**DHCP Snooping**

DHCP snooping is a security feature that acts like a firewall between untrusted hosts and

Trusted DHCP servers. The DHCP snooping feature performs the following activities:

■ Validates DHCP messages received from untrusted sources and filters out invalid

messages.

■ Rate-limits DHCP traffic from trusted and untrusted sources.

■ Builds and maintains the DHCP snooping binding database, which contains information

about untrusted hosts with leased IP addresses.

■ Utilizes the DHCP snooping binding database to validate subsequent requests from

untrusted hosts

**Step by step procedure for implementing DHCP:**

The following steps are required to implement DHCP snooping on your network:

**Step 1.** Define and configure the DHCP server. Configuration of this step does not take

place on the switch or router and is beyond t he scope of this book.

**Step 2.** Enable DHCP snooping on at least one VLAN. By default, DHCP snooping is

inactive on all VLANs.

**Step 3.** Ensure that DHCP server is connected through a trusted interface.

By default, the trust state of all interfaces is untrusted.

**Step 4.** Configure the DHCP snooping database agent. This step ensures that database

entries are restore d after a restart or switchover.

**Step 5.** Enable DHCP snooping globally

**Implementing DHCP on Switch:**

🡪Enable DHCP Snooping Globally

sw2(config)# **ip dhcp snooping**

🡪 Enable DHCP Snooping on VLAN 10

sw2(config)# **ip dhcp snooping vlan 10**

🡪 Configure Interface Fa1/0/24 as a Trusted interface

sw2(config)# **interface fa1/0/24**

sw2(config-if)# **ip dhcp snooping trust**

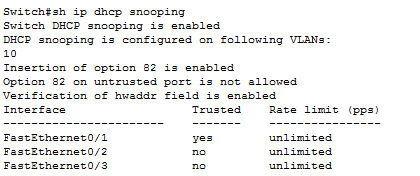
🡪Configure the DHCP snooping database agent to store the bindings at a given location

sw2(config)# **exit**

sw2#

🡪Verify DHCP Snooping Configuration

**Output:**



**Port Security**

How many MAC addresses should legitimately show up inbound on an access port?

Port security controls how many MAC addresses can be learned on a single switch port.

This feature is implemented on a port-by-port basis. A typical user uses just a single MAC

address.

Exceptions to this may be a virtual machine or two that might use different MAC

addresses than their host, or if there is an IP phone with a built-in switch, which may also

account for additional MAC addresses.

In any case, to avoid a user connecting dozens of devices to a switch that is then connected to their access port, you can use port security to limit the number of devices (MAC addresses) on each port.

This also protects against malicious applications that may be sending thousands of frames

into the network, with a different bogus MAC address for each frame, as the user tries to

exhaust the limits of the dynamic MAC address table on the switch, which might cause the

switch to forward all frames to all ports within a VLAN so that the attacker can begin to

sniff all packets. This is referred to as a *CAM table overflow attack*. *Content-addressable*

*memory (CAM)* is a fancy way to refer to the MAC address table on the switch.

**Implementing Port-Security on Switch:**

SW2(config-if)# **interface fa 0/2**

! Enable the feature per interface

SW2(config-if)# **switchport port-security**

! Set the maximum to desired number. Default is 1. If we administratively

! set the maximum to 1, the command won't show in the running configuration

! because the configuration matches the default value. It is handy to know

! this behavior, so you won't be surprised by what may seem to be a missing

! part of your configuration.

SW2(config-if)# **switchport port-security maximum 5**

! Set the violation action. Default is err-disable. Protect will simply

! not allow

! frames from MAC addresses above the maximum.

SW2(config-if)# **switchport port-security violation protect**

This will cause the dynamic mac addresses to be placed into running

! -config to save them to startup config, use copy run start

SW2(config-if)# **switchport port-security mac-address sticky**

! To verify settings, use this command

Output:

Switch#sh port-security

Secure Port MaxSecureAddr CurrentAddr SecurityViolation Security Action

(Count) (Count) (Count)

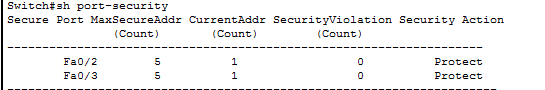
--------------------------------------------------------------------

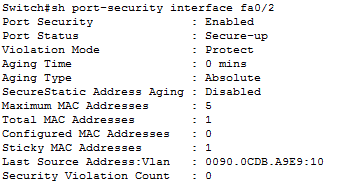
Fa0/2 5 1 0 Protect

Fa0/3 5 1 0 Protect

----------------------------------------------------------------------

Switch#





**Securing the Cisco IOS Image and Configuration Files**

If a router has been compromised, and the flash file system and NVRAM have been deleted,

Then there could be significant downtime as the files are put back in place before restoring normal router functionality. The Cisco Resilient Configuration feature is intended to improve the

recovery time by making a secure working copy of the IOS image and startup configuration

files (which are referred to as the *primary bootset*) that cannot be deleted by a remote user.

To enable and save the primary bootset to a secure archive in persistent storage, follow

Implementing Secure Boot

Secure the IOS image

R6(config)# **secure boot-image**

%IOS\_RESILIENCE-5-IMAGE\_RESIL\_ACTIVE: Successfully secured running image

! Secure the startup-config

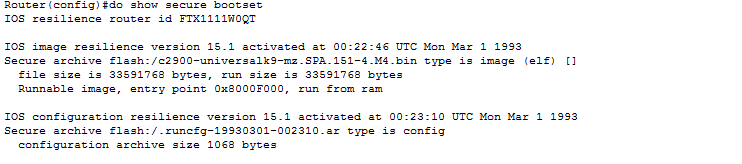
R6(config)# **secure boot-config**

%IOS\_RESILIENCE-5-CONFIG\_RESIL\_ACTIVE: Successfully secured config archive

[flash:.runcfg-20111222-230018.ar]

! Verify the bootset

Output:



**ARP Dynamic Inspection**

ARP provides IP communication within a Layer 2 broadcast domain by mapping an IP

address to a MAC address. ARP spoofing attacks and ARP cache poisoning can occur because ARP allows a gratuitous reply from a host even if an ARP request was not received. After the attack, all traffic from the device under attack flows through the attacker’s computer and then to the router, switch,or host.

An ARP spoofing attack can target hosts, switches, and routers connected to your Layer

2 network by poisoning the ARP caches of systems connected to the subnet and by intercepting

traffic intended for other hosts on the subnet.

DAI is a security feature that validates ARP packets in a network. DAI intercepts, logs, and

discards ARP packets with invalid IP-t o -MAC address bindings. This capability protects the

network from some man-in-the-middle attacks.

DAI determines the validity of an ARP packet based on valid IP-to-MAC address bindings

stored in a trusted database, the DHCP snooping binding database. As described in the previous

section, this database is built by DHCP snooping if DHCP snooping is enabled on the

VLANs and on the switch. If the ARP packet is received on a trusted interface, the switch

forwards the packet without any checks. On untrusted interfaces, the switch forwards the

packet only if it is valid.

Implementing ARP Dynamic Inspection

Enable DAI on VLAN 10

sw2(config)# **ip arp inspection vlan 10**

sw2(config)# **exit**

! Verify DAI Configuration for VLAN 10

sw2# **show ip arp inspection vlan 10**

Source Mac Validation : Disabled

Destination Mac Validation : Disabled

IP Address Validation : Disabled

Vlan Configuration Operation ACL Match Static ACL

---- ------------- --------- --------- ----------

10 Enabled Inactive

Vlan ACL Logging DHCP Logging Probe Logging

---- ----------- ------------ -------------

10 Deny Deny Off

! Configure Interface Fa1/0/24 as a Trusted DAI Interface

sw2(config)# **interface fa1/0/24**

sw2(config-if)# **ip arp inspection trust**

sw2(config-if)# **exit**

sw2(config)# **exit**

sw2# **show ip arp inspection interfaces**

Interface Trust State Rate (pps) Burst Interval

--------------- ----------- ---------- --------------

Fa1/0/1 Untrusted 15 1

Fa1/0/2 Untrusted 15 1

! output removed for brevity

Fa1/0/23 Untrusted 15 1

Fa1/0/24 Trusted None N/A