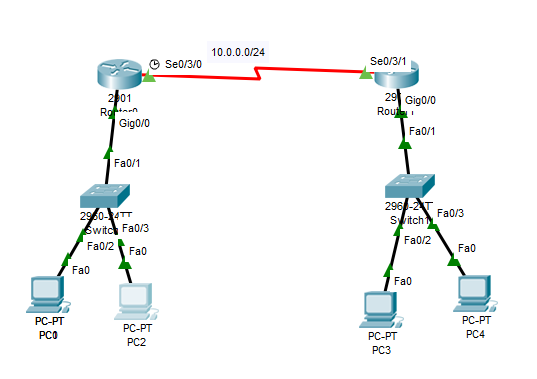
**CHAPTER 5**

**NETWORK SECURITY**

**10.1 Introduction**

Network security consists of the policies and practices to prevent and monitor unauthorized access, misuse, modification, or denial of a computer network and network-accessible resources. Only Network security can protect you from Trojan horse viruses. Network security involves the authorization of access to data in a system, controlled by the network administrator.



**Fig .10.1: Network Topology**

To avoid ARP SPOOFING, MAC FLOODING, and DHCP SPOOFING, we implement security policies. Firewall plays a preeminent role in network security. To prevent unauthorized access, we are using Cisco advanced security appliances. To connect different branches with security, we are implementing SITE TO SITE VPN. To overcome the network attacks, we are developing the Intrusion Prevention System. Cisco IOS Intrusion Prevention System (IPS) is an inline, deep-packet inspection feature that effectively mitigates a wide range of network attacks. These network security infrastructuresare implementing in On-premises, not in a cloud.

**10.2 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 the 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 restored 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**

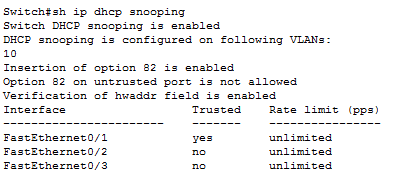
Configure the DHCP snooping database agent to store the bindings at a given location

sw2(config)# **exit**

sw2#

Verify DHCP Snooping Configuration

**Output:**

****

**Fig 10.2: DHCP Snooping Output**

**10.3 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 MaxSecure Addr CurrentAddr Security Violation Security Action

(Count) (Count) (Count)

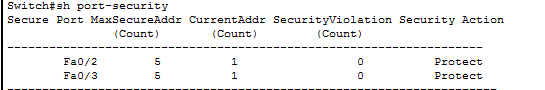
--------------------------------------------------------------------

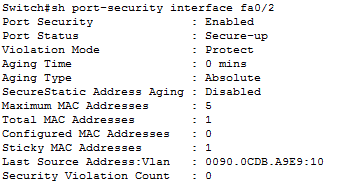
Fa0/2 5 1 0 Protect

Fa0/3 5 1 0 Protect

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

Switch#





**Fig 10.3: Port Security Output**

**10.4 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 start up configuration files (which are referred to as the *primary* boot set) that cannot be deleted by a remote user.

To enable and save the primary boot set to a secure archive in persistent storage, follow Secure the IOS image

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

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

! Secure the startup-config

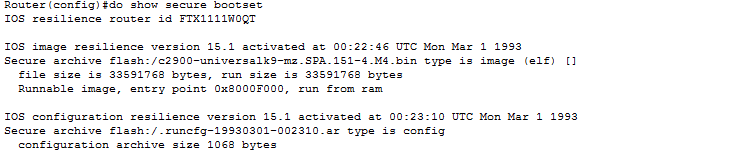
R6(config)# **secure boot-config**

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

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

! Verify the bootset

**Output:**



**Fig 10.5: Secured IOS Output**

**10.5 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, 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

Forward the packet without any checks. On untrusted interfaces, the switch forwards the packet only if it is valid.

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

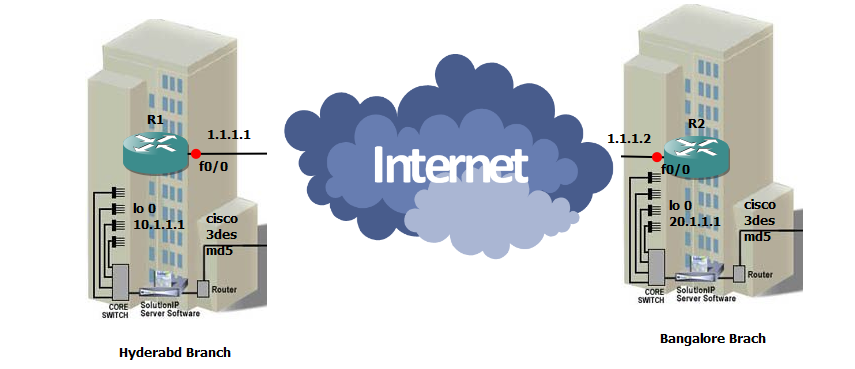
Fa1/0/2 Untrusted 15 1

! output removed for brevity

Fa1/0/23 Untrusted 15 1

Fa1/0/24 Trusted None N/A

**10.6 Site to Site VPN**

****

**Fig 10.6: Site to Site VPN**

It helps to connect two different branches or remote locations. It ensures that connectivity between two public networks and with security.

**Implementation Stages:**

Phase 1: Internet security association key management protocols (ISAKMP).

Phase 2: Internet protocol security. It is having ESP and AH.

Phase 3: Interesting traffic will be configured by ACL.

Phase 4: Mapping (Crypto Map).

Phase 5: Apply map on interface.

Here Pre shared key will be used for trust worthiness between two branches.

**Encryption:**

It will convert plain text into cipher text. It classified into 3 types.

1) DES-Data encryption standard

2) 3 DES

3) AES-Advanced encryption standard

**Hash:** It is used for checksum. It is having two types

1) MD5-Message Digest

2) SHA-Secure hashing Algorithm

**Ockley:** It is responsible to carry the message from Router 1 to Router 2.

If we are using both sites of routers are cisco vendors, then it is operated in main mode, in this 6 messages will be exchanged. For different vendors of router in both sites, it will be operated in Aggressive Mode only 3 messages exchanged.

In this VPN, we have 2 Tunnels. Those are

1) ISAKMP

2) IPSEC

**ISAKMP:** By using this tunnel session keys will be exchanged, this was generated by the **Diffie**–**Hellman** Algorithm. ISAKMP tunnel is responsible for IPSEC tunnel up and running.

**IPSEC:** This tunnel is used for actual data transmission. ESP and AH are responsible to carry data in IPSEC tunnel.

**Crypto Map:** It is used to identify the router the packet belongs to which network. Only one crypto map can be applied for single interface.

**HMac:** Hmac is used for sequence number and to add the Tags.

**10.6.1 VPN CONFIGURATION:**

**Router-1**

crypto isakmp policy 10

encr 3des

hash md5

authentication pre-share

group 2

crypto isakmp key cisco address 1.1.1.2

crypto isakmp key juniper address 1.1.1.3

!

!

crypto ipsec transform-set dell esp-3des esp-md5-hmac

crypto ipsec transform-set lenovo esp-aes esp-sha-hmac

!

crypto map irfan 110 ipsec-isakmp

set peer 1.1.1.2

set transform-set dell

match address 101

crypto map irfan 111 ipsec-isakmp

set peer 1.1.1.3

set transform-set lenovo

match address 102

!

interface Loopback0

ip address 10.1.1.1 255.0.0.0

!

interface FastEthernet0/0

ip address 1.1.1.1 255.0.0.0

duplex auto

speed auto

crypto map irfan

!

interface FastEthernet0/1

no ip address

shutdown

duplex auto

speed auto

!

ip route 0.0.0.0 0.0.0.0 FastEthernet0/0

!

no ip http server

no ip http secure-server

!

access-list 101 permit ip host 10.1.1.1 host 20.1.1.1

access-list 101 permit ip host 1.1.1.1 host 1.1.1.2

access-list 102 permit ip host 10.1.1.1 host 30.1.1.1

!

control-plane

!

line con 0

exec-timeout 0 0

privilege level 15

logging synchronous

line aux 0

exec-timeout 0 0

privilege level 15

logging synchronous

line vty 0 4

login

!

!

End

**Router-2**

crypto isakmp policy 10

encr 3des

hash md5

authentication pre-share

group 2

crypto isakmp key cisco address 1.1.1.1

!

!

crypto ipsec transform-set dell esp-3des esp-md5-hmac

!

crypto map irfan 110 ipsec-isakmp

set peer 1.1.1.1

set transform-set dell

match address 101

!

!

!

!

!

interface Loopback0

ip address 20.1.1.1 255.0.0.0

!

interface FastEthernet0/0

ip address 1.1.1.2 255.0.0.0

duplex auto

speed auto

crypto map irfan

!

interface FastEthernet0/1

no ip address

shutdown

duplex auto

speed auto

!

ip route 0.0.0.0 0.0.0.0 1.1.1.1

!

no ip http server

no ip http secure-server

!

access-list 101 permit ip host 20.1.1.1 host 10.1.1.1

access-list 101 permit ip host 1.1.1.2 host 1.1.1.1

!

!

control-planel!

!

line con 0

exec-timeout 0 0

privilege level 15

logging synchronous

line aux 0

exec-timeout 0 0

privilege level 15

logging synchronous

line vty 0 4

login

End

**OUTPUT**

R1#sh crypto isakmp sa

dst src state conn-id slot status

1.1.1.2 1.1.1.1 QM\_IDLE 1 0 ACTIVE

R1#sh crypto ipsec sa

interface: FastEthernet0/0

Crypto map tag: irfan, local addr 1.1.1.1

protected vrf: (none)

local ident (addr/mask/prot/port): (1.1.1.1/255.255.255.255/0/0)

remote ident (addr/mask/prot/port): (1.1.1.2/255.255.255.255/0/0)

current\_peer 1.1.1.2 port 500

PERMIT, flags={origin\_is\_acl,}

#pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0

#pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 0, #pkts compr. failed: 0

#pkts not decompressed: 0, #pkts decompress failed: 0

#send errors 0, #recv errors 0

local crypto endpt.: 1.1.1.1, remote crypto endpt.: 1.1.1.2

path mtu 1500, ip mtu 1500

current outbound spi: 0x0(0)

inbound esp sas:

inbound ah sas:

inbound pcp sas:

outbound esp sas:

outbound ah sas:

outbound pcp sas:

protected vrf: (none)

local ident (addr/mask/prot/port): (10.1.1.1/255.255.255.255/0/0)

remote ident (addr/mask/prot/port): (20.1.1.1/255.255.255.255/0/0)

current\_peer 1.1.1.2 port 500

PERMIT, flags={origin\_is\_acl,}

#pkts encaps: 14, #pkts encrypt: 14, #pkts digest: 14

#pkts decaps: 14, #pkts decrypt: 14, #pkts verify: 14

#pkts compressed: 0, #pkts decompressed: 0

#pkts not compressed: 0, #pkts compr. failed: 0

#pkts not decompressed: 0, #pkts decompress failed: 0

#send errors 6, #recv errors 0

local crypto endpt.: 1.1.1.1, remote crypto endpt.: 1.1.1.2

path mtu 1500, ip mtu 1500

current outbound spi: 0x65DCEB8B(1708977035)

inbound esp sas:

spi: 0xDA81E22F(3665945135)

transform: esp-3des esp-md5-hmac ,

in use settings ={Tunnel, }

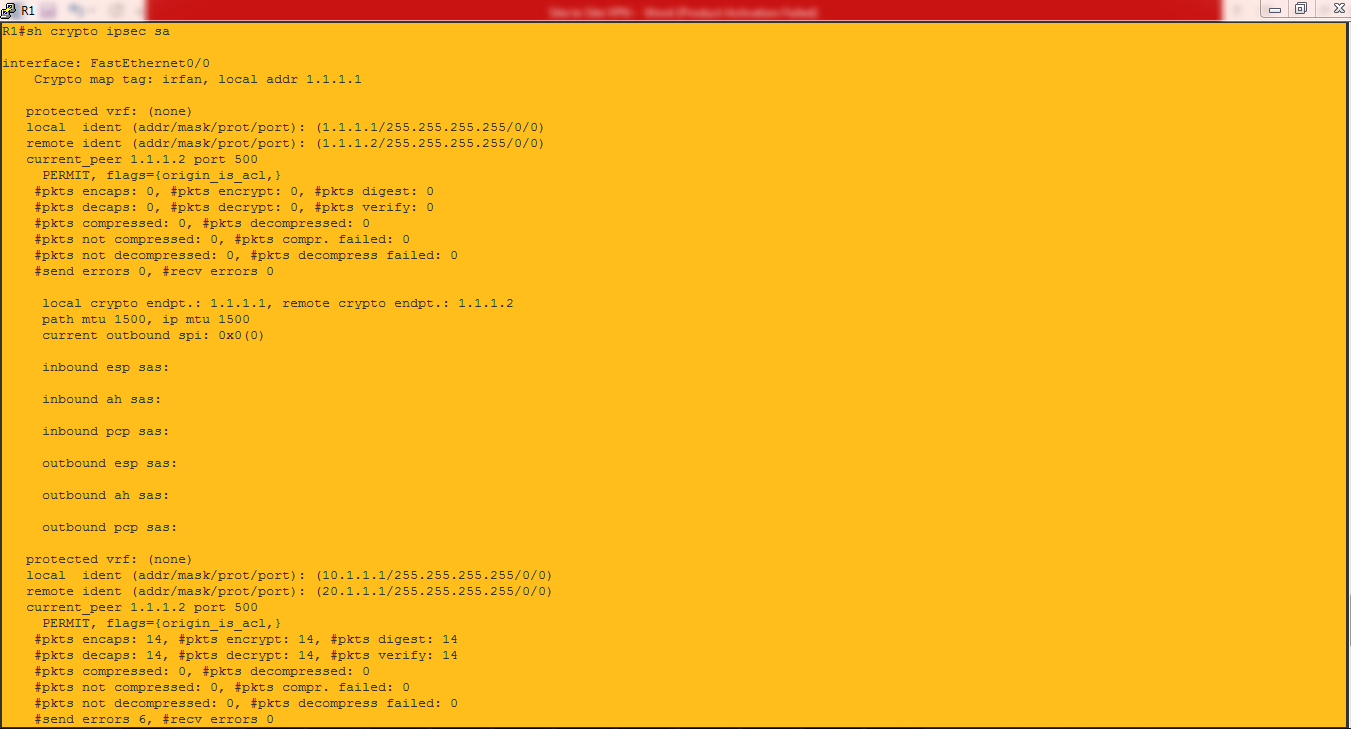
conn id: 2004, flow\_id: SW:4, crypto map: irfan

sa timing: remaining key lifetime (k/sec): (4441307/3533)

IV size: 8 bytes

replay detection support: Y

Status: ACTIVE



**Fig 10.6.1: VPN Output**

**Install an Intrusion Prevention System (IPS) on an EC2 Instance**

Intrusion Prevention Systems (IPS) monitor systems for malicious activity or policy violation, and report them to systems administrators or take automatic action on suspicious traffic within the network.

In this,we will use the popular **fail2ban** ([http://fail2ban.org](http://fail2ban.org/)) to automate a response to an attack on our EC2 instance.

With two EC2 instances, we will attack one from the other, and fail2ban will automatically block the attacking host's IP.

Introduction

Welcome to this hands-on AWS Learning Activity for installing an IPS on an EC2 instance. An Intrusion Prevention System, or IPS, monitors systems for malicious activity or policy violations, and report them to systems administrators or can take automatic action on suspicious traffic within the network. In this learning activity, we will use the popular open source package fail2ban to automate a response to an attack on our EC2 instance. We will simulate an attack from Instance B to Instance A. Let's say a hacker is trying to brute force an SSH login. Fail2ban, running on instance A, will detect multiple failed SSH login attempts, and block the malicious host's IP address for 10 minutes. Go ahead and install and configure fail2ban on Instance A. Then test multiple failed SSH logins from Instance B to Instance A. See if you can lock yourself out.