the **sh ip access-list interface** command does not show the interface ACL anymore (but it is still concatenated with the interface ACL unless explicit deny in the DACL exists).

## Default ACL Used for 802.1x

There are two types of default ACLs:

- auth-default-ACL-OPEN - used for open mode

- auth-default-ACL - used for closed access

Both auth-default-ACL and auth-default-ACL-OPEN are used when the port is in the unauthorized state. By default, closed access is used.

That means that before authentication all traffic is dropped except the one permitted by the auth-default-ACL.

This way DHCP traffic is permitted before successful authorization.

The IP address is allocated and the downloaded DACL can be correctly applied.

That ACL is created automatically and cannot be found in the configuration.

<#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
```

It is created dynamically for the first authentication (between authentication and authorization phase) and removed after the last session is removed.

Auth-Default-ACL permits only DHCP traffic. After authentication succeeds and the new DACL is downloaded, it is applied to that session.

When the mode is changed to open auth-default-ACL-OPEN appears and it is used and works in exactly the same way as Auth-Default-ACL:

<#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
```

Both ACLs can be customized, but they are never seen in the configuration.

<#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#
```

# Open Mode

The previous section described the behavior for ACLs (which includes the one used by default for open mode). The behavior for open mode is:

- it allows for all traffic (as per default auth-default-ACL-OPEN) when the session is in an unauthorized state.
- the session is in an unauthorized state during authentication/authorization (good for Encryption Appliance Model E (PXE) boot scenarios) or after that process fails (good for scenarios called "low impact mode").
- when the session moves to the authorized state for multiple platforms, ACLs are concatenated and the first DACL is used, then the interface ACL.
- for multi-auth or multi-domain there are possibly multiple sessions at the same time in different states (then the different ACL type applies for each session).

# When the Interface ACL is Mandatory

For multiple 6500/4500 platforms, the interface ACL is mandatory in order to apply the DACL correctly.

Here is an example with 4500 sup2 12.2.53SG6, no interface 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
```

Then after the host is authenticated, the DACL is downloaded. It is not applied and authorization fails.

<#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 0A304345000000060012C050



brisk#

show authentication sessions



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

Authz Failed

   0A304345000000060012C050
```

After the interface ACL is added:

```
<#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
```

The authentication and authorization succeeds and the DACL is applied correctly:

<#root>

```
brisk#

show authentication sessions



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

 Authz Success

  0A30434500000008001A2CE4
```

The behavior is not dependent on "authentication open". In order to accept the DACL, you need the interface ACL for both open/closed mode.

# DACL on 4500/6500

On the 4500/6500, the DACL is applied with acl_snoop DACLs. An example with 4500 sup2 12.2.53SG6 (phone + PC) is shown here. There is a separate ACL for voice (10) and data (100) VLAN:

<#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
```

ACLs are specific because IPDT has the correct entries:

<#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
```

Authenticated sessions confirm the addresses:

<#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
```

At this stage both the PC and the phone responds to ICMP echo, but the interface ACL presents only:

<#root>

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

**192.168.10.12**

```
 any
```

Why? Because the DACL has been pushed only for the phone (192.168.10.12). For the PC, the interface ACL with open mode is used:

<#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)
```

In summary, acl_snoop is created for both the PC and the phone, but the DACL is returned just for the phone. That is why that ACL is seen as binded to the interface.

# MAC Address Status for 802.1x

When 802.1x authentication starts, the MAC address is still seen as DYNAMIC but action for that packet is 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
```

After successful authentication the MAC address becomes static and the port number is provided:

<#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
```

That is true for all mab/dot1x session for both domains (VOICE/DATA).

# Troubleshoot

Remember to read the 802.1x configuration guide for your specific software version and platform.

If you open a TAC case, provide the output from these commands:

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

It is also good to collect a SPAN port packet capture and these debugs:

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

# Related Information

- **802.1X Authentication Services Configuration Guide, Cisco IOS XE Release 3SE (Catalyst 3850 Switches)**
- **Catalyst 3750-X and Catalyst 3560-X Switch Software Configuration Guide, Cisco IOS Release 15.2(1)E**

- **Catalyst 3750-X and 3560-X Software Configuration Guide, Release 15.0(1)SE**
- **Catalyst 3560 Software Configuration Guide, Release 12.2(52)SE**
- **Technical Support & Documentation - Cisco Systems**