

# How To | Use DHCP Snooping, Option 82, and Filtering on the SwitchBlade x908 switch, and x600, x900 series switches

## Introduction

It has increasingly become a legal requirement for service providers to identify which of their customers were using a specific IP address at a specific time. This means that service providers must be able to:

- know which customer was allocated an IP address at any time.
- guarantee that customers cannot avoid detection by spoofing an IP address that was not actually allocated to them.

These security features provide a traceable history in the event of an official query. Three components are used to provide this traceable history:

- DHCP snooping
- DHCP Option 82
- DHCP filtering

These components also inherently protect against various DoS and man-in-the-middle attacks, either malicious or caused by misconfigured devices.

<b>ACL</b>
Access Control List
<b>DHCP Option 82</b>
The DHCP Relay Agent Information Option. Inserts circuit specific information into a request that is being forwarded to a DHCP server. Specifically the option works by setting two sub-options: Circuit ID and Remote ID.
<b>DHCP snooping</b>
DHCP snooping provides an extra layer of security on the switch via dynamic IP source filtering.  DHCP snooping filters out traffic received from unknown, or 'untrusted' ports, and builds and maintains a DHCP snooping database.

## What information will you find in this document?

This document describes DHCP snooping, DHCP Option 82, and DHCP filtering, and takes you through step-by-step configuration examples.

<b>Table of Contents</b>	Introduction .....	1
	What information will you find in this document? .....	1
	Related How To Notes .....	3
	Which products and software version does it apply to? .....	3
	DHCP snooping .....	4
	AW+ DHCP snooping features .....	5
	Minimum configuration .....	6
	Configuring Option 82 .....	7
	ARP security .....	7
	DHCP filtering .....	8
	DHCP filtering when multiple VLANs are configured on the untrusted port .....	9
	The DHCP snooping database .....	10
	Saving the DHCP snooping database .....	10
	Showing the database .....	11
	Completely removing the DHCP snooping database .....	12
	Max-bindings .....	12
	Static binding .....	13
	DHCP snooping ACL usage in AW+ .....	14
	ACL show command output .....	15
	Resource considerations .....	20
	DHCP snooping configuration examples .....	23
	DHCP snooping with filtering and Option 82, while acting as a Layer 2 switch .....	23
DHCP snooping and DHCP relay .....	25	
DHCP snooping with ARP security .....	27	
DHCP snooping with SNMP monitoring .....	31	
Troubleshooting DHCP snooping in AW+ .....	33	
Debugging .....	34	
ACL debugging .....	35	
DHCP snooping binding database debugging .....	39	
DHCP snooping packet debugging .....	40	
ARP security debugging .....	41	
Recommendations for configuring DHCP snooping/ARP security .....	43	

## Related How To Notes

You also may find the following How To Notes useful:

- How To Use MAC-Forced Forwarding with DHCP Snooping to Create Enhanced Private VLANs  
[http://www.alliedtelesis.com/media/datasheets/howto/macff\\_w-dhcp\\_vlans\\_sd\\_b.pdf](http://www.alliedtelesis.com/media/datasheets/howto/macff_w-dhcp_vlans_sd_b.pdf)
- How to Use DHCP Snooping and ARP Security to Block ARP Poisoning Attacks  
[http://www.alliedtelesis.com/media/datasheets/howto/config\\_arp\\_security\\_sd\\_a.pdf](http://www.alliedtelesis.com/media/datasheets/howto/config_arp_security_sd_a.pdf)
- How To Configure Policy-based Routing  
[http://www.alliedtelesis.com/media/datasheets/howto/howto\\_aw+\\_config\\_policybased\\_routing.pdf](http://www.alliedtelesis.com/media/datasheets/howto/howto_aw+_config_policybased_routing.pdf)

## Which products and software version does it apply to?

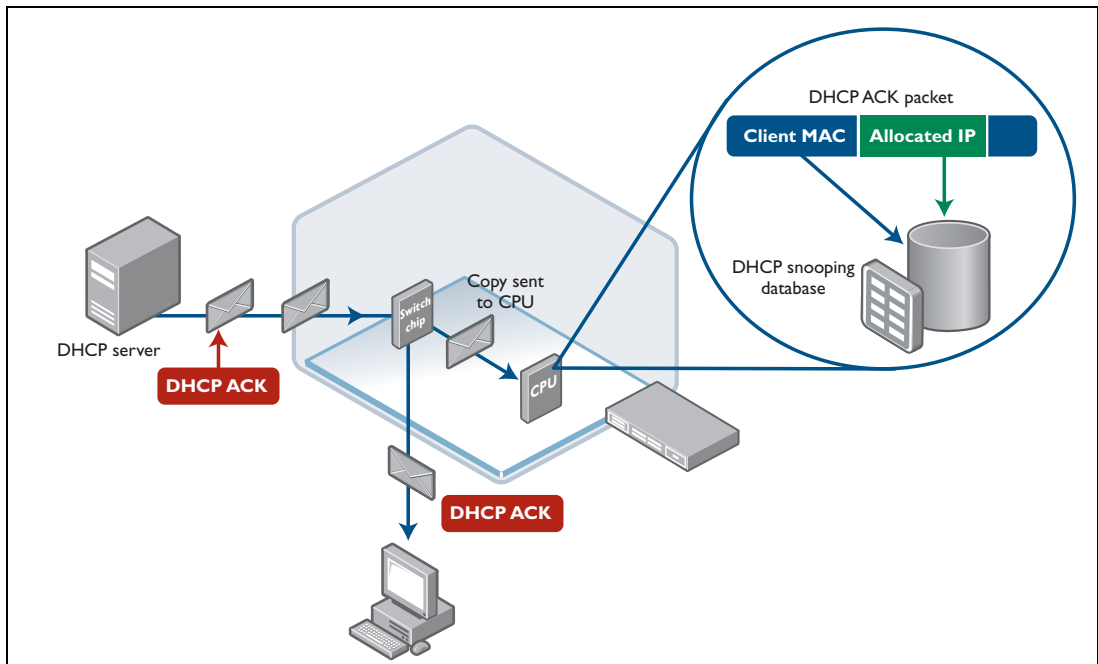
This How To Note applies to the following Allied Telesis managed Layer 3 switches:

- x600 series switches
- x900 series switches
- SwitchBlade™ x908 switch

It requires AlliedWare Plus™ (AW+) software version 5.3.4 or later

## DHCP snooping

DHCP snooping forces all DHCP packets to be sent up to the switch CPU before forwarding. The switch CPU is then able to look into the DHCP packets to see which IP addresses are being allocated to which clients. From this information, the switch maintains a database of the IP addresses that are currently allocated to downstream clients and the switch ports that the relevant clients are attached to.



**Note:** The DHCP snooping switch does not store a history log, the DHCP server does this.

DHCP snooping performs the following main tasks:

- Keeping a record of which IP addresses are currently allocated to hosts downstream of the ports on the switch.
- Deciding which packets are candidates for having Option 82 information inserted, and actively filtering out packets that are deemed to be invalid DHCP packets (according to criteria described below).

**Note:** Option 82 is enabled by default.

## AW+ DHCP snooping features

AW+ DHCP snooping features include:

- If there has been a violation with either ARP security or DHCP snooping:
  - the port can be disabled.
  - a trap can be sent to an SNMP management station.
- DHCP snooping can be disabled on a per VLAN basis.
- DHCP snooping can be configured on aggregated links (Static Channels, LACP).
- The database timer is per entry. Entries expire when the time left to expiry, for the entry, is 0 seconds.
- When a port goes down, DHCP snooping database entries learnt on that port can be deleted (this is configurable, and off by default).
- The switch can delete a DHCP snooping database entry when a DHCP release packet is received (this feature is configurable and enabled by default).
- DHCP snooping can be configured to drop packets if the source MAC and the client hardware address differ. This operation is enabled by default. It is disabled by the command:

```
awplus(config)#no ip dhcp snooping verify mac-address
```

---

**Note:** Do not use this feature in the situation when the DHCP Snooper is upstream of a DHCP-Relay, since the source MAC address of the packet does not match the client hardware address in the DHCP packets once they have been through a DHCP-Relay.

---

## Minimum configuration

On the SwitchBlade x908 switch, and the x600 and x900 series switches, the minimum configuration required to use DHCP snooping and provide unfiltered connectivity is provided below. With this configuration a client will be able to receive a DHCP address, and access the IP network unfiltered. Also, the administrator will be able to see the current valid entries in the DHCP snooping database.

**Note:** With this configuration, a client could manually change its IP and MAC address and be able to access the IP network unfiltered.

```
awplus(config)#service dhcp-snooping
awplus(config)#vlansdatabase
awplus(config-vlan)#vlan 50 name dhcpsnooping-clients
```

```
awplus(config)#int port1.0.1-port1.0.24
awplus(config-if)#switchport mode access
awplus(config-if)#switchport access vlan 50
```

On an x900 series switch, set port 24 (1.0.24), which has the DHCP server attached to it, to be the trusted port:

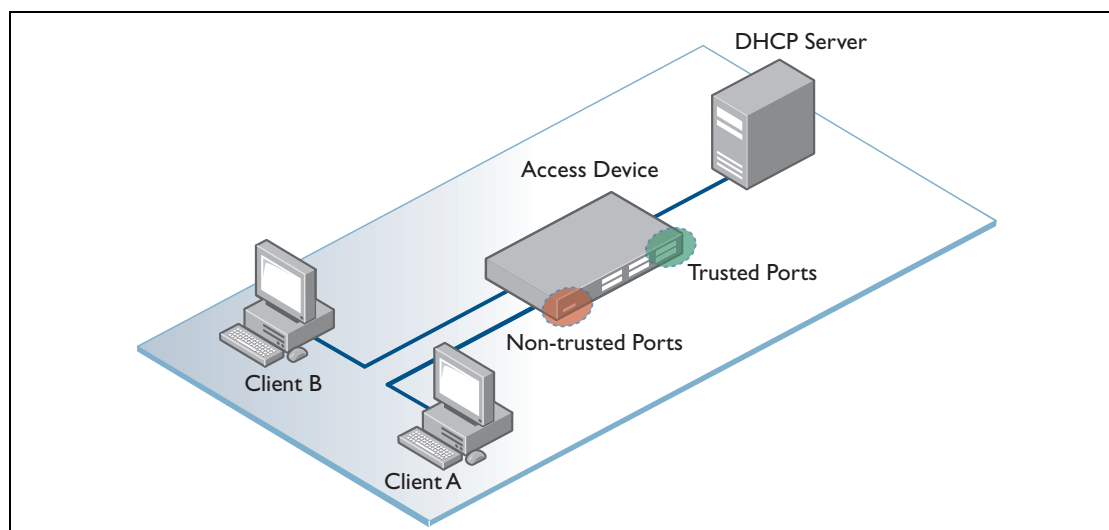
```
awplus(config)#int port1.0.24
awplus(config-if)#ip dhcp snooping trust
```

Configure the ports 1-23 for DHCP snooping and allocate them one client per port:

```
awplus(config)#int port1.0.1-port1.0.23
awplus(config-if)#ip dhcp snooping max-bindings 1
```

On the VLAN itself, enable DHCP snooping:

```
awplus(config)#int vlan50
awplus(config-if)#ip address 192.168.50.254/24
awplus(config-if)#ip dhcp snooping
```



## Configuring Option 82

DHCP Option 82 is enabled by default when DHCP snooping has been enabled. When the command **ip dhcp snooping agent-option** is enabled, the switch: inserts DHCP Option 82 into DHCP packets that it receives on untrusted ports, and removes DHCP Option 82 from DHCP packets that it sends to untrusted ports.

The Option 82 **enable** command is:

```
awplus(config)#ip dhcp snooping agent-option
```

The Option 82 **disable** command is:

```
awplus(config)#no ip dhcp snooping agent-option
```

The subscriber ID to be used on any given port can be configured with:

```
awplus(config)#interface port1.0.3
awplus(config-if)#ip dhcp snooping subscriber-id myid
```

The switch can also be configured to allow DHCP Option 82 reception on **untrusted ports**. When this is enabled, the switch accepts incoming DHCP packets that contain DHCP Option 82 data on untrusted ports.

```
awplus(config-if)#ip dhcp snooping agent-option allow-untrusted
```

This is used in circumstances where the DHCP snooping switch is upstream of another device that is inserting DHCP Option 82 information into DHCP packets.

## ARP security

It is also possible to enable DHCP snooping ARP security. If enabled, this will ensure that ARP packets received on untrusted ports are only permitted if they originate from an IP and MAC address from the DHCP snooping database.

► To enable DHCP snooping ARP security:

```
awplus(config)#int vlan1
awplus(config)#arp security
```

## DHCP filtering

The purpose of DHCP snooping-based filtering is to prevent IP addresses from being falsified or 'spoofed'. This guarantees that customers cannot avoid detection by spoofing an IP address that was not actually allocated to them. DHCP snooping-based filtering is achieved by creating specialised ACLs. The dynamic ACLs are configured with **dhcpsnooping** as a placeholder for the source IP address to match on.

The dynamic ACLs are attached to filters, which are applied to a port. Only those packets with a source IP address that matches one of the IP addresses that DHCP has allocated to the devices connected to that port are allowed through.

```
awplus(config)#access-list hardware acl1
awplus(config-ip-hw-acl)#permit ip dhcpsnooping any
awplus(config-ip-hw-acl)#deny ip any any
```

There are two options for the method by which the ACL can be applied to the untrusted ports: via a QoS policy-map, or directly as an interface ACL.

### Option 1: QoS configuration

#### 1. Enable QoS on the switch.

```
awplus(config)#mls qos enable
```

#### 2. Create the class-map.

```
awplus(config)#class-map 1
```

#### 3. Match the ACL group (acl1).

```
awplus(config-cmap)#match access-group acl1
```

#### 4. Add the class-map to the policy-map.

```
awplus(config)#policy-map 1
awplus(config-pmap)#class 1
```

#### 5. Apply the policy-map to all the ports on the range 1-23.

```
awplus(config)#int port1.0.1-1.0.23
awplus(config-if)#service-policy input 1
```

### Option 2: Interface ACL

#### ► Create the interface ACL.

```
awplus(config)#int port1.0.1-1.0.23
awplus(config-if)#access-group acl1
```

Do not add a **dhcpsnooping** ACL globally\* on the switch, not all ports are going to be client ports. Trying to configure this will result in an error similar to the one shown below:

```
awplus(config)#access-group acl1
% DHCP Snooping filters cannot be applied globally. Filters skipped
```

\* Global ACLs are not supported on the x600 series switches.

## DHCP filtering when multiple VLANs are configured on the untrusted port

In this scenario, the untrusted ports are configured in multiple VLANs.

The ports are configured to only accept packets that match entries in the DHCP snooping database. Moreover, for each port, there is a limit on the data rate that can be accepted in each VLAN. A user may send data in on VLAN2 at a certain rate, and at a different rate on VLAN3, and another rate on VLAN4. This corresponds to three different network services that each have their own bandwidth limit. This is shown in the example configuration below:

```
mls qos enable
access-list hardware drop
  deny ip any any
access-list hardware vlan2snoop
  permit ip dhcpsnooping any vlan 2
access-list hardware vlan3snoop
  permit ip dhcpsnooping any vlan 3
access-list hardware vlan4snoop
  permit ip dhcpsnooping any vlan 4

class-map vlan2snoop
  match access-group vlan2snoop
!
class-map vlan3snoop
  match access-group vlan3snoop
!
class-map vlan4snoop
  match access-group vlan4snoop
!
class-map drop
  match access-group drop
!
policy-map servicerates
  class default
  class vlan2snoop
    police single-rate 648 20000 80000 action drop-red
  class vlan3snoop
    police single-rate 1296 80000 160000 action drop-red
  class vlan4snoop
    police single-rate 648 20000 80000 action drop-red
  class drop
[cont...]
```

```

vlan database
  vlan 2-4 state enable
!
interface port1.0.1-1.0.24
  switchport mode trunk
  switchport trunk allowed vlan add 2-4
  switchport trunk native vlan none
  ip dhcp snooping max-bindings 3
  service-policy input servicerates

```

## The DHCP snooping database

---

The switch watches the DHCP packets that it is passing back-and-forth. It thereby maintains a database that lists the DHCP leases it knows are being held by devices downstream of its ports.

### Saving the DHCP snooping database

You can save the DHCP snooping backup database in three different locations:

- NVS (the default)
- Flash
- SD card

---

**Note:** Care needs to be taken if you specify the storage area to be the SD card option. A current DHCP snooping backup database file is essential to maintain connectivity for DHCP clients after a switch reboot. If you configure the switch to save this file to an SD card, we recommend that you ensure the card is always present.

---

The command to specify the location is:

```
awplus(config)#ip dhcp snooping database {nvs|flash|card}
```

Entering the command **ip dhcp snooping database ?** displays these options:

```

awplus#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
awplus(config)#ip dhcp snooping database ?
  card  Store the database in a SD card
  flash Store the database in flash
  nvs   Store the database in nvs

```

## Showing the database

To verify the status of snooped users, use the **show ip dhcp snooping binding** command.

```
awplus#sh ip dhcp snooping binding
DHCP Snooping Bindings:

Client IP          MAC Address        Server IP          VLAN  Port    Expiry(s)  Type
-----
192.168.10.1      00e0.18b3.d1a6    192.168.10.253   10   1.0.19  3207       Dyna

Total number of bindings in database: 1
```

A description of the terms used in the command output above follows:

Term	Description
<b>Client IP</b>	The IP address that has been allocated to the snooped DHCP client.
<b>MAC Address</b>	The MAC address of the snooped DHCP client.
<b>Server IP</b>	The IP address of the DHCP Server.
<b>VLAN</b>	The VLAN to which the snooped DHCP client is connected.
<b>Port</b>	The port to which the snooped DHCP client is connected.
<b>Expires</b>	The time, in seconds, until the DHCP client entry will expire.
<b>Type</b>	How the DHCP binding was entered into the database The source of the entry: <ul style="list-style-type: none"> <li>• <b>Dyna</b>: dynamically entered by snooping DHCP traffic, or configured by the ip dhcp snooping binding command.</li> <li>• <b>Static</b>: added statically by the ip source binding command.</li> <li>• <b>File</b>: reloaded into the current active database from the DHCP snooping database backup file.</li> </ul>
<b>Total number of bindings in database</b>	The total number of dynamic and static lease entries in the database

## Completely removing the DHCP snooping database

To completely remove the DHCP snooping database, delete the file: **nvs:/.dhcp.dsn.gz**

```
awplus#del nvs:/.dhcp.dsn.gz
Delete nvs:/.dhcp.dsn.gz? (y/n) [n]:y
Deleting..
Successful operation
```

To check that the database is empty, use the command **show ip dhcp snooping binding**, you will see an output similar to the one provided below:

```
awplus#show ip dhcp snooping binding
DHCP Snooping Bindings:

Client IP          MAC Address        Server IP          VLAN  Port    Expiry(s)  Type
-----
No entries in database
```

## Max-bindings

To configure how many times the dhcpsnooping filters on a port will be replicated:

```
awplus(config)#int port1.0.18
```

```
awplus(config-if)#ip dhcp snooping max-bindings <number of clients
that will be attached to this port>
```

Any filter that is using a classifier containing a **dhcpsnooping** parameter, and is applied to a port in the list, will be replicated **max-bindings** times as it is written into the hardware table. Then, as the IP addresses are allocated to devices on the port, the each newly learnt address can be written in one of these replicated filter entries.

For example, when the first device on the port receives a lease, the first member of the relevant set of replicated filters is filled in with the lease address. When a second device on the port receives a lease, the second member of the set of replicated filters receives the new lease address, and so on.

Be aware that if the number of client devices downstream of a port is greater than the max-bindings configured on the port, then once the placeholder source IP address in all the **dhcpsnooping** ALCs attached to that port have been filled in with the addresses leased to clients, any other clients will be blocked from accessing the network, even if they do receive a valid DHCP lease.

## Static binding

If there is a device with a statically configured IP address attached to a port in the DHCP snooping port range, then, with filtering enabled it is necessary to statically bind it to the port. This will ensure the device's IP connectivity to the rest of the network.

If a device with the IP address of 192.16.1.202 and MAC address of 00-00-00-00-00-ca is attached to VLAN 1 on port 2, then a static binding is configured by adding the following command to the basic DHCP configuration (see "[DHCP filtering](#)" on page 8).

```
awplus#ip dhcp snooping binding 172.16.1.202 0000.0000.00CA vlan 1
      interface port1.0.2 expiry 3600
```

Adding a static binding uses a lease on the port. If the maximum leases on the port is 1 (the default), the static binding means that no device on the port can acquire an address by DHCP.

## DHCP snooping ACL usage in AW+

---

There are a set of **show** commands that enable you to see the Access Control List (ACL) usage of DHCP snooping. Using these commands is an excellent way of troubleshooting DHCP snooping issues and ACL consumption, especially on the x600 switches:

```
awplus#show ip dhcp snooping acl
awplus#show ip dhcp snooping acl hardware
awplus#show ip dhcp snooping acl detail
```

All these commands are very useful when seeing which interfaces have which ACLs attached and how many hardware filter table entries each interface is consuming. Example output for each of these commands is provided starting on [page 15](#).

The configuration on the x600 switch which was used with these **show** commands is as follows:

```
snmp-server enable trap dhcpsnooping
snmp-server community public rw
snmp-server host 192.168.10.252 version 2c public
!
mls qos enable
access-list hardware acl1
permit ip dhcpsnooping any
deny ip any any
!
vlan database
  vlan 10 name dhcp-clients
!
interface port1.0.1-1.0.2
  switchport access vlan 10
  ip dhcp snooping trust
!
interface port1.0.5
  ip dhcp snooping max-bindings 1
  access-group acl1
!
interface port1.0.10-1.0.19
  switchport access vlan 10
  ip dhcp snooping max-bindings 5
!
interface port1.0.20
  switchport access vlan 10
  ip dhcp snooping max-bindings 10
  access-group acl1
  ip dhcp snooping violation trap
!
interface vlan10
  ip address 192.168.10.254/24
  ip dhcp snooping
```

## ACL show command output

Let's take a look at some sample output for each of the DHCP snooping **acl** show commands now:

### show ip dhcp snooping acl

```
awplus#show ip dhcp snooping acl
```

DHCP Snooping Based Filters Summary:

Interface	Bindings	Maximum Bindings	Template Filters	Attached Hardware Filters
port1.0.1	0	1	0	0
port1.0.2	0	1	0	0
port1.0.3	0	1	0	0
port1.0.4	0	1	0	0
<b>port1.0.5</b>	<b>0</b>	<b>1</b>	<b>1</b>	<b>1</b>
port1.0.6	0	1	0	0
port1.0.7	0	1	0	0
port1.0.8	0	1	0	0
port1.0.9	0	1	0	0
port1.0.10	0	<b>5</b>	0	0
port1.0.11	0	<b>5</b>	0	0
port1.0.12	0	<b>5</b>	0	0
port1.0.13	0	<b>5</b>	0	0
port1.0.14	0	<b>5</b>	0	0
port1.0.15	0	<b>5</b>	0	0
port1.0.16	0	<b>5</b>	0	0
port1.0.17	0	<b>5</b>	0	0
port1.0.18	0	<b>5</b>	0	0
port1.0.19	0	<b>5</b>	0	0
<b>port1.0.20</b>	<b>1</b>	<b>10</b>	<b>1</b>	<b>10</b>
port1.0.21	0	1	0	0
port1.0.22	0	1	0	0
port1.0.23	0	1	0	0
port1.0.24	0	1	0	0

Total Attached Hardware Filters: 11

**Note:** The hardware table usage for DHCP snooping ACLs on a given port is found by multiplying the max-bindings for a port by the number of DHCP snooping ACLs applied to the port.

A description for each of the columns in the **show ip dhcp snooping acl** output is provided in the table below:

### Column name descriptions

Column Name	Description
<b>Interface</b>	Port number.
<b>Bindings</b>	Number of entries in the DHCP snooping database that are associated with this port. In effect, it is the number of DHCP clients downstream of the port that have successfully received DHCP leases.
<b>Maximum Bindings</b>	Maximum number of DHCP snooping database entries that can be associated with the port - i.e. the maximum number of DHCP clients that can be downstream of this port and get DHCP leases.
<b>Template Filters</b>	Number of ACLs that have been configured on this port that use the <b>dhcpsnooping</b> keyword. They are referred to as template filters because when they are created, the source address that is configured to match on <b>dhcpsnooping</b> will not match an actual address. The filters are filled in with actual addresses to match on as DHCP snooping database entries are created that are associated with the port that the ACL is attached to.
<b>Attached Hardware Filters</b>	The actual number of entries in the hardware filter table that have been created to hold DHCP snooping ACLs attached to this port.

### Table output summary

The example output from the **show ip dhcp snooping acl** command shows that:

- On interface port1.0.5, we have attached an ACL and this has resulted in the creation of one hardware filter, because the max-binding on this port is 1.
- On the interfaces port1.0.10 - port1.0.19, we have increased the max-bindings on these ports to 5 but there aren't any ACLs on these interfaces yet and so no hardware filters created.
- Interface port1.0.20 has had the max-bindings increased to 10 and has a 'DHCP snooping' ACL attached to it. The reason why there are 10 hardware table entries used for this interface is because the max-bindings has been configured as 10, so up to 10 different clients can receive DHCP leases downstream of this port. For the ACL to operate correctly for all 10 clients, it is necessary to have 10 instances of the ACL created in hardware. Each of these can have the "DHCP snooping" placeholder in the ACL filled in with a client's DHCP-allocated IP address.

One client downstream of port1.0.20 has received an IP address from the DHCP Server. This has resulted in an entry being created in the x600's **DHCP snooping binding** database:

**awplus#show ip dhcp snooping binding**

DHCP Snooping Bindings:

Client IP	MAC Address	Server IP	VLAN	Port	Expiry(s)	Type
192.168.10.50	00e0.18b3.d1a6	192.168.10.254	10	<b>1.0.20</b>	2248	Dyna

**show ip dhcp snooping acl hardware****awplus#show ip dhcp snooping acl hardware**

DHCP Snooping Based Filters in Hardware:

Interface	Access-list(/ClassMap)	Source IP	Source MAC
port1.0.5	acl1	0.0.0.0	
<b>port1.0.20</b>	<b>acl1</b>	<b>192.168.10.50</b>	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	
port1.0.20	acl1	0.0.0.0	

**Table output summary**

This command shows which ACLs are attached to which ports and the actual values that are currently assigned to the DHCP snooping placeholder addresses in the ACLs.

The source IP and source MAC columns show what values are currently written into the source IP and source MAC fields of the hardware table entries. Most of the entries have 0.0.0.0 in the source IP field. This is because when a hardware filter entry is first created for an ACL with **dhcpsnooping** in place of an IP address, the value 0.0.0.0 is written into that IP field of the entry, as a place holder value. Then when a client has been issued a valid IP address and it has been learnt (snooped), that client's address will overwrite the 0.0.0.0 in one of the HW table entries created for that ACL on that port.

The example above shows that:

- Ten instances of acl1 attached to port1.0.20, which matches the max-binding of 10 on that port (as shown in the table output on [page 15](#)).
- Address 192.168.10.50 has been learnt by a client downstream of port1.01.20, so this address has been written into the **Source IP** field in one of the HW table entries associated with port1.0.20.
- Source MAC field of every entry is empty because none of the ACLs are configured to match on the MAC address.

**show ip dhcp snooping acl detail**

```

awplus#show ip dhcp snooping acl detail

DHCP Snooping Based Filters Information:

port1.0.1 : Maximum Bindings ..... 1
port1.0.1 : Template filters ..... 0
port1.0.1 : Attached hardware filters .. 0
port1.0.1 : Current bindings ..... 0, 1 free

port1.0.2 : Maximum Bindings ..... 1
port1.0.2 : Template filters ..... 0
port1.0.2 : Attached hardware filters .. 0
port1.0.2 : Current bindings ..... 0, 1 free

port1.0.3 : Maximum Bindings ..... 1
port1.0.3 : Template filters ..... 0
port1.0.3 : Attached hardware filters .. 0
port1.0.3 : Current bindings ..... 0, 1 free

port1.0.4 : Maximum Bindings ..... 1
port1.0.4 : Template filters ..... 0
port1.0.4 : Attached hardware filters .. 0
port1.0.4 : Current bindings ..... 0, 1 free

port1.0.5 : Maximum Bindings ..... 1
port1.0.5 : Template filters ..... 1
port1.0.5 : Attached hardware filters .. 1
port1.0.5 : Current bindings ..... 0, 1 free

port1.0.6 : Maximum Bindings ..... 1
port1.0.6 : Template filters ..... 0
port1.0.6 : Attached hardware filters .. 0
port1.0.6 : Current bindings ..... 0, 1 free

port1.0.7 : Maximum Bindings ..... 1
port1.0.7 : Template filters ..... 0
port1.0.7 : Attached hardware filters .. 0
port1.0.7 : Current bindings ..... 0, 1 free

port1.0.8 : Maximum Bindings ..... 1
port1.0.8 : Template filters ..... 0
port1.0.8 : Attached hardware filters .. 0
port1.0.8 : Current bindings ..... 0, 1 free
port1.0.9 : Maximum Bindings ..... 1
port1.0.9 : Template filters ..... 0
port1.0.9 : Attached hardware filters .. 0
port1.0.9 : Current bindings ..... 0, 1 free

port1.0.10 : Maximum Bindings ..... 5
port1.0.10 : Template filters ..... 0
port1.0.10 : Attached hardware filters .. 0
port1.0.10 : Current bindings ..... 0, 5 free

port1.0.11 : Maximum Bindings ..... 5
port1.0.11 : Template filters ..... 0
port1.0.11 : Attached hardware filters .. 0
port1.0.11 : Current bindings ..... 0, 5 free

[cont...]

```

```

port1.0.12 : Maximum Bindings ..... 5
port1.0.12 : Template filters ..... 0
port1.0.12 : Attached hardware filters .. 0
port1.0.12 : Current bindings ..... 0, 5 free

port1.0.13 : Maximum Bindings ..... 5
port1.0.13 : Template filters ..... 0
port1.0.13 : Attached hardware filters .. 0
port1.0.13 : Current bindings ..... 0, 5 free

port1.0.14 : Maximum Bindings ..... 5
port1.0.14 : Template filters ..... 0
port1.0.14 : Attached hardware filters .. 0
port1.0.14 : Current bindings ..... 0, 5 free

port1.0.15 : Maximum Bindings ..... 5
port1.0.15 : Template filters ..... 0
port1.0.15 : Attached hardware filters .. 0
port1.0.15 : Current bindings ..... 0, 5 free

port1.0.16 : Maximum Bindings ..... 5
port1.0.16 : Template filters ..... 0
port1.0.16 : Attached hardware filters .. 0
port1.0.16 : Current bindings ..... 0, 5 free

port1.0.17 : Maximum Bindings ..... 5
port1.0.17 : Template filters ..... 0
port1.0.17 : Attached hardware filters .. 0
port1.0.17 : Current bindings ..... 0, 5 free

port1.0.18 : Maximum Bindings ..... 5
port1.0.18 : Template filters ..... 0
port1.0.18 : Attached hardware filters .. 0
port1.0.18 : Current bindings ..... 0, 5 free

port1.0.19 : Maximum Bindings ..... 5
port1.0.19 : Template filters ..... 0
port1.0.19 : Attached hardware filters .. 0
port1.0.19 : Current bindings ..... 0, 5 free

port1.0.20 : Maximum Bindings ..... 10
port1.0.20 : Template filters ..... 1
port1.0.20 : Attached hardware filters .. 10
port1.0.20 : Current bindings ..... 1, 9 free
port1.0.20 Client 1 ..... 192.168.10.50 00e0.18b3.d1a6
port1.0.20 : Template: acl1 (via access-group)
port1.0.20 : 10 permit ip dhcpsnooping any
port1.0.21 : Maximum Bindings ..... 1
port1.0.21 : Template filters ..... 0
port1.0.21 : Attached hardware filters .. 0
port1.0.21 : Current bindings ..... 0, 1 free

port1.0.22 : Maximum Bindings ..... 1
port1.0.22 : Template filters ..... 0
port1.0.22 : Attached hardware filters .. 0
port1.0.22 : Current bindings ..... 0, 1 free
[cont...]

```

```

port1.0.23 : Maximum Bindings ..... 1
port1.0.23 : Template filters ..... 0
port1.0.23 : Attached hardware filters ... 0
port1.0.23 : Current bindings ..... 0, 1 free

port1.0.24 : Maximum Bindings ..... 1
port1.0.24 : Template filters ..... 0
port1.0.24 : Attached hardware filters ... 0
port1.0.24 : Current bindings ..... 0, 1 free

```

### Table output summary

As the command syntax implies (**show ip dhcp snooping acl detail**), this gives the details of every port on the switch. It will also show the IP address - MAC address pair of any of the clients that have been successful in obtaining an address from the DHCP server and which ACLs have been attached to those ports.

## Resource considerations

Because of the potential for classifier replication when configuring DHCP ACLs, you need to be cautious about running out of hardware filter resource, especially when using a complex QoS classifier solution.

---

**Note:** Using the commands **arp security** and **verify mac-address** will have no affect on hardware filter resources.

---

### Calculation

To calculate the number of hardware rules used:

number of ACL's \* number of ports attached \*dhcpsn max-binding per port.

To view these statistics, use the command: **show platform classifier statistics utilization brief** on both the x900/x600 series switches. An example of these outputs follow:

Taken from an  
x900

```

awplus#show platform classifier statistics utilization brief

[Instance 0]
[port1.0.1-1.0.12]
Number of PCE Entries:
Used / Total

-----
DHCP Snooping      2
Global ACL         0
ACL                0
QoS                0
Total              2 / 2048 ( 0%)

Profiles:
Legend of Offset Type) 1:Ether 2:IP 3:TCP/UDP
Packet Type  Offset Type      Used / Total
----- 0-----8-----15 -----
TCP (IPv4)    0000000000000000    0 / 16
UDP (IPv4)    2330000000000000    3 / 16
IPv4 fragment 2330000000000000    3 / 16
IPv4 other    0000000000000000    0 / 16
Ethernet      0000000000000000    0 / 16
IPv6          0000000000000000    0 / 16

[Instance 2]
[port1.0.13-1.0.24]
Number of PCE Entries:
Used / Total

-----
DHCP Snooping      2
Global ACL         0
ACL                0
QoS                0
Total              2 / 2048 ( 0%)

Profiles:
Legend of Offset Type) 1:Ether 2:IP 3:TCP/UDP
Packet Type  Offset Type      Used / Total
----- 0-----8-----15 -----
TCP (IPv4)    0000000000000000    0 / 16
UDP (IPv4)    2330000000000000    3 / 16
IPv4 fragment 2330000000000000    3 / 16
IPv4 other    0000000000000000    0 / 16
Ethernet      0000000000000000    0 / 16
IPv6          0000000000000000    0 / 16
    
```

**Taken from an  
x600**

```

[Instance 4]
Number of Entries:
Policy Type      Group ID      Used / Total
-----
ACL              1476395009   0 / 121 ( 0%)
Web Auth        1476395010   11 / 128 ( 8%)
DoS             Inactive      0 / 0 ( 0%)
VLAN Counter
  Group-Octet    Inactive      0 / 0 ( 0%)
  Group-Packet   Inactive      0 / 0 ( 0%)
QoS              0 / 768 ( 0%)

[Instance 16]
Number of Entries:
Policy Type      Group ID      Used / Total
-----
ACL              1476395009   0 / 121 ( 0%)
Web Auth        1476395010   11 / 128 ( 8%)
DoS             Inactive      0 / 0 ( 0%)
VLAN Counter
  Group-Octet    Inactive      0 / 0 ( 0%)
  Group-Packet   Inactive      0 / 0 ( 0%)
QoS              0 / 768 ( 0%)

```

## DHCP snooping configuration examples

Let us take a look at four configuration examples for DHCP snooping:

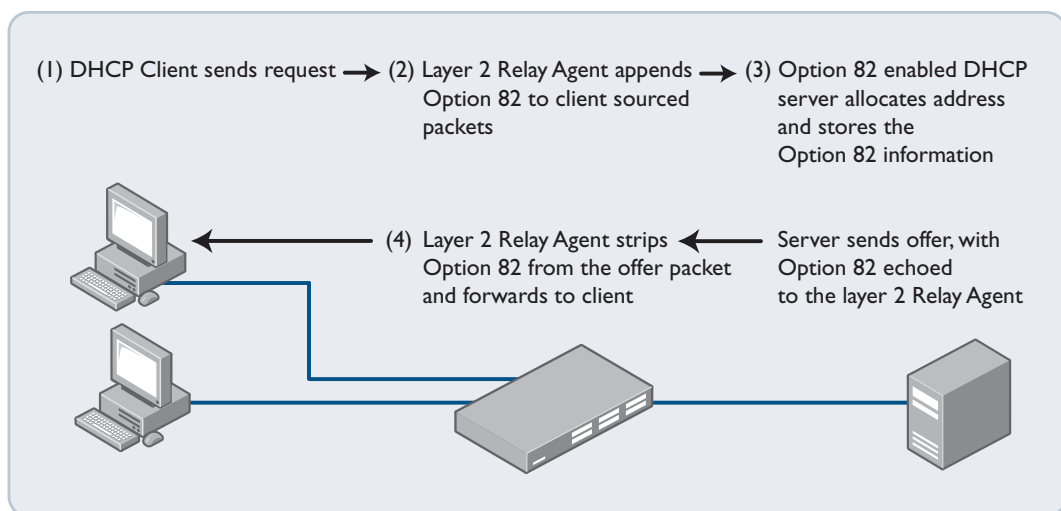
1. Configuring the switch for DHCP snooping, filtering, and Option 82, when it is acting as a Layer 2 switch. On [page 23](#)
2. A Layer 2 switch configured with DHCP snooping, and a Layer 3 switch with DHCP relay configured, connected to the uplink port of the Layer 2 switch. On [page 25](#)
3. DHCP snooping with ARP security. On [page 27](#)
4. SNMP monitoring DHCP snooping. This example includes the traps for both DHCP snooping and the ARP security feature. On [page 31](#)

### DHCP snooping with filtering and Option 82, while acting as a Layer 2 switch

In a Layer 2 switching environment, a switch configured with Option 82 snooping will snoop any client-originated DHCP packets and insert Option 82 information into it before forwarding the packet(s) to the DHCP server. In this sense, it is a Layer 2 relay agent; the packet source and destination addresses are not altered. This is, of course, quite different to a Layer 3 DHCP relay, as is enabled by **service DHCP-relay**.

DHCP servers that are configured to recognise the relay agent information option (Option 82), may use the information to keep a log of switches and port numbers that IP addresses have been allocated to, and may also use the information for various address assignment policies.

The DHCP server echoes the option back verbatim to the relay agent in server-to-client replies, and the relay agent strips the option before forwarding the reply to the client. This process is shown in the following figure.



The configuration steps are as follows:

► Add the tagged uplink ports to the VLAN.

```
awplus(config)#int port1.0.24
awplus(config)#switchport mode trunk
awplus(config)#switchport trunk allowed VLAN add 48
```

► Configure the VLAN for customers.

```
awplus(config)#int port1.0.1-1.0.23
awplus(config-if)#switchport access vlan 48
```

► Enable DHCP snooping.

```
awplus(config)#service dhcp-snooping
```

Insertion of Option 82 information into snooped DHCP packets is enabled by default. It is also possible to enable DHCP snooping ARP security using the **arp security** command. If enabled, this will ensure that ARP packets received on non-trusted ports are only permitted if they originate from an IP or MAC address that has been allocated by a DHCP server, and snooped by DHCP snooping:

```
awplus(config)#interface <vlan-list>
awplus(config-if)#arp security
```

► Define the DHCP snooping trusted ports.

```
awplus(config)#int port1.0.24
awplus(config)#ip dhcp snooping trust
```

These ports can receive Option 82 information by default, and the switch will permit them to send Option 82.

► Define the maximum number of DHCP leases permitted on each port.

```
awplus(config-if)#interface port1.0.1-port1.0.23
awplus(config-if)#ip dhcp snooping max-bindings 1
```

► Define the string that will be used in the **subscriber-id** suboption portion of the Option 82 inserted into DHCP packets.

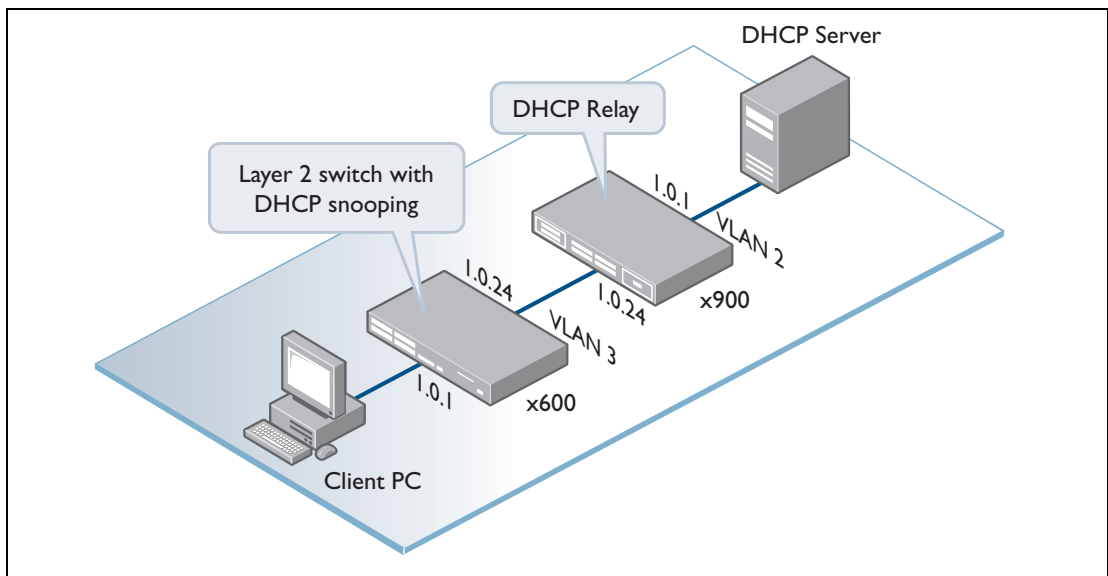
```
awplus(config-if)#ip dhcp snooping subscriber-id "Ground Floor
Room 1
```

- ▶ Create a filter to only allow packets from IP addresses that are in the DHCP snooping database, and apply that filter to the untrusted ports.

```
awplus(config-if-hw-acl)#permit ip dhcpsnooping any
awplus(config-if-hw-acl)#deny ip any any
awplus(config-if)#interface port1.0.1-port1.0.23
awplus(config-if)#access-group acl1
```

## DHCP snooping and DHCP relay

The second example has an x900 switch which is performing DHCP relay and an x600 switch which is acting as a Layer 2 switch with DHCP snooping configured.



This example also has the x600 adding Option 82 into the DHCP packets it snoops. In AW+, this is enabled by default, so there is no need for any extra configuration. The command to use to enable this feature, if this was disabled on the switch is:

```
awplus(config)#ip dhcp snooping agent-option
```

See "DHCP Option 82" on page 1.

**x600  
configuration**

```

service dhcp-snooping
vlan database
  vlan 3 name clients
  vlan 3 state enable
!
interface port1.0.1
switchport access vlan 3
  ip dhcp snooping max-bindings 5
!
interface port1.0.2-1.0.23
switchport access vlan 3
!
interface port1.0.24
switchport access vlan 3
  ip dhcp snooping trust
!
interface vlan3
  ip dhcp snooping

```

**x900  
configuration**

```

service dhcp-relay
vlan database
  vlan 2 name DHCP
  vlan 3 name Clients
  vlan 2-3 state enable
!
interface port1.0.1
switchport access vlan 2
!
interface port1.0.24
switchport access vlan 3
!
interface vlan2
  ip address 192.168.2.253/24
!
interface vlan3
  ip address 192.168.3.253/24
  ip dhcp-relay server-address 192.168.2.254

```

When a client connected to one of the ports of the snooping switch has received a DHCP lease, this can be seen in the DHCP snooping database.

```
awplus#sh ip dhcp snooping binding
```

```
DHCP Snooping Bindings:
```

```
Port: (1.0.12) indicates that port1.0.12 is provisioned
```

Client IP	MAC Address	Server IP	VLAN	Port	Expiry(s)	Type
192.168.3.1	0006.5b31.14af	192.168.2.254	3	1.0.1	2565	Dyna

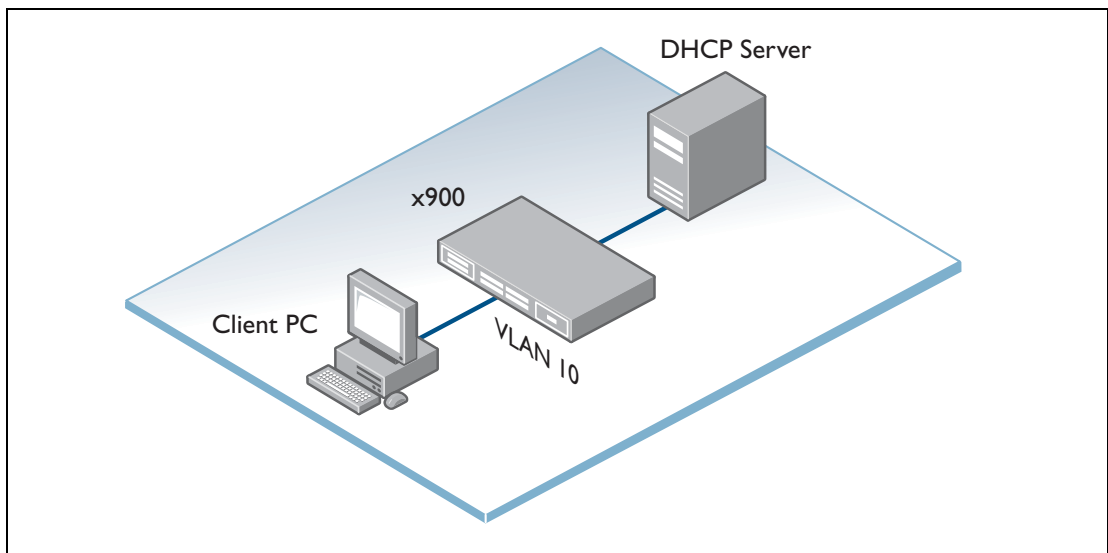
```
Total number of bindings in database: 1
```

## DHCP snooping with ARP security

The binding database created by DHCP snooping allows ARP packets to be checked and validated when received on untrusted ports. If the source of an ARP packet does not match an existing binding entry, the packet can be dropped and optionally logged, or the port can be shut down.

In the following example, we have clients that have been verified and added to the DHCP snooping binding database and we have a rogue device trying to gain access without getting authorised. The rogue device has the correct IP address for this network but the MAC address does not match the real MAC address of the host.

In this example, we will show the debugging and also the log messages generated when ARP security is enabled.



The x900 switch has VLAN 10 configured, with DHCP snooping enabled and ARP security enabled on VLAN 10.

The configuration for ARP security is not difficult, but, do note that ARP security needs to be enabled on the VLAN interface, not on the port itself.

```

snmp-server
snmp-server enable trap dhcpsnooping
snmp-server community public rw
snmp-server host 192.168.10.254 version 2c public
!
service dhcp-snooping
!
vlan database
  vlan 10 name clients
!
interface port1.0.1-1.0.2
switchport access vlan 10
ip dhcp snooping trust
!
interface port1.0.10-1.0.20
switchport access vlan 10
ip dhcp snooping max-bindings 3
ip dhcp snooping violation trap
arp security violation log trap

interface vlan10
ip address 192.168.10.250/24
ip dhcp snooping
arp security

```

The log below shows that we have sent in ARP packets that have a source MAC address/IP address that have not been verified and added to the DHCP snooping binding database. And since ARP security is enabled on VLAN 10, these packets are discarded.

```

awplus# debug arp security
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB., port1.0.20, vid
10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus Target: 0000.cd29.878d 192.168.10.250
13:57:16 awplus kernel: [ARPSEC 00126] ARP Discarded: No Binding
13:57:16 awplus kernel: [ARPSEC 00127] RXARPRequest on port1.0.20 (untrusted) vlan10
13:57:16 awplus kernel: [ARPSEC 00127] MAC: 0000.0a00.0000 > 0000.cd29.878d
13:57:16 awplus kernel: [ARPSEC 00127] ARP: Sender 0000.0a00.0000 192.168.10.20
13:57:16 awplus kernel: [ARPSEC 00127] ARP: Target 0000.cd29.878d 192.168.10.250

```

Below is a successful transaction from the DHCP server and the client with both ARP security and DHCP snooping enabled. We can see that since the client has not been added to the binding database, the x900 logs the broadcast DHCP packets, and then we have a successful transaction and the clients IP-MAC address has been added to the binding database:

```
awplus#
14:04:25awplusMSTP[1051]:CISTport1.0.19nowforwarding,propagatingTCtootherports
14:04:37 awplus DHCPSN[1799]: ARP Sec: ARP source IP not in snooping DB., port1.0.19, vid
10, Src 169.254.28.38, Mac 0006.5b31.14af
14:04:37 awplus DHCPSN[1799]: ARP Sec: ARP source IP not in snooping DB., port1.0.19, vid
10, Src 169.254.28.38, Mac 0006.5b31.14af
14:04:38 awplus DHCPSN[1799]: ARP Sec: ARP source IP not in snooping DB., port1.0.19, vid
10, Src 169.254.28.38, Mac 0006.5b31.14af
14:04:41 awplus DHCPSN[1799]: [PKT 00001] -----
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Received DHCP Snooping packet
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Received on VLAN 10(Ingress) IfIndex 5019
Length 342
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Processing BOOTP Request
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Rxd on untrusted port
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Option 82 added 14 bytes giving UDPLen:322
IPLen:342 PktLen:356
14:04:41 awplus DHCPSN[1799]: [PKT 00001] Sent DHCP Snooping packet for forwarding,
Length: 356
14:04:41 awplus DHCPSN[1799]: [PKT 00002] -----
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Received DHCP Snooping packet
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Received on VLAN 10(Ingress) IfIndex 5001
Length 590
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Processing BOOTP Reply
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Rxd on trusted port
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Client port: Found 5019 on VLAN 10
14:04:41 awplus DHCPSN[1799]: [PKT 00002] Sent DHCP Snooping packet for forwarding,
Length: 590
14:04:41 awplus DHCPSN[1799]: [PKT 00003] -----
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Received DHCP Snooping packet
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Received on VLAN 10(Ingress) IfIndex 5019
Length 348
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Processing BOOTP Request
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Rxd on untrusted port
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Option 82 added 22 bytes giving UDPLen:336
IPLen:356 PktLen:370
14:04:41 awplus DHCPSN[1799]: [PKT 00003] Sent DHCP Snooping packet for forwarding,
Length: 370
14:04:41 awplus DHCPSN[1799]: [PKT 00004] -----
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Received DHCP Snooping packet
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Received on VLAN 10(Ingress) IfIndex 5001
Length 590
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Processing BOOTP Reply
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Rxd on trusted port
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Client port: Found 5019 on VLAN 10
14:04:41 awplus DHCPSN[1799]: [PKT 00004] DHCP ACK Found
14:04:41 awplus DHCPSN[1799]: Binding Add: 192.168.10.1, chaddr 0006.5b31.14af, vlan10,
port1.0.19, Server 192.168.10.253, Type Dynamic, Expires
14:04:41 awplus DHCPSN[1799]: [PKT 00004] Sent DHCP Snooping packet for forwarding,
Length: 590
```

DHCP Snooping Bindings:

Client IP	MAC Address	Server IP	VLAN	Port	Expiry(s)	Type
192.168.10.1	0006.5b31.14af	192.168.10.253	10	1.0.19	3586	Dyna

In the initial configuration above, ports 1.0.10-1.0.20 had been configured just to generate ARP and SNMP trap security violations, with the command:

```
awplus(config-if)#arp security violation log trap
```

But, it is possible to take a more severe action when a violation is detected—namely to shut the port down. Violation descriptions are provided on [page 32](#).

This is achieved by changing the configuration of the violation action:

```
awplus(config)#int port1.0.10
awplus(config-if)#arp security violation link-down
```

The example below shows that when the violation has occurred, ARP security takes that port down.

Initially, we see that port 1.0.10 is up:

```
awplus(config-if)#
awplus#sho int brief
Interface          Status          Protocol
port1.0.1          admin up       running
port1.0.2          admin up       down
port1.0.3          admin up       down
port1.0.4          admin up       down
port1.0.5          admin up       down
port1.0.6          admin up       down
port1.0.7          admin up       down
port1.0.8          admin up       down
port1.0.9          admin up       down
port1.0.10        admin up       running
port1.0.11         admin up       down
port1.9.12         admin up       down
port1.0.13         admin up       down
port1.0.14         admin up       down
port1.0.15         admin up       down
port1.0.16         admin up       down
port1.0.17         admin up       down
port1.0.18         admin up       down
port1.0.19         admin up       down
port1.0.20         admin up       down
```

If you send in an ARP request that breaches the ARP security settings that have been configured, a set of log messages will be produced, as shown below:

```
awplus#sho log tail
```

```
<date> <time> <facility>.<severity> <program[<pid>]>: <message>
-----
2010 Jan 29 17:18:47 kern.warning awplus kernel: ARP: IP ac630105 VID 200
ifindex 5301 0000.0500.0200 (0000.0500.0100)
2010 Jan 29 17:18:47 kern.warning awplus kernel: ipi_sock_if_event:
Sending IPI_MSG_DHCPSN_ARPSEC_VIOLATION
2010 Jan 29 17:18:47 kern.warning awplus kernel:
ipi_msg_dhcpsn_arpsec_violation: Sending 20 bytes
2010 Jan 29 17:18:47 kern.warning awplus kernel: ARP violates DHCP
Snooping ARPsecurity
2010 Jan 29 17:18:47 kern.warning awplus kernel: ARP Sec Violation (4):
2010 Jan 29 17:18:47 kern.warning awplus kernel: Entry: IP ac630105 VID
200 ifindex 5301
```

And port1.0.10 is put into a shutdown state:

```
awplus#sho int br
```

Interface	Status	Protocol
port1.0.1	admin up	running
port1.0.2	admin up	down
port1.0.3	admin up	down
port1.0.4	admin up	down
port1.0.5	admin up	down
port1.0.6	admin up	down
port1.0.7	admin up	down
port1.0.8	admin up	down
port1.0.9	admin up	down
<b>port1.0.10</b>	<b>admin up</b>	<b>down</b>
port1.0.11	admin up	down
port1.0.12	admin up	down
port1.0.13	admin up	down
port1.0.14	admin up	down
port1.0.15	admin up	down
port1.0.16	admin up	down
port1.0.17	admin up	down
port1.0.18	admin up	down
port1.0.19	admin up	down
port1.0.20	admin up	down

There is no timeout period on this port shutdown. Once the port has been taken down, you will have to manually re-enable the port by going into interface **configuration** mode for that port and entering the command: **no shutdown**

## DHCP snooping with SNMP monitoring

Monitoring the x900 using SNMP—in this example we have configured the traps for both DHCP snooping and ARP security.

An important point to remember when using SNMP to monitor the switch is that there is a mechanism to limit the rate at which traps are sent for repeated intrusions from the same host. If there is an intrusion from a specific source MAC address, then no more traps relating to intrusions from that same MAC address will be sent for 60 seconds.

In this example, the switch has the DHCP server connected to port1.0.1 and the SNMP management station connected to port1.0.2; both of these ports are trusted.

To configure the traps to send when violations are seen on particular interfaces, use the commands:

DHCP snooping trap:

```
awplus#conf t
awplus(config)#int port1.0.10-1.0.20
awplus(config-if)#ip dhcp snooping violation trap
```

ARP security trap:

```
awplus#conf t
awplus(config)#int port1.0.10-1.0.20
awplus(config-if)#arp security violation trap
```

Full configuration for this example is as follows:

```
snmp-server
snmp-server enable trap dhcpsnooping
snmp-server community public rw
snmp-server host 192.168.10.254 version 2c public

service dhcp-snooping

vlan database
  vlan 10 name clients
!
interface port1.0.1-1.0.2
switchport access vlan 10
ip dhcp snooping trust

interface port1.0.10-1.0.20
switchport access vlan 10
ip dhcp snooping max-bindings 3
ip dhcp snooping violation trap
arp security violation log trap
!
interface vlan10
ip address 192.168.10.250/24
ip dhcp snooping
arp security
```

### What is a DHCP violation?

Error adding to database	Invalid BOOTP packet	Invalid DHCP ACK
Invalid IP packet	Invalid DHCP release	Max bindings exceeded
Insertion of Option 82 failed	Invalid Option 82 info received	Static entry already exists
Option 82 received on untrusted port	Option 82 would have been transmitted on untrusted port	BOOTP reply received on untrusted port
SRC MAC does not match BOOTP chaddr		

## Troubleshooting DHCP snooping in AW+

When troubleshooting DHCP snooping, AW+ provides various commands/debugging options to help troubleshoot any problems. The diagnostic options available range from the **show** commands, logging of DHCP snooping events, to the actual debugging of the protocol messages.

### DHCP show commands

```
awplus#show ip dhcp snooping ?
acl          DHCP snooping access-list
binding      Binding database
interface    Interface information
statistics   DHCP Snooping statistics
```

### DHCP debugging options

```
awplus#debug ip dhcp snooping ?
acl          Access-lists
all          Turn on all Debugging
db          Binding database debug
packet       Packet debug
```

### ARP security debugging

```
awplus>ena
awplus#debug arp security
awplus#term mon
```

These **show** commands are useful when troubleshooting ARP security.

```
show ip dhcp snooping binding
show ip dhcp snooping acl
show ip dhcp snooping acl hardware
show ip dhcp snooping acl detail
show platform classifier statistics utilization brief
```

### DHCP logging

Logging can be enabled for DHCP snooping with the following commands:

```
awplus#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
awplus(config)#log monitor program dhcpsn
```

Example DHCP snooping log message:

```
awplus#06:12:30 awplus DHCPSPN[1680]: Binding Add: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.20, Server 192.168.10.253, Type Dynamic,
Expires in 3600 seconds
06:12:33 awplus DHCPSPN[1680]: Binding Update: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.20, Server 192.168.10.253, Type Dynamic,
Expires in 3600(*) seconds
```

## Debugging

### Enabling DHCP snooping debugging

When debugging DHCP snooping on the AW+ platform, we need to enable terminal monitoring so the output comes on the terminal. Use the commands:

```
awplus#debug ip dhcp snooping packet
awplus#terminal monitor
```

When there has been a successful transaction between client and the DHCP server which results in the switch, which is configured for DHCP snooping, adding that client's MAC and IP addresses to the binding database, output like the following is seen:

```
11:03:25awplusDHCPSPN[1653]: [PKT00004]-----
11:03:25awplusDHCPSPN[1653]: [PKT00004]ReceivedDHCPsnoopingpacket
11:03:25 awplus DHCPSPN[1653]: [PKT 00004] Received on VLAN 10(Ingress)
IfIndex 5001 Length 590 11:03:25
awplus DHCPSPN[1653]: [PKT 00004] Processing BOOTP Reply
11:03:25 awplus DHCPSPN[1653]: [PKT 00004] Rxd on trusted port
11:03:25 awplus DHCPSPN[1653]: [PKT 00004] Client port: Found 5009 on
VLAN 10 11:03:25
awplus DHCPSPN[1653]: [PKT 00004] DHCP ACK Found
11:03:25 awplus DHCPSPN[1653]: Binding Add: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.9, Server 192.168.10.253, Type Dynamic,
Expires
11:03:25 awplus DHCPSPN[1653]: [PKT 00004] Sent DHCP Snooping packet for
forwarding, Length: 590
```

Below is an example of the debug output when a port has a maximum binding limit of 1 and another device attached to this port is requesting a DHCP lease.

```
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] -----
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] Received DHCP Snooping packet
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] Received on VLAN 10(Ingress)
IfIndex 5009 Length 590
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] Processing BOOTP Request
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] Rxd on untrusted port
11:05:10 awplus DHCPSPN[1653]: [PKT 00008] Discard packet, DHCP REQUEST
max bindings exceeded
```

## ACL debugging

DHCP snooping ACL debug will be generated whenever an action that affects the ACLs takes place, such as:

- Binding is learnt
- Binding expires
- ACLs are attached to port
- ACLs are detached from port
- Maximum bindings are changed on port

Enable this debugging with the command:

```
awplus#debug ip dhcp snooping acl
```

When ACLs are being created for the different clients that have been added to the binding database on a VCStack, the debug output is as below:

```
04:54:33 awplus DHCPSN[1701]: Binding Add: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.9, Server 192.168.10.253, Type Dynamic,s

04:54:33 awplus NSM[1008]: [DHCPSN-ACL] Stack master: propagating lease
update for 192.168.10.1 to slaves via CPG

04:54:33 awplus NSM[1008]: [DHCPSN-ACL] Adding cmap 1 binding 1 for
192.168.10.1
```

The configuration used for debugging has the following ACL attached to a port:

```
awplus(config)#access-list hardware test
awplus(config-ip-hw-acl)#permit ip dhcpsnooping any
awplus(config-ip-hw-acl)#deny ip any any
```

When a client successfully obtains a DHCP lease, it is added to the binding database. With this debugging enabled we can see that an ACL has been added to the hardware filter table for that particular client 192.168.10.1.

Here is some output from an x600 showing this debug: (In this example we use static entries to demonstrate the debugging messages).

```

awplus(config-if)#access-group test

13:20:16 awplus NSM[1004]: [DHCP SN-ACL] Set ACL - dhcp snooping
filter ifindex 5001
13:20:16 awplus NSM[1004]: [DHCP SN-ACL] Set filter seq 10 binding
1 ip 0.0.0.0
13:20:16 awplus NSM[1004]: [DHCP SN-ACL] Template filter test-10
added with pointer 0x102bd0b8
awplus(config-if)#exit

awplus(config)#ip source binding 192.168.1.1 vlan 1 int port1.0.1
13:20:24 awplus IMISH[6991]: ip source binding 192.168.1.1 vlan 1
int port1.0.1
13:20:24 awplus NSM[1004]: [DHCP SN-ACL] Stack master: propagating
lease update for 192.168.1.1 to slaves via CPG
13:20:24 awplus NSM[1004]: [DHCP SN-ACL] Adding access-group binding
1 for 192.168.1.1
13:20:24 awplus DHCP SN[7048]: Binding Add: 192.168.1.1, chaddr
0000.0000.0000, vlan1, port1.0.1, Server 0.0.0.0, Type Static.

awplus(config)#
awplus(config)#no ip source binding 192.168.1.1
13:20:30 awplus NSM[1004]: [DHCP SN-ACL] Stack master: propagating
lease update for 192.168.1.1 to slaves via CPG
13:20:30 awplus NSM[1004]: [DHCP SN-ACL] Deleting access-group
binding 1 for 0.0.0.0
13:20:30 awplus DHCP SN[7048]: Binding Delete: 192.168.1.1, chaddr
0000.0000.0000, vlan1, port1.0.1, Server 0.0.0.0, Type Static.

awplus(config)#int port1.0.1
awplus(config-if)#no access-group test
13:21:39 awplus NSM[1004]: [DHCP SN-ACL] Set ACL - dhcp snooping
filter ifindex 5001
13:21:39 awplus NSM[1004]: [DHCP SN-ACL] Template pointer stored
0x102bd0b8, template pointer deleting 0x102bd0b8

awplus(config-if)#ip dhcp snooping max-bindings 10
13:22:31 awplus NSM[1004]: [DHCP SN-ACL] Initialise clients on 5001
13:22:31 awplus NSM[1004]: [DHCP SN-ACL] Client list size 10
13:22:31 awplus DHCP SN[7048]: [ACL] Max bindings rx'ed from NSM:
ifindex 5001 bindings 10

```

## Debug messages

The two debug messages in response to the command **access-group test** are:

```

NSM[1004]: [DHCP SN-ACL] Set filter seq 10 binding 1 ip 0.0.0.0
NSM[1004]: [DHCP SN-ACL] Template filter test-10 added with pointer 0x102bd0b8

```

A detailed description of the two **access-group test** debug messages you see above follows.

## Understanding the first access-group test debug message

The first debug message shown above, is when the DHCP snooping filter is added to the hardware. There are three components to this message: seq (10), binding (1), ip (0.0.0.0). **seq** refers to the sequence number from the ACL. The ACL we had here is:

```
awplus(config)#access-list hardware test
awplus(config-ip-hw-acl)#permit ip dhcpsnooping any
awplus(config-ip-hw-acl)#deny ip any any
```

This ACL is using the default sequence numbers, so because the DHCP snooping filter is the first one on the list, it has a sequence number of 10. We could have configured the list like this to get different numbers:

```
awplus(config)#access-list hardware test
awplus(config-ip-hw-acl)#15 permit ip dhcpsnooping any
awplus(config-ip-hw-acl)#30 deny ip any any
```

In which case, the DHCP snooping sequence number reported in the debug message would have been 15.

Or

```
awplus(config)#access-list hardware test
awplus(config-ip-hw-acl)#permit ip 1.2.3.4/32 any
awplus(config-ip-hw-acl)#permit ip dhcpsnooping any
awplus(config-ip-hw-acl)#deny ip any any
```

In which case the sequence number would be 20, as it is using default numbering and it is the second entry in the ACL.

### Reserving ACL entry space

The **binding** refers to the **max-bindings** setting on the port. When ACLs are configured for DHCP snooping, the switch reserves maximum bindings space in the hardware ACL table. This avoids problems where DHCP snooping needs to add an ACL entry for a newly learnt client, but the HW has run out of space. By pre-allocating and reserving these entries up-front during configuration, we can ensure that DHCP snooping will not hit the problem where a new client cannot be added because the hardware ACL space is exhausted.

In the example above, the **max-bindings** setting was the default setting on the port, which is 1. If **max-bindings** was 3, you would see the debug message repeated 3 times, with the binding incrementing from 1 to 3 in each group, as it would allocate (reserve) 3 spaces in hardware for the ACL.

The **ip** part is 0.0.0.0, in this example, because we have not learnt any IP addresses on this port yet, so the default address, 0.0.0.0, that does not match any traffic, has been installed to the hardware entry. If we had been running DHCP snooping without any ACLs attached, and had learnt a client IP address on that port, say 192.168.23.5, then when the ACL is initially applied to the HW (which is what we are doing with the **access-group test** command) it would automatically be installed with the IP address 192.168.23.5, instead of the blank IP address.

## Understanding the second access-group test debug message

The second debug line shown on [page 36](#), Template filter test-10 added with pointer 0x102bd0b8, is basically for tracking purposes. It has two components: the ACL name and sequence number (test-10) and the template pointer (0x102bd0b8).

The specific details of what the software is doing at this point is really only relevant to developers, but in general what it is saying is "I have found a DHCP snooping filter from the ACL named 'test' at sequence number 10, and I will use this memory address to reference it". The memory address is used for debugging purposes, particularly around the dynamic ACLs. One thing to note, in particular, is that in the message output when the ACL is removed again, the same memory address is used:

```
Template pointer stored 0x102bd0b8, template pointer deleting
0x102bd0b8.
```

If these numbers were different, it would indicate an error had occurred.

Consider another example, where an ACL containing two entries is attached to a port that has DHCP snooping **max-bindings** set to 3.

```
awplus(config)#access-list hardware myACL
awplus(config-ip-hw-acl)#permit ip dhcpsnooping host 1.2.3.4/32
awplus(config-ip-hw-acl)#deny ip dhcpsnooping any

awplus(config)#interface port1.0.12
awplus(config-if)#ip dhcp snooping max-bindings 3
awplus(config-if)#access-group myACL
```

If two clients downstream of the port had already received DHCP leases for the IP addresses 192.168.1.1 and 192.168.1.2, then you would get the following debug messages:

```
[DHCP SN-ACL] Set filter seq 10 binding 1 ip 192.168.1.1
[DHCP SN-ACL] Set filter seq 10 binding 2 ip 192.168.1.2
[DHCP SN-ACL] Set filter seq 10 binding 3 ip 0.0.0.0
[DHCP SN-ACL] Template filter myACL-10 added with pointer 0xcafebabe
[DHCP SN-ACL] Set filter seq 20 binding 1 ip 192.168.1.1
[DHCP SN-ACL] Set filter seq 20 binding 2 ip 192.168.1.2
[DHCP SN-ACL] Set filter seq 20 binding 3 ip 0.0.0.0
[DHCP SN-ACL] Template filter myACL-20 added with pointer 0xdeadbeef
```

## DHCP snooping binding database debugging

DHCP snooping binding database debugging outputs messages relating to entries being added to and deleted from the DHCP snooping database.

In this example, there has been a successful transaction and we can see the client has been added to the binding database:

```
awplus# debug ip dhcp snooping db
```

```
18:51:50 awplus NSM[1030]: Port up notification received for port1.0.19
18:52:33 awplus DHCPSN[1761]: [DB] CHCache Added VID:10
MAC:0006.5b31.14af Port:5019
18:52:33 awplus DHCPSN[1761]: [DB] CHCache Refresh VID:10
MAC:0006.5b31.14af Port:5019->5019
18:52:33 awplus DHCPSN[1761]: [DB] 192.168.10.1, DB Entry Add
18:52:33 awplus DHCPSN[1761]: [DB] 192.168.10.1, Found existing entry
18:52:33 awplus DHCPSN[1761]: [DB] 192.168.10.1, update_flags:00000040
18:52:33 awplus DHCPSN[1761]: Binding Update: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.19, Server 192.168.10.253, Type Dynamic,
Exs
18:52:33 awplus DHCPSN[1761]: [DB] 192.168.10.1, Setting timeout in 3600
seconds
18:52:38 awplus DHCPSN[1761]: [DB] Write Database Mon, 8 Mar 2010 18:52:38
18:52:38 awplus DHCPSN[1761]: [DB] Saved binding db to flash successfully
18:52:43 awplus DHCPSN[1761]: [DB] CHCache Timeout VID:10
MAC:0006.5b31.14af Port:5019
18:52:43 awplus DHCPSN[1761]: [DB] CHCache Delete VID:10
MAC:0006.5b31.14af Port:5019
```

## DHCP snooping packet debugging

DHCP snooping packet debugging outputs messages relating to the DHCP packets that are snooped as they pass through the switch.

```
awplus#debug ip dhcp snooping packet
```

```
19:04:25 awplus DHCPSN[1761]: [PKT 00015] -----
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Received DHCP Snooping packet
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Received on VLAN 10(Ingress)
IfIndex 5019 Length 342
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Processing BOOTP Request
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Rxd on untrusted port
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Option 82 added 16 bytes giving
UDPLen:324 IPLen:344 PktLen:358
19:04:25 awplus DHCPSN[1761]: [PKT 00015] Sent DHCP Snooping packet for
forwarding, Length: 358
19:04:25 awplus DHCPSN[1761]: [PKT 00016] -----
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Received DHCP Snooping packet
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Received on VLAN 10(Ingress)
IfIndex 5001 Length 590
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Processing BOOTP Reply
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Rxd on trusted port
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Client port: Found 5019 on
VLAN 10
19:04:25 awplus DHCPSN[1761]: [PKT 00016] DHCP ACK Found
19:04:25 awplus DHCPSN[1761]: Binding Update: 192.168.10.1, chaddr
0006.5b31.14af, vlan10, port1.0.19, Server 192.168.10.253, Type Dynamic,
Exs
19:04:25 awplus DHCPSN[1761]: [PKT 00016] Sent DHCP Snooping packet for
forwarding, Length: 590
```

## ARP security debugging

ARP security debugging outputs messages relating to ARP security events.

```
awplus#
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus DHCPSN[1809]: ARP Sec: ARP source IP not in snooping DB.,
port1.0.20, vid 10, Src 192.168.10.20, Mac 0000.0a00.0000
13:57:16 awplus Target: 0000.cd29.878d 192.168.10.250
13:57:16 awplus kernel: [ARPSEC 00126] ARP Discarded: No Binding
13:57:16 awplus kernel: [ARPSEC 00127] RX ARP Request on
port1.0.20(untrusted) vlan10
13:57:16 awplus kernel: [ARPSEC 00127] MAC: 0000.0a00.0000 >
0000.cd29.878d 13:57:16 awplus kernel: [ARPSEC 00127] ARP: Sender
0000.0a00.0000 192.168.10.20
13:57:16 awplus kernel: [ARPSEC 00127] ARP: Target 0000.cd29.878d
192.168.10.250
```

**show ip dhcp snooping statistics**

06:12:42 awplus IMISH[2098]: sh ip dhcp snooping statistics

DHCP Snooping Statistics:

Interface	In Packets	In BOOTP Requests	In BOOTP Replies	In Discards
vlan1	0	0	0	0
vlan10	9	5	4	0
port1.0.1	4	0	4	0
port1.0.2	0	0	0	0
port1.0.3	0	0	0	0
port1.0.4	0	0	0	0
port1.0.5	0	0	0	0
port1.0.6	0	0	0	0
port1.0.7	0	0	0	0
port1.0.8	0	0	0	0
port1.0.9	0	0	0	0
port1.0.10	0	0	0	0
port1.0.11	0	0	0	0
port1.0.12	0	0	0	0
port1.0.13	0	0	0	0
port1.0.14	0	0	0	0
port1.0.15	0	0	0	0
port1.0.16	0	0	0	0
port1.0.17	0	0	0	0
port1.0.18	0	0	0	0
port1.0.19	0	0	0	0
port1.0.20	5	5	0	0
port1.0.21	0	0	0	0
port1.0.22	0	0	0	0
port1.0.23	0	0	0	0
port1.0.24	0	0	0	0

# Recommendations for configuring DHCP snooping/ARP security

When configuring DHCP snooping, think of the other features that you are wanting to configure on the switch. With both the x600 and the x900 series switches, there are features which consume either ACL entries or bytes.

With the x908/x900 series switches there is a 16 byte limitation on how much of a given packet can be matched on in the switch's silicon. Different features when enabled will also use bytes out of this 16 (EPSR, Loop Protection etc.), and when configuring ACL's these values below need to be taken into consideration:

The fields and their values are listed below:

Value (bytes)	Field
4	destination IP address
4	source IP address
2	source TCP port
2	destination TCP port
2	destination UDP port
2	source UDP port
1	DSCP field
1	ToS field
1	protocol field
6	destination MAC address
6	source MAC address
6	MAC type
2	IPX socket number
4	IPX destination address
4	VLAN priority

The x600 series switches have a limit of **121** ACLs allocated for use by other features and the ACLs themselves. With a larger network (a lot of clients/ max-bindings) these ACL entries can be consumed very quickly. If your network is going to have features that are going to be using ACL entries, then it would be a good idea for the DHCP snooping ACLs to create QoS policies and have the ACLs attached to these policies.