Practical 1: Configure Routers for Syslog, NTP and SSH operation.

Objectives:

1. Configure OSPF MD5 authentication.
2. Configure NTP.
3. Configure routers to log messages to the syslog server.
4. Configure R3 to support SSH connections.

PART 1: CONFIGURE ROUTER

Step 1: Configure password for vty lines

(Execute Command on all routers)

```
R(config) #line vty 0 4
R(config-line) #password vtypa55
R(config-line) #login
```

Step 2: Configure secret on router

Execute Command on all routers

```
R(config) # enable secret enpa55
```

Step 3: Configure OSPF on routers

```
R1(config) #router ospf 1
R1(config-router) #network 192.168.1.0 0.0.0.255 area 0
R1(config-router) #network 10.1.1.0 0.0.0.3 area 0

R2(config) #router ospf 1
R2(config-router) #network 10.1.1.0 0.0.0.3 area 0
R2(config-router) #network 10.2.2.0 0.0.0.3 area 0

R3(config) #router ospf 1
R3(config-router) #network 192.168.3.0 0.0.0.255 area 0
R3(config-router) #network 10.2.2.0 0.0.0.3 area 0
```

Step 4: Test Connectivity

```
PC-A > ping 192.168.3.5
```

```
Successful
PC-B > ping 192.168.3.5
Successful
```

PART 2: CONFIGURE OSPF MD5 Authentication

Step 1: Configure OSPF MD5 authentication for all the routers in area 0. (Execute Command on all routers)

```
R(config)# router ospf 1
R(config-router)# area 0 authentication message-digest
```

Step 2: Configure the MD5 key for all the routers in area.

```
R1(config)# int se0/1/0
R1(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

R2(config)# int se0/1/0
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
R2(config-if)# int se0/1/1
R2(config-if)# ip ospf message-digest-key 1 md5 MD5pa55

R3(config)# int se0/1/0
R3(config-if)# ip ospf message-digest-key 1 md5 MD5pa55
```

Step 3: Verify configurations.

(Execute Command on all routers)

```
R# show ip ospf interface
Message-digest Authentication Enabled
Youngest key ID is 1
```

PART 3: CONFIGURE NTP

Step 1: Enable NTP authentication on PC-A.

a. On PC-A, click NTP under the Services tab to verify NTP service is enabled.

b. To configure NTP authentication, click Enable under Authentication. Use key 1 and password NTPpa55 for authentication.

Step 2: Configure Routers as NTP clients.

(Execute Command on all routers)

```
R(config)# ntp server 192.168.1.5
```

Step 3: Configure routers to update hardware clock.

(Execute Command on all routers)

```
R(config)# ntp update-calendar
```

Step 4: Verify that the hardware Clock.

```
R# show clock
```

Step 5: Configure NTP authentication on the routers.

(Execute Command on all routers)

```
R(config)# ntp authenticate
R(config)# ntp trusted-key 1
R(config)# ntp authentication-key 1 md5 NTPpa55
```

Step 6: Configure routers to timestamp log messages.

(Execute commands on all routers)

```
R(config)# service timestamps log datetime msec
```

PART 4: CONFIGURE ROUTERS TO LOG MESSAGE TO THE SYSLOG SERVICE

Step 1: Configure the routers to identify the remote host (Syslog Server) that will receive logging messages.

(Execute Command on all routers)

```
R(config)# logging host 192.168.1.6
```

Step 2: Verify logging configuration.

(Execute Command on all routers)

```
R# show logging
```

O/P 2 message lines log

Step 3: Examine logs of the Syslog Server.

In the services of syslog server select syslog service observe the logs above.

Part 5: Configure R3 to Support SSH Connections

Step 1: Configure a domain name

```
R3(config)# ip domain-name ccnasecurity.com
```

Step 2: Configure users for login to the SSH server on R3.

```
R3(config)# username SSHadmin privilege 15 secret sshpa55
```

Step 3: Configure the incoming vty lines on R3.

```
R3(config)# line vty 0 4
R3(config-line)# login local
R3(config-line)# transport input ssh
```

Step 4: Erase existing key pairs on R3.

```
R3(config)# crypto key zeroize rsa
```

Step 5: Generate the RSA encryption key pair for R3.

```
R3(config)# crypto key generate rsa
How many bits in the modulus[512]:1024
```

Step 6: Verify the SSH configuration.

```
R3# show ip ssh
SSH enabled-version 1.99
Authentication time out: 120 secs; Authentication retries : 3
R3#
```

Step 7: Configure SSH timeouts and authentication parameters.

```
R3(config)# ip ssh time-out 90
R3(config)# ip ssh authentication-retries 2
R3(config)# ip ssh version 2
```

Step 8: Verify the SSH configuration

```
R3# show ip ssh
SSH enabled-version 2.0
Authentication time out: 90 secs; Authentication retries : 2
R3#
```

Step 9: Attempt to connect to R3 via Telnet from PC-C.

Open the Desktop of PC-C. Select the Command Prompt icon.

```
PC> telnet 192.168.3.1
(Unsuccessful)
```

Step 10: Connect to R3 using SSH on PC-C.

```
PC> ssh –l SSHadmin 192.168.3.1
Password: sshpa55
R3#
```

Step 11: Connect to R3 using SSH on R2.

```
R2# ssh –v 2 –l SSHadmin 10.2.2.1
Password: sshpa55
R3#
```

Practical 2: Configure AAA Authentication on Routers

Objectives:

1. Configure a local user account on R1 and configure authentication on the console and vty lines using local AAA
2. Verify local AAA authentication from the R1 console and the PC0 client and PC1 client.

PART 1: CONFIGURE ROUTER

Step 1: Configure password for vty lines

```
R1(config) # line vty 0 4
R1(config-line) #password vtypa55
R1(config-line) #login
```

Step 2: Configure secret on router

```
R1(config) # enable secret adminpa55
```

Step 3: Configure OSPF on routers

```
R1(config) #router ospf 1
R1(config-router) #network 192.168.1.0 0.0.0.255 area 0
```

Step 4: Configure OSPF MD5 authentication for all router in area 0

```
R1(config) #router ospf 1
R1(config-router)# area 0 authentication message-digest
```

Step 5: Configure MD5 key for all routers in area 0

```
R1(config)# int gig0/0
R1(config-if)# ip ospf message-digest-key 1 md5 pa55
```

Step 6: Verify MD5 authentication configuration.

```
R1# show ip ospf interface
Message-digest Authentication Enabled
Youngest key ID is 1
```

Step 7: Verify end-to-end connectivity

```
PC0 > ping 192.168.1.1
Successful
PC1 > ping 192.168.1.1
Successful
```

PART 2: Configure Local AAA Authentication for Console Access on R1

Step 1: Configure Local username on R1

```
R1(config)# username admin secret adminpa55
```

Step 2: Configure local AAA authentication for console access on R1.

```
R1(config)# aaa new-model
R1(config)# aaa authentication login default local
```

Step 3: Configure the line console to use the defined AAA authentication method.

```
R1(config)# line console 0
R1(config-line)# login authentication default
```

Step 5: Verify the AAA authentication method.

```
R1(config-line)# end
R1# exit
User Access Verification
```

```
Username: admin
Password: adminpa55
R1>
```

PART 3: Configure Local AAA Authentication for vty Lines on R1

Step 1: Configure domain name and crypto key for use with SSH.

```
R1(config)# ip domain-name ccnasecurity.com
R1(config)# crypto key generate rsa
How many bits in the modulus [512]: 1024
```

Step 2: Configure a named list AAA authentication method for the vty lines on R1.

```
R1(config)# aaa authentication login SSH-LOGIN local
```

Step 3: Configure the vty lines to use the defined AAA authentication method.

```
R1(config)# line vty 0 4
R1(config-line)# login authentication SSH-LOGIN
R1(config-line)# transport input ssh
R1(config-line)# end
```

Step 4: Verify the AAA authentication method.

```
PC0> ssh –l Admin 192.168.1.1
Password: adminpa55
R1>
PC1> ssh –l Admin 192.168.1.1
Password: adminpa55
R1>
```

PRACTICAL 3: CONFIGURE EXTENDED ACL'S

3.A)

OBJECTIVE:-

*configure,apply and verify an extended numbered acl

*configure,apply and verify an extended named acl

PART 1: CONFIRURE ROYTER

STEP1: configure password for vty lines

```
#line vty 0 4
#password vtypa55
#login
```

STEP 2: CONFIGURE SECRET PASSWORD AN ROUTER

```
#enable secret enpa55
```

PART 2: CONFIGURE,APPLY AND VERIFY & EXTENDED NUMBERED ACL

STEP 1: configure an acl to permit ftp and icmp

```
#access-list 100 permit tcp 172.22.34.64.0.0.0.31 host 172.22.34.62 eq ftp
#access_list 100 permit icmp 172.22.34.64.0.0.0.31 host 172.22.34.62
```

STEP 2: APPLY THE ACL ON THE CORRECT INTERFACE TO TRAFFIC

```
#int gig0/0
#ip access-group 100 in
```

STEP 3: VERIFY THE ACL IMPLEMENTATION

a.ping from pc1 to server.

```
  pc1>ping 172.22.34.62
```

b.ftp from pc1 to server.

the username and password are both cisco.

```
  pc1> ftp 172.22.34.62
```

c.exit the ftp service of the server.

```
  ftp>quit
```

d.ping from pc1 to pc2.

```
pc1>ping 172.22.34.98
```

PART 3: CONFIGURE,APPLY & VERIFY AN EXTENDED NAMED ACL

STEP 1: configure an acl to permit http access and icmp

```
#ip access-list extended HTTP-only
#permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www
#permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62
```

STEP 2: apply the cal on the correct interface to filter traffic

```
#int gig0/1
#ip access-group HTTP-only in
```

STEP 3: VERIFY the acl implementation

a.ping from pc2 to server

```
  pc2>ping 172.22.34.62
```

b.ftp from pc2 to server.

```
  pc2> ftp 172.22.34.62
```

c.open the web browser on pc2

  in url TYPE-> http:// 172.22.34.62

d.ping from pc2 to pc1

```
  pc1>ping 172.22.34.66
```

3.B)

OBJECTIVE:

configure,apply and verify an extended numbered acl

PART 1: CONFIGURE SWITCH AND ROUTER

STEP 1: CONFIGURE SWITCH AND ROUTER

```
SWA#int vlan1
SWA#ip address 10.101.117.50 255.255.255.248
SWA#no shut
SWA#ip default-gateway 10.101.117.49

SWB#int vlan1
SWB#ip address 10.101.117.34 255.255.255.240
SWB#no shut
SWB#ip default-gateway 10.101.117.33

SWC#int vlan1
SWC#ip address 10.101.117.2 255.255.255.224
SWC#no shut
SWC#ip default-gateway 10.101.117.1
```

STEP 2: CONFIGURE THE SECRET ON ROUTER AND SWITCH

(Execute command on all switch and router)

```
RTA/SW#enable secret enpa55
```

STEP 3: CONFIGURE THE CONSOLE PASSWORD AN ROUTER AND SWITCH

(Execute command on all switch and router)

```
RTA/SW#lineconsole 0
RTA/SW#password conpa55
RTA/SW#login
```

STEP 4: TEST CONNECTIVITY

ping from pca to pcb

```
pca>ping 10.101.117.35


pca>ping 10.101.117.2


pcb>ping 10.101.117.2
```

PART 2: CONFIGURE SWITCH AND ROUTER TO SUPPORT SSH CONNECTION

STEP 1: CONFIGURE DOMAIN NAME AND CRYPTO KEY FOR USE WITH SSH

(all switch/router)

```
#ip domain-name conasecurity.com
```

STEP 2: CONFIGURE USERS TO LOGIN TO SSH

(all switch/router)

```
#username admin secret adminpa55
```

STEP 3: CONFIGURE INCOMING VTY LINES

(all switch/router)

```
#line vty 04
#login local
#crypto key generate rsa
1024
```

STEP 4: VERIFY THE SSH CONNECTION

```
pca> ssh -1 admin 10.101.117.34
password: adminpa55
swb>
```

```
pca> ssh -1 admin 10.101.117.2
password: adminpa55
swc>
```

```
pcb> ssh -1 admin 10.101.117.50
password: adminpa55
swa>
```

```
pcb> ssh -1 admin 10.101.117.2
password: adminpa55
swc>
```

```
swc> ssh -1 admin 10.101.117.50
password: adminpa55
swa>
```

```
swc> ssh -1 admin 10.101.117.34
password: adminpa55
swa>
```

PART 3: CONFIGURE APPLY AND VERIFY AN EXTENDED NUMBERED ACL

STEP 1: CONFIGURE THE EXTENDED ACL

```
#access-list 199 permit tcp 10.101.117.32 0.0.0.15 10.101.117.0 0.0.0.31 eq 22
#access-list 199 permit icmp any any
```

STEP 2: APPLY THE EXTEND ACL

```
#int gig0/2
#ip access-group 199 out
```

STEP 3: VERIFY THE EXTENDED ACL IMPLEMENTATION

a. ping from pcb to all the other ip addresses in the network.

```
  pcb>ping 10.101.117.51
  pcb>ping 10.101.117.2
```

b. ssh from pcb to swc

```
  pcb>ssh -1 admin 10.101.117.2
  password: adminpa55
  swc>
```

c. exit the ssh session to swc

```
 swc>exit
```

d. ping from pca to all the other ip addresses in the network pca>ping 10.101.117.35

```
  pca>ping 10.101.117.2
```

e. ssh from pca to swc

```
  pca>ssh -1 admin 10.101.117.2
```

f. ssh from pca to swb

```
  pca>ssh -1 admin 10.101.117.34
  password: adminpoa55
```

g. After logging into swb do not log out.

ssh to swc in privileage exec mode.

```
  swb#ssh -1 admin 10.101.117.2
  password:adminpa55
  swc>
```

PRACTICAL 4: CONFIGURE IP & IPV6 ACL TO MITIGATE ATTACK

4.A] OBJECTIVE:

-verify connectivity among devices before firewall configuration

-use acls to ensure remote access to the router is avaliable from management station poc

-configure acls on r1 and r3 to mitigate attack

-verify acl funstionality

PART 1: CONFIGURE ROUTER

STEP 1: CONFIGURE ROUTER

(Execute command on all routers)

```
R(config)#enable secret enpa55
```

STEP 2: CONFIGURE CONSOLE PASSWORD ON ROUTER

(Execute command on all routers)

```
R(config)#line console 0
R(config-line)#password conpa55
R(config-line)#login
```

STEP 3: CONFIGURE SHH LOGINN ON ROUTER EXECUTE COMMAND ON ALL ROUTERS

(Execute command on all routers)

```
R(config)#ip domain-name conasecurity.com
R(config)#username admin secret adminpa55
R(config)#line vty 04
R(config-line)#login local
R(config-line)#crypto key generate rsa
1024
```

STEP 4: CONFIGURE LOOP BACK ADDRESS ON ROUTER 2

```
R2(config)#int loopback 0
R2(config-if)#ip address 192.168.2.1 255.255.255.0
R2(config-if)#no shut
```

STEP 5: CONFIGURE STATIC ROUTING ON ROUTERS

```
R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2
R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2
R1(config)#ip route 192.168.2.0 255.255.255.0 10.1.1.2

R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1

R3(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.2
R3(config)#ip route 192.168.2.0 255.255.255.0 10.2.2.2
```

```
R3(config)#ip route 10.1.1.0 255.255.255.0 10.2.2.2
```

STEP 6:FROM PCA VERIFY CONNECTIVELY TO PC-C & R2

```
pca>ping 192.168.3.3
(Successful)
```

```
pca>ping 192.168.2.1
(Successful)
```

```
pca>ssh -l admin 192.168.2.1
password:adminpa5
R2>exit
```

STEP 7:Step 2: From PC-C, verify connectivity to PC-A and R2.

```
PCC> ping 192.168.1.3
(Successful)
PCC> ping 192.168.2.1
(Successful)
PCC> ssh –l admin 192.168.2.1
Password: adminpa55
R2>exit
```

Open a web browser to the PC-A server (192.168.1.3) to display the web page.

Close the browser when done.

Desktop->Web Browser->192.168.1.3

(Successful)

Part 2: Secure Access to Routers

Step 1: Configure ACL 10 to block all remote access to the routers except from PC-C

Execute command on all routers

```
R(config)# access-list 10 permit host 192.168.3.3
```

Step 2: Apply ACL 10 to ingress traffic on the VTY lines.

Execute command on all routers

```
R(config)# line vty 0 4
```

```
R(config-line)# access-class 10 in
```

Step 3: Verify exclusive access from management station PC-C.

```
PCC> ssh –l admin 192.168.2.1
Password: adminpa55
R2>exit
```

Step 4: Verify denial from PC-A.

```
PCA> ssh –l admin 192.168.2.1
(Unsuccessful)Connection refused by remote host
```

Part 3: Create a Numbered IP ACL 120 on R1

Step 1: Verify that PC-C can access the PC-A via HTTPS using the web browser.

Be sure to disable HTTP and enable HTTPS on server PC-A in Services tab.

Click on PC-A -> Services -> HTTP amd enable HTTPS on server

Step 2: Configure ACL 120 to specifically permit and deny the specified traffic.

```
R1(config)# access-list 120 permit udp any host 192.168.1.3 eq domain
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq smtp
R1(config)# access-list 120 permit tcp any host 192.168.1.3 eq ftp
R1(config)# access-list 120 deny tcp any host 192.168.1.3 eq 443
R1(config)# access-list 120 permit tcp host 192.168.3.3 host 10.1.1.1 eq 22
```

Step 3: Apply the ACL to interface

```
R1(config)# int se0/1/0
R1(config-if)# ip access-group 120 in
```

Step 4: Verify that PC-C cannot access PC-A via HTTPS using the web browser.

PC-C Desktop->Web Browser->192.168.1.3

(Unsuccessful) Request timed out

Part 4: Modify an Existing ACL on R1

Step 1: Verify that PC-A cannot successfully ping the loopback interface on R2.

```
PCA> ping 192.168.2.1
(Unsuccessful) Request timed out
```

Step 2: Make any necessary changes to ACL 120 to permit and deny the specified traffic.

```
R1(config)# access-list 120 permit icmp any any echo-reply
R1(config)# access-list 120 permit icmp any any unreachable
R1(config)# access-list 120 deny icmp any any
R1(config)# access-list 120 permit ip any any
```

Step 3: Verify that PC-A can successfully ping the loopback interface on R2.

```
PCA> ping 192.168.2.1 (Successful)
```

Part 5: Create a Numbered IP ACL 110 on R3

Step 1: Configure ACL 110 to permit only traffic from the inside network.

```
R3(config)# access-list 110 permit ip 192.168.3.0 0.0.0.255 any
```

Step 2: Apply the ACL to interface

```
R3(config)# int gig0/1
R3(config-if)# ip access-group 110 in
```

Part 6: Create a Numbered IP ACL 100 on R3

Step 1: Configure ACL 100 to block all specified traffic from the outside network.

```
R3(config)# access-list 100 permit tcp 10.0.0.0 0.255.255.255 host 192.168.3.3 eq 22
R3(config)# access-list 100 deny ip 10.0.0.0 0.255.255.255 any
R3(config)# access-list 100 deny ip 172.16.0.0 0.15.255.255 any
R3(config)# access-list 100 deny ip 192.168.0.0 0.0.255.255 any
R3(config)# access-list 100 deny ip 127.0.0.0 0.255.255.255 any
```

```
R3(config)# access-list 100 deny ip 224.0.0.0 15.255.255.255 any
R3(config)# access-list 100 permit ip any any
```

Step 2: Apply the ACL to interface

```
R3(config)# int se0/1/0
R3(config-if)# ip access-group 100 in
```

Step 3: Confirm that the specified traffic entering interface Serial is handled correctly.

```
PCC> ping 192.168.1.3
(Unsuccessful) Request timed out
PCC> ssh –l admin 192.168.2.1
Password: adminpa55
R2>exit
```

4.B]

Topology:

Objective:

- Configure, Apply, and Verify an IPv6 ACL
- Configure, Apply, and Verify a Second IPv6 ACL

Part 1: Configure Router:

Step 1: Configure secret on router

Execute command on all routers

```
R(config)# enable secret enpa55
```

Step 2: Assign static ipv6 address

```
R1(config)# int gig0/0
R1(config-if)# ipv6 address 2001:DB8:1:10::1/64
R1(config-if)# ipv6 address FE80::1 link-local
R1(config-if)# no shut

R1(config)# int gig0/1
R1(config-if)# ipv6 address 2001:DB8:1:11::1/64
```

```
R1(config-if)# ipv6 address FE80::1 link-local
R1(config-if)# no shut

R1(config)# int se0/1/0
R1(config-if)# ipv6 address 2001:DB8:1:1::1/64
R1(config-if)# ipv6 address FE80::1 link-local
R1(config-if)# no shut

R2(config)# int se0/1/0
R2(config-if)# ipv6 address 2001:DB8:1:1::2/64
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# no shut

R2(config)# int se0/1/1
R2(config-if)# ipv6 address 2001:DB8:1:2::2/64
R2(config-if)# ipv6 address FE80::2 link-local
R2(config-if)# no shut

R3(config)# int gig0/0
R3(config-if)# ipv6 address 2001:DB8:1:30::1/64
R3(config-if)# ipv6 address FE80::3 link-local
R3(config-if)# no shut

R3(config)# int se0/1/0
R3(config-if)# ipv6 address 2001:DB8:1:2::1/64
R3(config-if)# ipv6 address FE80::3 link-local
R3(config-if)# no shut
```

Step 3: Enable IPv6 routing

```
R1(config)# ipv6 unicast-routing
R1(config)# ipv6 route 2001:DB8:1:2::0/64 2001:DB8:1:1::2
R1(config)# ipv6 route 2001:DB8:1:30::0/64 2001:DB8:1:1::2

R2(config)# ipv6 unicast-routing
R2(config)# ipv6 route 2001:DB8:1:10::0/64 2001:DB8:1:1::1
R2(config)# ipv6 route 2001:DB8:1:11::0/64 2001:DB8:1:1::1
```

```
R2(config)# ipv6 route 2001:DB8:1:30::0/64 2001:DB8:1:2::1
```

```
R3(config)# ipv6 unicast-routing
R3(config)# ipv6 route 2001:DB8:1:10::0/64 2001:DB8:1:2::2
R3(config)# ipv6 route 2001:DB8:1:11::0/64 2001:DB8:1:2::2
R3(config)# ipv6 route 2001:DB8:1:1::0/64 2001:DB8:1:2::2
```

Step 4: Verify connectivity

```
PC1> ping 2001:DB8:1:30::30
(Successful)
PC2> ping 2001:DB8:1:30::30
(Successful)
```

Part 2: Configure, Apply, and Verify an IPv6 ACL

Step 1: Configure an ACL that will block HTTP and HTTPS access.

```
R1(config)# ipv6 access-list BLOCK_HTTP
R1(config-ipv6-acl)# deny tcp any host 2001:DB8:1:30::30 eq www
R1(config-ipv6-acl)# deny tcp any host 2001:DB8:1:30::30 eq 443
R1(config-ipv6-acl)# permit ipv6 any any
R1(config-ipv6-acl)# exit
```

Step 2: Apply the ACL to the correct interface.

```
R1(config)# int gig0/1
R1(config-if)# ipv6 traffic-filter BLOCK_HTTP in
```

Step 3: Verify the ACL implementation

Open a web browser to the PC1 to display the web page.

Desktop->Web Browser->http://2001:DB8:1:30::30

(Successful)

Desktop->Web Browser->https://2001:DB8:1:30::30

(Successful)

Open a web browser to the PC2 to display the web page.

Desktop->Web Browser->http://2001:DB8:1:30::30

(Unsuccessful) – Request Timeout

Desktop->Web Browser->https://2001:DB8:1:30::30

(Unsuccessful) – Request Timeout

PC2> ping 2001:DB8:1:30::30

(Successful)

Part 3: Configure, Apply, and Verify a Second IPv6 ACL

Step 1: Create an access list to block ICMP.

```
R3(config)# ipv6 access-list BLOCK_ICMP
R3(config-ipv6-acl)# deny icmp any any
R3(config-ipv6-acl)# permit ipv6 any any
R3(config-ipv6-acl)# exit
```

Step 2: Apply the ACL to the correct interface.

```
R3(config)# int gig0/0
R3(config-if)# ipv6 traffic-filter BLOCK_ICMP out
```

Step 3: Verify that the proper access list functions.

PC2> ping 2001:DB8:1:30::30

(Unsuccessful) - Destination host unreachable

PC1> ping 2001:DB8:1:30::30

(Unsuccessful) - Destination host unreachable

Open a web browser to the PC1 to display the web page.

Desktop->Web Browser->http://2001:DB8:1:30::30

(Successful)

Desktop->Web Browser->https://2001:DB8:1:30::30

(Successful)

PRACTICAL 5: CONFIGURE ZONE BASED POLICY FIREWALL (ZPF)

OBJECTIVE:-

*verify connectivity among devices before firewall configuration

*configure a zone based policy firewall on R3

*verify ZPF functionality using ping, and web browser

PART 1: CONFIGURE ROUTER

STEP1: configure console password

Execute command on all routers

```
R(config)#line console 0
R(config-line)#password conpa55
R(config-line)#login
```

STEP2: configure password for vty lines

Execute command on all routers

```
R(config)#line vty 0 4
R(config-line)#password vtypa55
R(config-line)#login
```

STEP3: configure secret on router

Execute command on all router

```
R(config)#enable secret enpa55
```

STEP4: configure SSH login on router

Execute command on all routers

```
R(config)#ip domain-name ccnasecurity.com
R(config)#username admin secret adminpa55
R(config)#line vty 0 4
R(config-line)#login local
R(config-line)#crypto key generate rsa
How many bits [512]: 1024
```

STEP5: configure static on routers

```
R1(config)#ip route 10.2.2.0 255.255.255.252 10.1.1.2
R1(config)#ip route 192.168.3.0 255.255.255.0 10.1.1.2

R2(config)#ip route 192.168.1.0 255.255.255.0 10.1.1.1
R2(config)#ip route 192.168.3.0 255.255.255.0 10.2.2.1

R3(config)#ip route 192.168.1.0 255.255.255.0 10.2.2.2
R3(config)#ip route 10.1.1.0 255.255.255.252 10.2.2.2
```

a. PCA>ping 192.168.3.3 (success)

b. access R2 ussing ssh

```
PCC>ssh -l admin 10.2.2.2
password: adminpa55
P2>exit
```

c. from PCC open web browser to PCA server

Desktop-- Web Browser-- URL: http://192.168.1.3 (success)

PART2: CREATE THE FIREWALL ZONE ON R3

STEP1: verify that security technology package

```
R3#show version
```

output--

```
ipbase      ipbasek9    permanent   ipbasek9
security    none        none        none
data        none        none        none
```

STEP2: enable security tecjnology package

```
R3(config)#license boot module c1900 technology-package securityk9
```

STEP3: save the rinning-config and reload router

```
R3#copy run start
R3#reload
```

STEP4: verify the security technology package

```
R3#show version
```

```
ipbase      ipbasek9    permanent   ipbasek9
security    securityk9  evaluation  securityk9
data        disable     none        none
```

STEP5: create an internal zone

```
R3(config)#zone security IN-ZONE
R3(config-sec-zone)#exit
```

STEP6: create an external zone

```
R3(config)#zone security OUT-ZONE
R3(config-sec-zone)#exit
```

PART3: IDENTIFY TRAFFIC USING CLASS-MAP

STEP1: create ACL that defines internal traffic

```
R3(config)#access-list 101 permit ip 192.168.3.0 0.0.0.255 any
```

STEP2: create class map referencing internal traffic ACL

```
R3(config)#class-map type inspect match-all IN-NET-CLASS-MAP
R3(config-cmap)#match access-group 101
R3(config-cmap)#exit
```

PART4: SPECIFY FIREWALL POLICIES

STEP1; create a policy map to determine what to do with matched traffic

```
R3(config)#policy-map type inspect IN-2-OUT-PMAP
```

STEP2: specify class type of inspect and reference class map IN-NET-CLASS-MAP

```
R3(config-pmap)#class type inspect IN-NET-CLASS-MAP
```

STEP3: specify action of inspect for this policy map

```
R3(config-pmap-c)#inspect
R3(config-pmap-c)#exit
R3(config-pmap)#exit
```

PART5: APPLY FRIREWALL POLICIES

STEP1: create a pair of zones

```
R3(config)#zone-pair security IN-2-OUT-ZPAIR source IN-ZONE destination OUT-ZONE
```

STEP2: specify policy map for handling traffic between two zones

```
R3(config-sec-zone-pair)#service-policy type inspect IN-2-OUT-PMAP
R3(config-sec-zone-pair)#exit
```

STEP3: assign interfaces to appropriate security zones

```
R3(config)#int gig0/0
R3(config-if)#zone-member security IN-ZONE
R3(config-if)#exit
R3(config)#int se0/1/0
R3(config-if)#zone-member security OUT-ZONE
R3(config-if)#exit
```

STEP4: copy the running configuration to startup configuration

```
R3#copy run start
R3#reload
```

PART6: TEST FIREWALL FUNCTIONALITY FROM IN-ZONE TO OUT-ZONE

STEP1: from internal PCC ping external PCA server

```
PCC>ping 192.168.1.3 (success)
```

STEP2: access R2 using SSH

```
PCC>ssh -l admin 10.2.2.2
Password: adminpa55
R2>
```

STEP3: view established sessions

```
R3#show policy-map type inspect zone-pair sessions
```

(session will be established)

STEP4: from PCC exit SSH session on R2 and close command prompt

```
R2>exit
```

STEP5: from internal PCC open web browser to PCA server web page

Desktop-- Web Browser-- URL: http://192.168.1.3 (success)

STEP6: view extablished sessions

```
R3#show policy-map type inspect zone-pair sessions
```

(session will be established)

PART7: TEST FIREWALL FUNCTIONALITY FROM OUT-ZONE TO IN-ZONE

STEP1: from internal PCA ping the external PCC server

```
PCA>ping 192.168.3.3 (unsuccess-- time out)
```

STEP2: from R2 ping PCC

```
R2#ping 192.168.3.3 (unsuccess-- time out)
```

PRACTICAL 6: CONFIGURE IOS INTRUSION PREVENTION SYSTEM (IPS)

OBJECTIVE:-

*enable IOS IPS

*configure logging

*modify IPS signature

*verify IPS

PART1: CONFIGURE ROUTER

STEP1: configure secret on router

(Execute command on all router)

```
R(config)#enable secret enpa55
```

STEP2: configure console password on router

(Execute command on all router)

```
R(config)#line console 0
R(config-line)#password conpa55
R(config-line)#login
```

STEP3: configure SSH ligin on router

(Execute command on all router)

```
R(config)#ip domain-name ccnasecurity.com
R(config)#username admin secret adminpa55
R(config)line vty 0 4
```

```
R(config-line)#login local
R(config)#crypto key generate rsa
How many bits [512]: 1024
```

STEP4: configure OSPF on router

```
R1(config)#router ospf 1
R1(config-router)#network 192.168.1.0 0.0.0.255 area 0
R1(config-router)#network 10.1.1.0 0.0.0.3 area 0

R2(config)#router ospf 1
R2(config-router)#network 10.1.1.0 0.0.0.3 area 0
R2(config-router)#network 10.2.2.0 0.0.0.3 area 0

R3(config)#router ospf 1
R3(config-router)#network 10.2.2.0 0.0.0.3 area 0
R3(config-router)#network 192.168.3.0 0.0.0.255 area 0
```

STEP5: verify network connectivity

```
PCA>ping 192.168.3.2 (success)
PCC>ping 192.168.1.2 (success)
```

PART2: ENABLE IOS IPS

STEP1: verify the security technology package

```
R1#show version
```

(output)

```
Technology PAckage License Information for module "c1900"
Technology      Technology-package      Technology-package
                current         type            next     reboot

ipbase          ipbasek9        permanent       ipbasek9
security        none            none            none
data            none            none            none
```

STEP2: enable security technology package

```
R1(config)#license boot module c1900 technology-package securityk9
```

STEP3: save runnng config and reload router

```
R1#copy run start
R1#reload
```

SETP4: verify the security technology package

```
R1#show version
```

(output)

```
Technology PAckage License Information for module "c1900"
Technology    Technology-package    Technology-package
              current     type        next    reboot

ipbase        ipbasek9    permanent   ipbasek9
security      securityk9  evaluation  securtiyk9
data          disable     none        none
```

STEP5: create an IOS IPS configuration directory in flash

```
R1#mkdir ipsdir
create directory filename [ipsdir]? <Enter>
```

STEP6: configure IPS signature storage location

```
R1(config)#ip ips config location flash:ipsdir
```

STEP7: create an IPS rule

```
R1(config)#ip ips name iosips
```

STEP8: enable logging

```
R1(config)#ip ips notify log
R1#clock set hr:min:sec date month year (enter current data)
R1(config)#service timestamps log datetime msec
R1(config)#logging host 192.168.1.50
```

STEP9: configure IOS IPSto use signature categories

```
R1(config)#ip ips signature-category
```

```
R1(config-ips-category)#category all
R1(config-ips-category-action)#retired true
R1(config-ips-category-action)#exit
R1(config-ips-category)category ios_ips basic
R1(config-ips-category-action)#retired false
R1(config-ips-category-action)#exit
R1(config-ips-category)#exit
Do you want to accept changes? [confirm] <Enter>
```

Step10: apply IPS rule to interface

```
R1(config)#int gig0/0
R1(config-if)#ip ips iosips out
```

STEP11: use show commands to verify IPS

```
R1#show ip ips  all
```

(output)

STEP12: view syslog message

Click syslog server-- Services tab-- SYSLOG

(output)

PART3: MODIFY THE SIGNATURE

STEP1: change the event-action of signature

```
R1(config)#ip ips signature-definition
R1(config-sigdef)#signature 2004 0
R1(config-sigdef-sig)#status
R1(config-sigdef-sig-status)#retired false
R1(config-sigdef-sig-status)#enable true
R1(config-sigdef-sig-status)#exit
R1(config-sigdef-sig)#engine
R1(config-sigdef-sig-engine)#event-action produce-alert
R1(config-sigdef-sig-engine)#event-action deny-packet-inline
R1(config-sigdef-sig-engine)#exit
R1(config-sigdef-sig)#exit
R1(config-sigdef)#exit
```

```
Do you want to accept these changes? [confirm] <Enter>
```

STEP2: use show commands to verify IPS

```
R1#show ip ips all
```

(output)

STEP3: verify that IPS is working property

```
PCC>ping 192.168.1.2 (unsuccess-- time out)
PCA>ping 192.168.1.2 (success)
```

STEP4: verify syslog message

SYSLOG server

(output)

PRACTICAL 7: LAYER 2 SECURITY

OBJECTIVES:-

*assign central switch as root bridge

*secure-spanning tree parameter to prevent STP manipulation attacks

*enable part secutiy toprevent CAM table

PART1: CONFIGURE SWITCH/ROUTER

STEP1: configure secret

(Execute command on all switch and router)

```
R1/SW(config)#enable secret enpa55
```

STEP2: configure console passsword

(Execute command on all switch and router)

```
R1/SW(config)#line console 0
R1/SW(config-line)#password conpa55
R1/SW(config-line)#login
```

STEP3: configure SSH login

(Execute commadn on all switch and router)

```
R1/SW(config)#ip domain-name ccnasecurity.com
R1/SW(config)#username admin secret adminpa55
```

```
R1/SW(config)#line vty 0 4
R1/Sw(config-line)#login local
R1/SW(config-line)#crypto key generate rsa
```

PART2: CONFIGURE ROOT BRIDGE

STEP1: determine the current root bridge

```
SW#show spanning-tree
```

(output)

```
Spanning-tree enabled protocol IEEE
This bridge is the root.
```

STEP2: assign central as primary root bridge

STEP3: assign Sw-1 as secondary root bridge

```
SW1(config)#spanning-tree vlan 1 root secondary
SW1#show spanning tree
```

(output)

PART3: PROTECT AGAINST SSTP ATTACK

STEP1: enable port fast on all access ports

```
SWA/B(config)#int range fa0/1-4
SWA/B(config-if-range)#spanning-tree postfast
```

STEP2: enable BPDU guard on all access ports

```
SWA/B(config)#int range fa0/1-4
SWA/B(config-if-range)#spanning-tree bpduguard enable
```

STEP3: enable root guard

```
SW1/2(config)#int range fa0/23-24
SW1/2(config-if-range)#spanning-tree guard root
```

PART4: CONFIGURE PORT SECURITY AND DISABLE UNUSED PORTS

STEP1: configure basic port security on all ports connected to host devices

```
SWA/B(config)#int range fa0/1-22
SWA/B(config-if-range)#switchport mode access
SWA/B(config-if-range)#switchport port-security
SWA/B(config-if-range)#switchport port-security maximum 2
SWA/B(config-if-range)#switchport port-security violation shutdown
SWA/B(config-if-range)#switchport port-security mac-address sticky
```

STEP2: verify port security

```
SWA/B#show port-security int fa0/1
```

(output)

STEP3: desabled unused ports

```
SWA/B(config)#int range fa0/5-22
SWA/B(config-if-range)#shutdown
```

STEP4: verify connectivity

```
C1>ping 10.1.1.11 (success)
C1>ping 10.1.1.14 (success)
```

STEP5: verify port security

```
SWA/B #show port-security int fa0/1
```

(output)

PRACTICAL 8: LAYER 2 VLAN SECURITY

OBJECTIVES:

*connect a new redundnt link between SW1 and SW2

*enable trunking and configure security on new trunk link between SW1 and SW2

*create a new managemnet VLAN (VLAN20) and attach a management PC to VLAN

*implement a ACL to preven outside users from accessing management VLAN

PART 1:CONFIGURE SWITCH/ROUTER

STEP 1:Configure secret .

Execute command on all switches/routers

```
SW/R1(config)#enable secret enpa55
```

STEP 2:Configure console password

Execute command on all switches/routers

```
SW/R1(config)#line console 0
SW/R1(config-line)#password conpa55
SW/R1(config-line)#login
```

STEP 3:Configure SSH login

Execute command on all switches/routers

```
SW/R1(config)#ip domain-name ccnasecurity.com
SW/R1(config)#username admin secret adminpa55tz
SW/R1(config)#line vty 0 4
SW/R1(config-line)#login local
SW/R1(config)#cyrpto key generate rsa
How many bits in the modulus[512]:1024
```

PART 2: Create VLAN and assign access mode and trunk mode to interfaces

Step 1: Check existing VLAN

(Execute command on all switches)

```
SW# show vlan brief
```

(Output)

Step 2: Create new VLAN

(Execute command on all switches)

```
SW(config)# vlan 5
SW(config-vlan) # exit
SW(config)# vlan 10
SW(config-vlan) # exit
SW(config)# vlan 15
SW(config-vlan) # exit
```

(Output)

Step 3: Check the new VLAN

(Execute command on all switches)

```
SW# show vlan brief
```

(Output)

Step 4: Assign access mode to VLAN switch interfaces

(Execute command on switches SWA/SWB)

```
SWA(config)# int fa0/2
SWA(config -if)# switchport mode access
SWA(config -if)# switchport access vlan 10

SWA(config)# int fa0/3
SWA(config -if)# switchport mode access
SWA(config -if)# switchport access vlan 10

SWA(config)# int fa0/4
SWA(config -if)# switchport mode access
SWA(config -if)# switchport access vlan 5

SWB(config)# int fa0/1
SWB(config -if)# switchport mode access
SWB(config -if)# switchport access vlan 5

SWB(config)# int fa0/2
SWB(config -if)# switchport mode access
SWB(config -if)# switchport access vlan 5

SWB(config)# int fa0/3
SWB(config -if)# switchport mode access
SWB(config -if)# switchport access vlan

SWB(config)# int fa0/4
SWB(config -if)# switchport mode access
SWB(config -if)# switchport access vlan 10
```

Step 5: Check the access mode allocations

```
SWA# show vlan brief
```

(Output)

```
SWB# show vlan brief
```

(Output)

Step 6: Assign trunk mode to other switch interfaces

```
SWA(config)# int fa0/24
SWA(config -if)# switchport mode trunk
SWA(config -if)# switchport trunk native vlan 15

SWB(config)# int fa0/24
SWB(config -if)# switchport mode trunk
SWB(config -if)# switchport trunk native vlan 15

SW1(config)# int fa0/24
SW1(config -if)# switchport mode trunk
SW1(config -if)# switchport trunk native vlan 15

SW1(config)# int gig0/1
SW1(config -if)# switchport mode trunk
SW1(config -if)# switchport trunk native vlan 15

SW2(config)# int fa0/24
SW2(config -if)# switchport mode trunk
SW2(config -if)# switchport trunk native vlan 15

SW2(config)# int gig0/1
SW2(config -if)# switchport mode trunk
SW2(config -if)# switchport trunk native vlan 15

Central(config)# int range gig0/1-2
Central(config –if-range)# switchport mode trunk
Central(config –if-range)# switchport trunk native vlan 15

Central(config)# int fa0/1
Central(config –if)# switchport mode trunk
Central(config –if)# switchport trunk native vlan 15
```

Step 7: Check the trunk mode allocations

```
Central# show int trunk
(Output)
SW1/2# show int trunk
(Output)
SWA/B# show int trunk
(Output)
```

Step 8: Create sub-interfaces on router to support VLAN

```
R1(config)# int gig0/0.1
R1(config - subif)# encapsulation dot1q 5
R1(config - subif)# ip address 192.168.5.100 255.255.255.0

R1(config)# int gig0/0.2
R1(config - subif)# encapsulation dot1q 10
R1(config - subif)# ip address 192.168.10.100 255.255.255.0

R1(config)# int gig0/0.15
R1(config - subif)# encapsulation dot1q 15
R1(config - subif)# ip address 192.168.15.100 255.255.255.0
```

PART 3: Verify Connectivity

Step 1: Verify connectivity between C2 (VLAN 10) and C3 (VLAN 10).

```
C2> ping 192.168.10.2
(Successful)
```

Step 2: Verify connectivity between C2 (VLAN 10) and D1 (VLAN 5).

```
PC2> ping 192.168.5.3
(Successful)
```

PART 4: CREATE A REDUNDANT LINK BETWEEN SW-1 AND SW-2

STEP 1:Connect SW-1 and SW-2

Using a crossover cable,connect port Fa0/23on SW-1 to port Fao/23 on SW-2

STEP-2:Enable trunking,including all trunk security mechanisms on the link between SW-1 and SW-2

Execute command on SW-1 and SW-2

```
SW1/2(config)#int fa0/23
SW1/2(config-if)#switchport mode trunk
SW1/2(config-if)#switchport trunk native vlan 15
SW1/2(config-if)#switchport no negotiate
```

PART 5:ENABLE VLAN 20 AS A MANAGEMENT VLAN

STEP 1:Enable a management VLAN(VLAN 20) on SW-A

```
SW-A(config)#vlan 20
SW-A(config-if)#exit
SW-A(config)#int vlan 20
SW-A(config-if)#ip address 192.168.20.1 255.255.255.0
```

STEP 2:Enable the same management VLAN on all other switches

Execute command on SW-B,SW-1,SW-2 and central

```
SW(config)#vlan 20
SW(config-vlan)#exit
```

Create an interface VLAN 20 on all switches and assign an IP Address within the 192.168.20/24 network.

```
SW-B(config)#int vlan 20
SW-B(config-if)#ip address 192.168.20.2 255.255.255.0
```

```
SW-1(config)#int vlan 20
SW-1(config-if)#ip address 192.160.20.3 255.255.255.0
```

```
SW-2(config)#int vlan 20
SW-2(config-if)#ip address 192.168.20.4 255.255.255.0
```

```
central(config)#int vlan 20
central(config-if)#ip address 192.168.20.5 255.255.255.0
```

STEP 3: Connect and configure the management PC.Connect the managemnet PC to SW-A port Fa0/1 and ensure that it is assigned and available Ip address 192.168.20.50

STEP 4:On SW-A, ensure the management PC is part of VLAN 20.

```
SW-A(config)#int fa0/1
SW-A(config-if)#switchport mode access
SW-A(config-if)#switchport access vlan 20
```

STEP 5: Verify connectivity of the management PC to all switches

```
C1>ping 192.168.20.1(SW-A)
(successful)
C1>ping 192.168.20.2(SW-B)
(successful)
C1>ping 192.168.20.3(SW-1)
(successful)
C1>ping 192.168.20.4(SW-2)
(successful)
C1>ping 192.168.20.5(Central)
(successful)
```

PART 6:ENABLE THE MANAGEMENT PC TO ACCESS ROUTER R1

STEP 1:Enable a new subinterface on router R1.

```
R1(config)#int gig0/0.3
R1(config-subif)#encapsulation dotlq 20
R1(config-subif)#ip address 192.168.20.100 255.255.255.0
```

STEP 2:Set default gateaway in management PC.

C1-192.168.20.100

Step 3: Verify connectivity between the management PC and R1.

```
C1> ping 192.168.20.100
(Successful)
```

Step 4: Enable security.

```
R1(config)# access-list 101 deny ip any 192.168.20.0 0.0.0.255
R1(config)# access-list 101 permit ip any any
R1(config)# access-list 102 permit ip host 192.168.20.50 any
```

Step 5: Apply ACL on correct interfaces

```
R1(config)# int gig0/0.1
R1(config-subif)# ip access-group 101 in
R1(config-subif)# int gig0/0.2
R1(config-subif)# ip access-group 101 in
R1(config-subif)# line vty 0 4
R1(config-line)# access-class 102 in
```

Step 6: Verify security

```
C1>ssh-1 admin 192.168.20.100
Password:
R1>exit.
```

Step 7: Verify connectivity between the management PC and SW-A, SW-B and R1

```
C1> ping 192.168.20.1 (SW-A)
(Successful)
C1> ping 192.168.20.2 (SW-B)
(Successful)
C1> ping 192.168.20.100 (R1)
(Successful)
```

Step 8: Verify connectivity between the D1 and management PC.

```
D1>ping 192.168.20.50
(Unsuccessful – Destination host unreachable)
```

PRACTICAL 9: SITE TO SITE IPSEC VPN USING CLI

OBJECTIVES:-

*verify connectivity throughout the network

*configure R1 to support a site-to-siteIPsec VPN with R3

PART1: CONFIGURE ROUTER

STEP1: configure secret on router

(Execute command on all router)

```
R(config)#enable secret enpa55
```

STEP2: configure console password on router

(Execute command on all router)

```
R(config)#line console 0
R(config)#password conpa55
R(config-line)#login
```

STEP3: configure SSH login on router

```
R(config)#ip domain-name ccnasecurity.com
R(config)#username admin secret adminpa55
R(config)#line vty 0 4
R(config-line)#login local
R(config)#crypto key generate rsa
How many bits [512]: 1024
```

STEP4: configure ospf on router

```
R1(config)#router ospf 1
R1(config)#network 192.168.1.0 0.0.0.255 area 0
R1(config)#network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config)router ospf 1
R2(config)#network 192.168.2.0 0.0.0.255 area 0
R2(config)#network 10.2.2.0 0.0.0.3 area 0
R2(config)#network 10.1.1.0 0.0.0.3 area 0
```

```
R3(config)#router ospf 1
R3(config)#network 192.168.3.0 0.0.0.255 area 0
R3(config)#network 10.2.2.0 0.0.0.3 area 0
```

STEP5: verify connectivity

From PCA verify connectivity

```
PCA>ping 192.168.3.3 (success)
PCA>ping 192.168.2.3 (success)
PCB>ping 192.168.3.3 (success)
```

PART2: CONFIGURE IPsec PARAMETERS ON R1

STEP1: check if seccurity technology package is enable

```
R1#show version
```

(output)

```
Technology package license information for module "c1900"
Technology      Technology-Package              Technology-Package
                current type           next     reboot
ipbase          ipbasek9        permanent       ipbasek9
security        none            none            none
data            none            none            none
```

STEP2: enable security technology package

```
R1(config)#license boot module c1900 technology-package securityk9
```

STEP3: save the running config and reload router to enable security license

```
R1#copy run start
R1#reload
```

STEP4: verify security technology package is enabled

```
R1#show version
```

(output)

```
Technology package license information for module "c1900"
Technology      Technology-Package              Technology-Package
                current type           next     reboot
ipbase          ipbasek9        permanent       ipbasek9
security        securityk9      evaluation      ecurityk9
data            none            none            none
```

STEP5: identify interesting traffic on R1

```
R1(config)#access-list 110 permit ip 192.168.1.0 0.0.0.255 192.68.3.0 0.0.0.255
```

STEP6: configure IKE phase 1 ISAKMP policy on R1

```
R1(config)#crypto isakmp policy 10
R1(config-isakmp)#encryption aes 256
R1(config-isakmp)#authentication pre-share
R1(config-isakmp)#group 5
R1(config-isakmp)#exit
R1(config)#crypto isakmp key vpnpass address 10.2.2.2
```

STEP7: configure IKE Phase 2 IPsec Policy on R1

```
R1(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
R1(config)#crypto map VPN-MAP 10 ipsec-isakmp
R1(config-crypto-map)#description VPN connection to R3
R1(config-crypto-map)#set peer 10.2.2.2
R1(config-crypto-map)#set transform-set VPN-SET
R1(config-crypto-map)#match address 110
R1(config-crypto-map)#exit
```

STEP8: configure crypto map outgoing interface

```
R1(config)#int se0/1/0
R1(config-if)#crypto map VPN-MAP
```

PART3: CONFIGURE IPsec PARAMETER ON R3

STEP1: check if security technology package is enabled

```
R3#show version
```

(output)

```
Technology package license information for module "c1900"
Technology       Technology-Package                 Technology-Package
                 current type             next      reboot
ipbase           ipbasek9         permanent         ipbasek9
security         none             none              none
data             none             none              none
```

STEP2: enable security technology package

```
R3(config)#license boot module c1900 technology-package securityk9
```

STEP3: save running config of reload router to enable security license

```
R3#copy run start
R3#reload
```

STEP4: verify security technology package is enabled

```
R3#show version
(output)

Technology package license information for module "c1900"
Technology      Technology-Package              Technology-Package
                current type            next    reboot
ipbase          ipbasek9        permanent       ipbasek9
securitysecurityk9      solution        ecurityk9
data            none            none            none
```

STEP5: Configgure router R3 to support a site-to-site VPN with R1

```
R3(config)#access-list 110 permit ip 192.168.0.0 0.0.0.255 192.168.1.0 0.0.0.255
```

STEP6: configure IKE phase 1 ISAKMP properties on R3

```
R3(config)#crypto isakmp policy 10
R3(config-isakmp)#encryption aes 256
R3(config-isakmp)#authentication pre-share
R3(config-isakmp)#group 5
R3(config-isakmp)#exit
R3(config)#crypto isakmp key vpnpa55 address 10.1.1.2
```

STEP7: configure the IkE phase 2 IPsec policy on R3

```
R3(config)#crypto ipsec transform-set VPN-SET esp-aes esp-sha-hmac
R3(config)#crypto map VPN-MAP 10 ipsec-isakmp
R3(config-crypto-map)#description VPN connection to R1
R3(config-crypto-map)#set peer 10.1.1.2
R3(config-crypto-map)#set transform-set VPN-SET
```

```
R3(config-crypto-map)#match address 110
R3(config-crypto-map)#exit
```

STEP8: configure crypto map on outgoing interface

```
R3(config)#int se0/1/0
R3(config-if)#crypto map VPN-MAP
```

PART4: VERIFY THE IPsec VPN

STEP1: verify the tunnel prior to interestng traffic

```
R1#show crypto ipsec sa
```

(output)

```
#pkts encaps:0, #pkts encrypt:0, #pkts digest:0
#pkts decaps:0, #pkts decrypt:0, #pkts verify:0
```

STEP2: create interesting traffic

```
PCC>ping 192.168.1.3 (success)
```

STEP3: verify tunnel after interesting traffic

```
R1#show crypto ipsec sa
```

(output)

```
#pkts encaps:4, #pkts encrypt:4, #pkts digest:0
#pkts decaps:4, #pkts decrypt:4, #pkts verify:0
```

STEP4: create uninteresting traffic

```
PCB>ping 192.168.1.3 (success)
R1#ping 192.168.3.3 (success)
R1#ping 192.168.1.3 (success)
```

STEP5: verify tunnel

```
R1#show crypto ipsec sa
#pkts encaps:4, #pkts encrypt:4, #pkts digest:0
#pkts decaps:4, #pkts decrypt:4, #pkts verify:0
```

PRACTICAL 10:CONFIGURE ASA BASIC SETTINGS AND FIREWALL

OBJECTIVE:

*verify connectivity and explore the ASA

*configure basic ASA settings and interface security

*configure routing,address translation,and interace security level using CLI.

*configure DHCP,AAA and SSH

*configure DMZ,static NAT and ACLs.

PART 1: CONFIGURE ROUTER

STEP 1: configure secret on router

Execute command on all routers

```
R(config)#enable secret enpa55
```

STEP 2:configure console password on router

Execute command on all router

```
R(config)#line console 0
R(config-line)#password conpa55
R(config-line)#login
```

STEP 3:configure SSH login on router

Execute command on all router

```
R(config)#ip domain-name ccnasecurity.com
R(config)#username admin password adminpa55
R(config)#line vty 0 4
R(config-line)#login local
R(config)#crypto key generate rsa
How many bits in the module [512]:1024
```

STEP 4:Configure OSPF or router

```
R1(config)#router ospf 1
R1(config-router)#network 209.165.200.0 0.0.0.7 area 0
R1(config-router)#network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config)#router ospf 1
R2(config-router)#network 10.1.1.0 0.0.0.3 area 0
```

```
R2(config-router)#network 10.2.2.0 0.0.0.3 area 0
```

```
R3(config)#router ospf 1
R3(config-router)#network 172.16.3.0 0.0.0.255 area 0
R3(config-router)#network 10.2.2.0 0.0.0.3 area 0
```

Step 5:Verify connectivity

send packet from:

PCC->R1,R2,R3

(successful)

send packet from:

PCC->ASA,PC-B, DMZ server

(unsucessful)

PART 2: EXPLORE THE ASA

Step 1:Determine the ASA version,interfaces and license.

Enter privileged EXEC mode

```
ASA#en
```

a password has not been set

Press enter when promoted for a password

```
ASA#show version
```

Hardware:ASA5505,512 MB,RAM,CPU,Geode 500 Mhz

Internal ATA Compact flash ,D8MB

Step 2:Determine the file system and contents of the flash memory

```
ASA#show file system
```

PART 3: CONFIGURE ASA SETTINGS AND INTERFACE SECURITY

Step 1:Configure the hostname and domain name

```
ASA(config)#hostname CCNAS-ASA
CCNA-ASA(config)#domain-name ccnasecurity.com
```

Step 2:Configure the enable mode password

```
CCNA-ASA(config)#enable password enpa55
```

Step 3:Set the date and time

```
CCNAS-ASA(config)#clock set hr:min:sec date:month:year
```

Step 4:Configure the inside and outside interfaces

```
CCNAS-ASA(config)#int vlan 1
CCNAS-ASA(config-if)#nameif inside
CCNAS-ASA(config-if)#ip address 192.168.1.1 255.255.255.0
CCNAS-ASA(config-if)#security-level 100
CCNAS-ASA(config-if)#int vlan 2
CCNAS-ASA(config-if)#nameif outside
CCNAS-ASA(config-if)#ip address 209.165.200.226 255.255.255.248
CCNAS-ASA(config-if)#security-level 0
```

Step 5:Check the Configurations

```
CCNAS-ASA# show int ip brief

Interface                IP address         ok?         method      status       protocol
Ethernet0/0        unassigned        YES          unset       up                 up
Ethernet0/1        unassigned        YES          unset       up                 up
Ethernet0/2        unassigned        YES          unset       up                 up
Ethernet0/3        unassigned        YES          unset       down         down
Ethernet0/4        unassigned        YES          unset       down         down
Ethernet0/5        unassigned        YES          unset       down         down
Ethernet0/6        unassigned        YES          unset       down         down
Ethernet0/7        unassigned        YES          unset       down         down
vlan 1             192.168.1.1       YES          unset       up                 up
vlan 2             209.165.200.226   YES          unset       up                 up
```

```
CCNAS-ASA#show ip address

system ip address

Interface          name        ip address         subnet mask       method
vlan 1      inside       192.168.1.1        255.255.255.0     manual
vlan 2          outside         209.165.200.226         255.255.255.248         manual

CCNAS-ASA#show switch vlan


vlan                    name            status          ports
1                       inside          up                      Et0/1,Et0/2,Et0/3,Et0/4,Et0/5,Et0/6,Et0/7
2                       outside         up                      Et0/0
```

Step 3:Test connectivity to the ASA(send packets)

PCB-> ASA

(successful)

PCB->R1

(unsuccessful)

PART 4: CONFIGURE ROUTING,ADDRESS TRANSACTION AND INSPECTION POLICY

step 1: configure a static default router for the ASA

```
CCNAS-ASA #show route


C 192.168.1.0  255.255.255.0 directly connected inside vlan1 205.165.200.0129 is subnetted, 2 subnets


C 209.165.200.0 255.255.255.248 is directly connected outside vlan2


C 209.165.200.224 255.255.255.248 is directly connected outside vlan2


CCNAS-ASA(config)#route outside 0.0.0.0 0.0.0.0 201.165.200.225
CCNAS-ASA #show route


C 192.168.1.0  255.255.255.0 directly connected inside vlan1 209.165.200.0129 is subnetted, 2 subnets


C 209.165.200.0 255.255.255.248 is directly connected inside vlan1


C 209.165.200.224 255.255.255.248 is directly connected outside vlan2


st 0.0.010 (110) via 209.168.200.226
```

step 2: Test connectivity (send packets)

ASA->R1

(successful)

step 3: configure address translation using PAT & network objects.

```
CCANAS-ASA(config)#object network include-net
CCNAS-ASA(config-network-object)#subnet 192.168.1.0 255.255.255.0
CCNAS-ASA(config-network-object)#not (inside,outside) dynamic interface
CCNA-ASA(config-network-object)#end
```

step 4: Test connectivity

```
CCNA-ASA #show run

object network inside-net
subnet192.168.1.0 255.255.255.0
```

PCB->R1(send packets)

(Unsuccessful)

```
CCNA-ASA #show nat

Auto NAT policies (section2)
(inside) to (outside) source dyanamic inside-net interface
translate-hits=1,untranslate-hits=1
```

step 5: Modify the default MPF application inspection global service policy

```
CCNAS-ASA(config)#class-nap inspection-default
CCNAS-ASA(config-map)#match default-inspection traffic
CCNAS-ASA(config-map)#exit
CCNAS-ASA(config)#poicy-map global policy.
CCNAS-ASA(config-pmap)#class inspection default
CCNAS-ASA(config-pmap-c)#inspect icmp
CCNAS-ASA(config-pmap-c)#exit
CCNAS-ASA(config)#service policy global policy global
```

step 6: Test connectivity(send packets)

PCB->R1

(successful)

PART 5:CONFIGURE DHCP,AAA AND SSH

step 1: configure the ASA as a DHCP server.

```
(CCNAS-ASA(config)#dhcpd address 192.168.1.5- 192.168.1.36 inside
CCNAS-ASA(config)#dhcpd dns 209.165.201.2 int inside
CCNAS-ASA(config)#dhcpd enable inside
CCNAS-ASA(config)#dhcpd enable inside
```

change PC-B from a static IP addresses to a DHCP client,and verify that it receives IP addressing information.

step 2: comfigure AAA to use the local database for authentication

```
CCNAS-ASA(config)#username admin password adminpa55
CCNAS-ASA(config)#aaa authentication ssh console LOCAL.
```

step 3:configure remote access to the ASA

```
CCNAS-ASA(config)#crypto key generate rsa modulus 1024
Do you really want to replace them?[yes/no]:no
CCNAS-ASA(config)#ssh 192.168.1.0 255.255.255.0 inside
CCNAS-ASA(config)#ssh 172.16.3.3 255.255.255.255 outside
CCNAS-ASA(config)#ssh timeout 10.
```

step 4: verify ssh session

```
PCB > ssh-1 admin 192.168.1.1
Password : adminpa55
CCNA-ASA > exit
PCC> ssh-1 admin 209.168.200.226
Password: adminpa55
CCNAS-ASA > exit
```

PART 6:CONFIGURE A DMZ,STATIC NAT AND ACLS

step 1: configure the DMZ interface VLAN3 on the ASA.

```
CCNAS-ASA(config)#int vlan3
CCNAS-ASA(config-if)#ip address 192.168.2.1 255.255.255.0
```

```
CCNAS-ASA(config-if)#no forward int vlan1

CCNAS-ASA(config-if)#nameif dmz

CCNAS-ASA(config-if)#security-level 70

CCNAS-ASA(config-if)#int et0/2

CCNAS-ASA(config-if)#switchport access vlan3
```

step 2: check the configurations

```
CCNAS-ASA #show int ip brief


Interface           IP Address          ok?           method        status  protocol

Ethernet0/0         unassigned          YES     unset         up                  up

Ethernet0/1         unassigned          YES     unset         up                  up

Ethernet0/2         unassigned          YES     unset         up                  up

Ethernet0/3         unassigned          YES     unset         down                down

Ethernet0/4         unassigned          YES     unset         down                down

Ethernet0/5         unassigned          YES     unset         down                down

Ethernet0/6         unassigned          YES     unset         down                down

Ethernet0/7         unassigned          YES     unset         down                down

vlan1               192.168.1.1         YES     manual        up                  up

vlan2               109.165.200.226           YES     manual        up                  up

vlan3               192.168.2.1         YES     manual        up                  up


CCNAS-ASA #show ip address

system IP Addresses:

Interface     Name          IP Address              subnet mask         method

vlan1         inside  192.168.1.1           255.255.255.0   manual

vlan2         outside 209.165.200.226       255.255.255.0   manual

vlan3         dmz           192.168.2.1             255.255.255.0   manual


CCNAS-ASA #show switch vlan


VLAN    Name          status  Port

1       inside              up            Et0/1,Et0/3,Et0/4,Et0/5,Et0/6,Et0/7

2       outside             up            Et0/0

3       dmz                       up            Et0/2
```

step 3: configure static NAT to the DMZ server using a network object

```
CCNAS-ASA(config)#object network dmz_server
CCNAS-ASA(config-network-object)#host 192.168.2.3
CCNAS-ASA(config-network-object)#nat(dmz,outside)static 209.165.200.227
CCNAS-ASA(config-network-object)#exit
```

step 4: Configure an ACL to allow access to the DMZ from the internet

```
CCNAS-ASA(config)#access-list OUTSIDE-DMZ permit icmp any host 192.168.2.3
CCNAS-ASA(config)#access-list OUTSIDE-DMZ permit tcp any host 192.168.2.3 eq80
CCNAS-ASA(config)#access-group OUTSIDE-DHL in int outside.
```

step 5: Test access to the DMZ server.

The ability to successfully test outside to the DMZ web server was not in place,therefore,successful testing is not required.