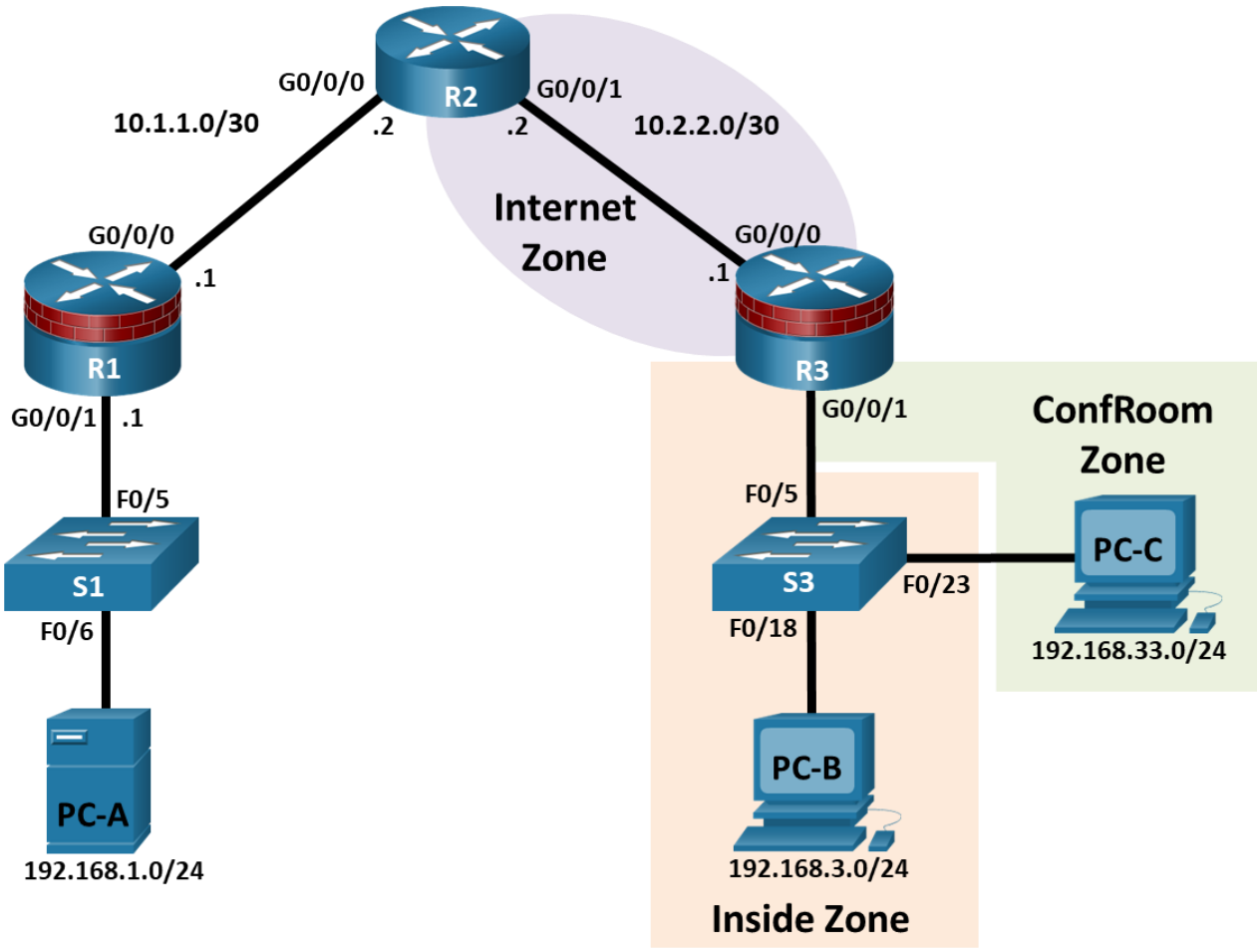
Lab - Configure ZPFs

Name: Toff Darell Vergara Date: October 20, 2025

Section Code: T135

Topology



IP Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0/0	10.1.1.1	255.255.255.252	N/A
	G0/0/1	192.168.1.1	255.255.255.0	N/A

Device	Interface	IP Address	Subnet Mask	Default Gateway
R2	G0/0/0	10.1.1.2	255.255.255.252	N/A
R2	G0/0/1	10.2.2.2	255.255.255.252	N/A
R3	G0/0/0	10.2.2.1	255.255.255.252	N/A
R3	G0/0/1.3	192.168.3.1	255.255.255.0	N/A
R3	G0/0/1.33	192.168.33.1	255.255.255.0	N/A
PC-A	NIC	192.168.1.3	255.255.255.0	192.168.1.1
PC-B	NIC	192.168.3.3	255.255.255.0	192.168.3.1
PC-C	NIC	192.168.33.3	255.255.255.0	192.168.33.1

Blank Line, No additional information

Objectives

Part 1: Basic Device Configuration

- Configure host names, interface IP addresses, and access passwords on routers.
- Configure the static routes to enable end-to-end connectivity on routers.
- Configure access and trunk ports on a switch.

Part 2: Configuring a Zone-Based Policy Firewall (ZPF)

- Use the CLI to configure a Zone-Based Policy Firewall.
- Use the CLI to verify the configuration.

Part 3: Verify ZPF Firewall Functionality

Background

The most basic form of a Cisco IOS firewall uses access control lists (ACLs) to filter IP traffic and monitor established traffic patterns. A traditional Cisco IOS firewall is an ACL-based firewall.

The newer Cisco IOS Firewall implementation uses a zone-based approach that operates as a function of interfaces instead of access control lists. A Zone-Based Policy Firewall (ZPF) allows different inspection policies to be applied to multiple host groups connected to the same router interface. It can be configured for extremely advanced, protocol specific, granular control. It prohibits traffic via a default deny-all policy between different firewall zones. ZPF is suited for multiple interfaces that have similar or varying security requirements.

In this lab, you build a multi-router network, configure the routers and PC hosts, and configure a Zone-Based Policy Firewall using the Cisco IOS command line interface (CLI).

Note: The routers used with hands-on labs are Cisco 4221 with Cisco IOS XE Release 16.9.4 (universalk9 image). The switches used in the labs are Cisco Catalyst 2960+ with Cisco IOS Release 15.2(2) (lanbasek9 image). Other routers, switches, and Cisco IOS versions can be used. Depending on the model and Cisco IOS version, the commands available and the output produced might vary from what is shown in the labs. Refer to the Router Interface Summary Table at the end of the lab for the correct interface identifiers.

Note: Make sure that the routers and switches have been erased and have no startup configurations.

Required Resources

- 3 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 2960+ with Cisco IOS Release 15.2(7) lanbasek9 image or comparable)

- 3 PCs (Windows OS with a terminal emulation program, such as Tera Term or PuTTy installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Basic Device Configuration

In this part of this lab, you set up the network topology and configure basic settings, such as the interface IP addresses, static routing, device access, and passwords.

Note: All tasks should be performed on routers R1, R2, and R3. The procedures are shown for only one of the routers.

Step 1: Cable the network as shown in the topology.

Attach the devices as shown in the topology diagram, and cable as necessary.

Step 2: Disable DNS lookup.

Open configuration window

To prevent the router from attempting to translate incorrectly entered commands, disable DNS lookup.

Step 3: Configure basic settings for each router.

a. Configure host names as shown in the topology.

b. Configure the interface IP addresses as shown in the IP addressing table. The IP address configuration for router R3 is provided below.

```
R3(config)# interface GigabitEthernet0/0/0
R3(config-if)# ip address 10.2.2.1 255.255.255.252
R3(config-if)# no shutdown
R3(config-if)# interface GigabitEthernet0/0/1
R3(config-if)# no shutdown
R3(config-if)# interface GigabitEthernet0/0/1.3
R3(config-if)# encapsulation dot1Q 3
R3(config-if)# ip address 192.168.3.1 255.255.255.0
R3(config-if)#interface GigabitEthernet0/0/1.33
R3(config-if)# encapsulation dot1Q 33
R3(config-if)# ip address 192.168.33.1 255.255.255.0
```

Step 4: Configure static routes on R1, R2, and R3.

a. To achieve end-to-end IP reachability, proper static routes must be configured on R1, R2 and R3. R1 and R3 are stub routers, and as such, only need a default route pointing to R2. R2, behaving as the ISP, must know how to reach R1's and R3's internal networks before end-to-end IP reachability is achieved. Below is the static route configuration for R1, R2 and R3. On R1, use the following command:

```
R1(config)# ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

b. On R2, use the following commands.

```
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
R2(config)# ip route 192.168.33.0 255.255.255.0 10.2.2.1
```

c. On R3, use the following command.

```
R3(config)# ip route 0.0.0.0 0.0.0.0 10.2.2.2
```

Step 5: Configure S3.

a. Configure trunk link:

```
S3(config)# interface f0/5
S3(config-if)# switchport mode trunk
```

b. Configure access ports.

```
S3(config)# interface f0/18
S3(config-if)# switchport mode access
S3(config-if)# switchport access vlan 3
S3(config-if)# interface f0/23
S3(config-if)# switchport mode access
S3(config-if)# switchport access vlan 33
```

Step 6: Configure PC host IP settings.

Configure a static IP address, subnet mask, and default gateway for PC-A, PC-B, and PC-C as shown in the IP addressing table.

Step 7: Verify basic network connectivity.

a. Ping from R1 to R3.

If the pings are not successful, troubleshoot the basic device configurations before continuing.

b. Ping from PC-A on the R1 LAN to PC-B and PC-C on the R3 LANs.

If the pings are not successful, troubleshoot the basic device configurations before continuing.

Note: If you can ping from PC-A to PC-C, you have demonstrated that the end-to-end IP reachability has been achieved. If you cannot ping but the device interfaces are UP and IP addresses are correct, use the **show interface**, **show ip interface,** and **show ip route** commands to help identify problems.

Step 8: Configure a user account, encrypted passwords and crypto keys for SSH.

Note: Passwords in this task are set to a minimum of 10 characters, but are relatively simple for the benefit of performing the lab. More complex passwords are recommended in a production network.

a. Configure a minimum password length using the **security passwords** command to set a minimum password length of 10 characters.

```
R1(config)# security passwords min-length 10
```

b. Configure a domain name.

```
R1(config)# ip domain-name netsec.com
```

c. Configure crypto keys for SSH

```
R1(config)# crypto key generate rsa general-keys modulus 1024
```

d. Configure an admin01 user account using **algorithm-type scrypt** for encryption and a password of cisco12345.

```
R1(config)# username admin01 algorithm-type scrypt secret cisco12345
```

e. Configure line console 0 to use the local user database for logins. For additional security, the **exec-timeout** command causes the line to log out after **5** minutes of inactivity. The **logging synchronous** command prevents console messages from interrupting command entry.

Note: To avoid repetitive logins during this lab, the **exec-timeout** command can be set to **0 0**, which prevents it from expiring; however, this is not considered to be a good security practice.

```
R1(config)# line console 0
R1(config-line)# login local
R1(config-line)# exec-timeout 5 0
R1(config-line)# logging synchronous
```

f. Configure line aux 0 to use the local user database for logins.

```
R1(config)# line aux 0
R1(config-line)# login local
R1(config-line)# exec-timeout 5 0
```

g. Configure line vty 0 4 to use the local user database for logins and restrict access to SSH connections only.

```
R1(config)# line vty 0 4
R1(config-line)# login local
R1(config-line)# transport input ssh
R1(config-line)# exec-timeout 5 0
```

h. Configure the enable password with strong encryption.

```
R1(config)# enable algorithm-type scrypt secret class12345
```

Step 9: Save the basic running configuration for all three routers.

Save the running configuration to the startup configuration from the privileged EXEC prompt.

```
R1# copy running-config startup-config
```

Close configuration window

Part 2: Configuring a Zone-Based Policy Firewall (ZPF)

In this part, you will create a zone-based policy firewall on R3 using the command line interface (CLI), making it act not only as a router but also as a firewall. R3 is currently responsible for routing packets for the three networks connected to it. R3's interface roles are configured as follows:

G0/0/0 is connected to the Internet. Because this is a public network, it is considered an untrusted network and should have the lowest security level.

G0/0/1.3 is connected to the internal network. Only authorized users have access to this network. In addition, vital institution resources also reside in this network. The internal network is to be considered a trusted network and should have the highest security level.

G0/0/1.33 is connected to a conference room. The conference room is used to host meetings with people who are not part of the organization.

The security policy to be enforced by R3 when it is acting as a firewall dictates that:

- No traffic initiated from the Internet should be allowed into the internal or conference room networks.
- Returning Internet traffic (return packets coming from the Internet into the R3 site, in response to requests originating from any of the R3 networks) should be allowed.
- Computers in the R3 internal network are considered *trusted* and are allowed to initiate any type traffic (TCP, UDP or ICMP based traffic).

- Computers in the R3 conference room network are considered *untrusted* and are allowed to initiate only web traffic (HTTP or HTTPS) to the Internet.
- No traffic is allowed between the internal network and the conference room network. There is no guarantee regarding the condition of guest computers in the conference room network. Such machines could be infected with malware and might attempt to send out spam or other malicious traffic.

Step 1: Verify end-to-end network connectivity.

In this step, you will verify end-to-end network connectivity before implementing ZPF.

Open configuration window

a. Ping from R1 to R3 using both of R3's G0/0/1 interface IP addresses (192.168.3.1 and 192.168.33.1).

   If the pings are not successful, troubleshoot the basic device configurations before continuing.

b. Ping from PC-A on the R1 LAN to PC-C on the R3 conference room LAN.

   If the pings are not successful, troubleshoot the basic device configurations before continuing.

c. Ping from PC-A on the R1 LAN to PC-B on the R3 internal LAN.

   If the pings are not successful, troubleshoot the basic device configurations before continuing.

Step 2: Display the R3 running configurations.

In this step, you will verify R3 running configurations before implementing ZPF.

a. Issue the **show ip interface brief** command on R3 to verify the correct IP addresses were assigned. Use the Address Table to verify the addresses.

b. Issue the **show ip route** command on R3 to verify it has a static default route pointing to R2's G0/0/1 interface.

c. Issue the **show run** command to review the current basic configuration on R3.

```
R3#show ip in
R3#show ip int
R3#show ip interface be
R3#show ip interface br
Interface              IP-Address      OK? Method Status                Protocol
GigabitEthernet0/0/0   10.2.2.1        YES manual up                    up
GigabitEthernet0/0/1   unassigned      YES unset  up                    up
GigabitEthernet0/0/1.3 192.168.3.1     YES manual up                    up
GigabitEthernet0/0/1.33192.168.33.1    YES manual up                    up
GigabitEthernet0/0/2   unassigned      YES unset  administratively down down
Vlan1                  unassigned      YES unset  administratively down down
R3#
```

```
R3#show ip rou
R3#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default, U - per-user static route, o - ODR
       P - periodic downloaded static route

Gateway of last resort is 10.2.2.2 to network 0.0.0.0

     10.0.0.0/8 is variably subnetted, 2 subnets, 2 masks
C       10.2.2.0/30 is directly connected, GigabitEthernet0/0/0
L       10.2.2.1/32 is directly connected, GigabitEthernet0/0/0
     192.168.3.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.3.0/24 is directly connected, GigabitEthernet0/0/1.3
L       192.168.3.1/32 is directly connected, GigabitEthernet0/0/1.3
     192.168.33.0/24 is variably subnetted, 2 subnets, 2 masks
C       192.168.33.0/24 is directly connected, GigabitEthernet0/0/1.33
L       192.168.33.1/32 is directly connected, GigabitEthernet0/0/1.33
S*   0.0.0.0/0 [1/0] via 10.2.2.2

R3#
```

```
!
no ip domain-lookup
ip domain-name netsec.com
!
!
spanning-tree mode pvst
!
class-map type inspect match-any INSIDE_PROTOCOLS
 match protocol tcp
 match protocol udp
 match protocol icmp
class-map type inspect match-any CONFROOM_PROTOCOLS
 match protocol http
 match protocol https
 match protocol dns
!
policy-map type inspect INSIDE_TO_INTERNET
 class type inspect INSIDE_PROTOCOLS
  inspect
!
policy-map type inspect CONFROOM_TO_INTERNET
 class type inspect CONFROOM_PROTOCOLS
  inspect
!
policy-map type inspect inside
 class type inspect class-default
  pass
!
!
!
zone security INSIDE
zone security CONFROOM
zone security INTERNET
zone-pair security INSIDE_TO_INTERNET source INSIDE destination INTERNET
 service-policy type inspect INSIDE_TO_INTERNET
zone-pair security CONFROOM_TO_INTERNET source CONFROOM destination INTERNET
 service-policy type inspect CONFROOM_TO_INTERNET
zone-pair security INSIDE_TO_INTERNET source INTERNET destination CONFROOM
zone-pair security INSIDE source INSIDE destination CONFROOM
 service-policy type inspect inside
!
!
interface GigabitEthernet0/0/0
 ip address 10.2.2.1 255.255.255.252
 zone-member security INTERNET
 duplex auto
 speed auto
!
interface GigabitEthernet0/0/1
 no ip address
```

Step 3: Creating the security zones.

A security zone is a group of interfaces with similar security properties and requirements. For example, if a router has three interfaces connected to internal networks, all three interfaces can be placed under the same zone named “internal”. Because all security properties are configured to the zone instead of to the individual router interfaces, the firewall design is much more scalable.

In this lab, the R3 site has three interfaces; one connected to an internal trusted network, one connected to the conference room network and another connected to the internet. Because all three networks have different security requirements and properties, we will create three different security zones.

Security zones are created in global configuration mode, and the command allows for zone name definition. In R3, create three zones named **INSIDE**, **CONFROOM** and **INTERNET**:

```
R3(config)# zone security INSIDE
R3(config)# zone security CONFROOM
R3(config)# zone security INTERNET
```

Step 4: Creating Security Policies

Before ZPF can decide if some specific traffic should be allowed or denied, it must be told *what* traffic is to be considered. Cisco IOS uses class-maps to select traffic. *Interesting traffic* is a common denomination for traffic that has been selected by a class-map.

While class-maps select traffic, it is not their job to decide what happens to the selected traffic; Policy-maps decide the *fate* of the selected traffic.

ZPF traffic policies are defined as policy-maps and use class-maps to select traffic. In other words, class-maps define *what* traffic is to be policed while policy-maps define the *action* to be taken upon the selected traffic.

Policy-maps can drop, pass or inspect traffic. Because we want the firewall to *watch* traffic moving in the direction of zone-pairs, we will create inspect policy-maps. Inspect policy-maps allow for dynamic handling of the return traffic.

First, you will create class-maps. After the class-maps are created, you will create policy-maps and attach the class-maps to the policy-maps.

a. Create an inspect class-map to match traffic to be allowed from the INSIDE zone to the **INTERNET** zone. Because we trust the INSIDE zone, we allow all the main protocols.

   In the commands below, the first line creates an inspect class-map. The **match-any** keyword instructs the router that any of the **match** protocol statements will qualify as a successful match resulting in a policy being applied. The result is a match for TCP or UDP or ICMP packets.

   The **match** commands refer to specific Cisco NBAR supported protocols. For more information, perform an internet search for Cisco NBAR.

```
R3(config)# class-map type inspect match-any INSIDE_PROTOCOLS
R3(config-cmap)# match protocol tcp
R3(config-cmap)# match protocol udp
R3(config-cmap)# match protocol icmp
```

b. Similarly, create a class-map to match the traffic to be allowed from the **CONFROOM** zone to the **INTERNET** zone. Because we do not fully trust the **CONFROOM** zone, we must limit what the server can send out to the Internet:

```
R3(config)# class-map type inspect match-any CONFROOM_PROTOCOLS
R3(config-cmap)# match protocol http
R3(config-cmap)# match protocol https
R3(config-cmap)# match protocol dns
```

c. Now that the class-maps are created, you can create the policy-maps.

In the commands below, the first line creates an inspect policy-map named **INSIDE_TO_INTERNET**. The second line binds the previously created **INSIDE_PROTOCOLS** class-map to the policy-map. All packets matched by the **INSIDE_PROTOCOLS** class-map will be subjected to the action taken by the **INSIDE_TO_INTERNET** policy-map. Finally, the third line defines the actual action this policy-map will apply to the matched packets. In this case, the matched packets will be inspected.

The next three lines creates a similar policy-map named **CONFROOM_TO_INTERNET** and attaches the **CONFROOM_PROTOCOLS** class-map.

The commands are as follows:

```
R3(config)# policy-map type inspect INSIDE_TO_INTERNET
R3(config-pmap)# class type inspect INSIDE_PROTOCOLS
R3(config-pmap-c)# inspect
R3(config)# policy-map type inspect CONFROOM_TO_INTERNET
R3(config-pmap)# class type inspect CONFROOM_PROTOCOLS
R3(config-pmap-c)# inspect
```

Step 5: Create the Zone Pairs

A zone pair allows you to specify a unidirectional firewall policy between two security zones.

For example, a commonly used security policy dictates that the internal network can initiate any traffic towards the Internet but no traffic originating from the Internet should be allowed to reach the internal network.

This traffic policy requires only one zone pair, **INTERNAL to INTERNET**. Because zone-pairs define unidirectional traffic flow, another zone-pair must be created if Internet-initiated traffic must flow in the **INTERNET to INTERNAL** direction.

Notice that Cisco ZPF can be configured to inspect traffic that moves in the direction defined by the zone pair. In that situation, the firewall *watches* the traffic and dynamically creates rules allowing the return or related traffic to flow back through the router.

To define a zone pair, use the **zone-pair security** command. The direction of the traffic is specified by the source and destination zones.

For this lab, you will create two zone-pairs:

INSIDE_TO_INTERNET: Allows traffic leaving the internal network towards the Internet.

CONFROOM_TO_INTERNET: Allows Internet access from the ConfRoom network.

a. Creating the zone-pairs:

```
R3(config)# zone-pair security INSIDE_TO_INTERNET source INSIDE destination
INTERNET
R3(config)# zone-pair security CONFROOM_TO_INTERNET source CONFROOM
destination INTERNET
```

b. Verify the zone-pairs were correctly created by issuing the **show zone-pair security** command. Notice that no policies are associated with the zone-pairs yet. The security policies will be applied to zone-pairs in the next step.

```
R3# show zone-pair security
```

```
R3#sho
R3#show z
R3#show zo
R3#show zone-[a
R3#show zone-pa
R3#show zone-pair se
R3#show zone-pair security
Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INSIDE  Destination-Zone INTERNET
    service-policy INSIDE_TO_INTERNET

Zone-pair name CONFROOM_TO_INTERNET
    Source-Zone CONFROOM  Destination-Zone INTERNET
    service-policy CONFROOM_TO_INTERNET

Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INTERNET  Destination-Zone CONFROOM
    service-policy not configured

Zone-pair name INSIDE
    Source-Zone INSIDE  Destination-Zone CONFROOM
    service-policy inside

R3#
```

Step 6: Applying Security Policies

a. As the last configuration step, apply the policy-maps to the zone-pairs:

```
R3(config)# zone-pair security INSIDE_TO_INTERNET
R3(config-sec-zone-pair)# service-policy type inspect INSIDE_TO_INTERNET
R3(config)# zone-pair security CONFROOM_TO_INTERNET
R3(config-sec-zone-pair)# service-policy type inspect CONFROOM_TO_INTERNET
```

b. Issue the **show zone-pair security** command once again to verify the zone-pair configuration. Notice that the service-polices are now displayed:

```
R3# show zone-pair security
```

```
R3#sho
R3#show z
R3#show zo
R3#show zone-[a
R3#show zone-pa
R3#show zone-pair se
R3#show zone-pair security
Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INSIDE  Destination-Zone INTERNET
    service-policy INSIDE_TO_INTERNET

Zone-pair name CONFROOM_TO_INTERNET
    Source-Zone CONFROOM  Destination-Zone INTERNET
    service-policy CONFROOM_TO_INTERNET

Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INTERNET  Destination-Zone CONFROOM
    service-policy not configured

Zone-pair name INSIDE
    Source-Zone INSIDE  Destination-Zone CONFROOM
    service-policy inside

R3#
```

c. To obtain more information about the zone-pairs, their policy-maps, the class-maps and match counters, use the **show policy-map type inspect zone-pair** command:

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

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

policy exists on zp INSIDE_TO_INTERNET
 Zone-pair: INSIDE_TO_INTERNET

  Service-policy inspect : INSIDE_TO_INTERNET

    Class-map: INSIDE_PROTOCOLS (match-any)
      Match: protocol tcp
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol udp
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol icmp
        0 packets, 0 bytes
        30 second rate 0 bps
      Inspect

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes

policy exists on zp CONFROOM_TO_INTERNET
 Zone-pair: CONFROOM_TO_INTERNET

  Service-policy inspect : CONFROOM_TO_INTERNET

    Class-map: CONFROOM_PROTOCOLS (match-any)
      Match: protocol http
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol https
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol dns
        0 packets, 0 bytes
        30 second rate 0 bps
      Inspect

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes

policy exists on zp INSIDE_TO_INTERNET
```

```
policy exists on zp CONFROOM_TO_INTERNET
 Zone-pair: CONFROOM_TO_INTERNET

  Service-policy inspect : CONFROOM_TO_INTERNET

    Class-map: CONFROOM_PROTOCOLS (match-any)
      Match: protocol http
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol https
        0 packets, 0 bytes
        30 second rate 0 bps
      Match: protocol dns
        0 packets, 0 bytes
        30 second rate 0 bps
      Inspect

    Class-map: class-default (match-any)
      Match: any
      Drop (default action)
        0 packets, 0 bytes

policy exists on zp INSIDE_TO_INTERNET
 Zone-pair: INSIDE_TO_INTERNET

  Service-policy inspect :

policy exists on zp INSIDE
 Zone-pair: INSIDE

  Service-policy inspect : inside

    Class-map: class-default (match-any)
      Match: any
      Pass

R3#
```

Step 7: Assign Interfaces to the Proper Security Zones

Interfaces (physical and logical) are assigned to security zones with the **zone-member security** interface command.

a. Assign R3's G0/0 to the **CONFROOM** security zone:

```
R3(config)# interface g0/0/1.33
R3(config-if)# zone-member security CONFROOM
```

b. Assign R3's G0/1 to the INSIDE security zone:

```
R3(config)# interface g0/0/1.3
R3(config-if)# zone-member security INSIDE
```

c. Assign R3's S0/0/1 to the **INTERNET** security zone:

```
R3(config)# interface g0/0/0
R3(config-if)# zone-member security INTERNET
```

Step 8: Verify Zone Assignment

a. Issue the show zone security command to ensure the zones were properly created, and the interfaces were correctly assigned:

```
R3# show zone security
zone self
  Description: System defined zone

zone service
  Description: System defined zone

zone INSIDE
  Member Interfaces:
    GigabitEthernet0/0/1.3

zone CONFROOM
  Member Interfaces:
    GigabitEthernet0/0/1.33

zone INTERNET
  Member Interfaces:
    GigabitEthernet0/0/0




R3# show zone security
```

b. Even though no commands were issued to create a “self” zone, the output above still displays it.

Question:
Why is R3 displaying a zone named “self”? What is the significance of this zone?

Type your answers here.

Close configuration window

Part 3: Verify ZPF Firewall Functionality

```
R3#show zo
R3#show zom
R3#show zone se
R3#show zone security
zone self
  Description: System defined zone


zone INSIDE
  Member Interfaces:
    GigabitEthernet0/0/2


zone CONFROOM
  Member Interfaces:
    GigabitEthernet0/0/1


zone INTERNET
  Member Interfaces:
    GigabitEthernet0/0/0


R3#
```

Step 1: Traffic originating on the Internet

a. To test the firewall’s effectiveness, ping PC-B from PC-A. In PC-A, open a command prompt and issue a ping to 192.168.3.3.

```
PC-A:\> ping 192.168.3.3
```

Question:
Was the ping successful? Explain.

Type your answers here. The self zone is a built-in security zone in Cisco routers.

b. Ping PC-C from PC-A. In PC-A, open a command window and ping 192.168.33.3.

```
PC-A:\> ping 192.168.33.3
```

Question:
Was the ping successful? Explain.

Type your answers here. No, the ICMP packets from PC-A entered R3 through its Serial0/0/1 interface

c. Ping PC-A from PC-B. In PC-B, open a command window and issue a ping to 192.168.1.3.

```
PC-B:\> ping 192.168.1.3
```

Question:
Was the ping successful? Explain.

Type your answers here. No, the ICMP packets from PC-A went through R3’s Serial0/0/1 interface

d. Ping PC-A from PC-C. In PC-C, open a command window and ping 192.168.1.3

```
PC-C:\> ping 192.168.1.3
```

Question:

Was the ping successful? Explain.

Type your answers here. Yes, the ICMP packets from PC-B entered R3 using its G0/0/1 interface. Since G0/0/1.3 is part of the INSIDE zone, that's correct.

Step 2: The Self Zone Verification

a. From PC-A ping R3's G0/0/1.3 interface:

```
PC-A:\> ping 192.168.3.1
```

Question:

Was the ping successful? Is this the correct behavior? Explain.

Type your answers here. No, the ICMP packets from PC-C passed through R3's G0/0 interface. G0/0/1.33 is assigned to the CONFROOM zone

b. From PC-C ping R3's G0/0/1.3 interface:

```
PC-C:\> ping 192.168.3.1
```

Question:

Was the ping successful? Is this the correct behavior? Explain.

Type your answers here. Yes, the ping worked successfully, so the behavior is correct

Challenge (optional)

Create the proper zone-pair, class-maps, and policy-maps and configure R3 to prevent Internet originating traffic from reaching the Self Zone.

Appendix – Multiple Interfaces under the Same Zone (optional)

One benefit of ZPF firewalls is that they scale well compared to the classic firewall. If a new interface with the same security requirements is added to the firewall, the administrator can simply add the new interface as a member of an existing security zone. However, some IOS versions will not allow devices connected to different interfaces of the same zone to communicate by default. In those cases, a zone-pair must be created using the same zone as source and destination.

Traffic between similarly zoned interfaces will always be bidirectional due to the fact that the zone-pair's source and destination zones are the same. Because of that, there is no need to inspect traffic to allow for automatic return traffic handling; return traffic will always be allowed because it will always conform to the zone-pair definition. In this case, the policy-map should have a **pass** action instead of **inspect**. Because of the **pass** action, the router will not inspect packets matched by the policy-map, it will simply forward it to its destination.

In the context of this lab, if R3 had a G0/0/1.2 interface also assigned to the INSIDE zone, and the router IOS version did not support allowing traffic between interfaces configured to the same zone, the extra configuration would look like this:

New zone-pair: **Inside to Inside**; allows routing of traffic among the internal trusted interfaces.

Creating the policy-map (notice that no explicit class-map is needed because we use the default "catch-all" class):

```
R3(config)# policy-map type inspect inside
R3(config-pmap)# class class-default
R3(config-pmap-c)# pass
```

Creating the zone-pair and assigning the new policy-map to it. Notice that the INSIDE zone is both the source and the destination of the zone-pair:

```
R3(config)# zone-pair security INSIDE source INSIDE destination INSIDE
R3(config-sec-zone-pair)# service-policy type inspect inside
```

```
R3#show zone-pair security
Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INSIDE  Destination-Zone INTERNET
    service-policy INSIDE_TO_INTERNET

Zone-pair name CONFROOM_TO_INTERNET
    Source-Zone CONFROOM  Destination-Zone INTERNET
    service-policy CONFROOM_TO_INTERNET

Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INTERNET  Destination-Zone CONFROOM
    service-policy not configured

Zone-pair name INSIDE
    Source-Zone INSIDE  Destination-Zone CONFROOM
    service-policy inside

R3#
```

To verify the existence of the new pair, use **show zone-pair security:**

```
R3# show zone-pair security
Zone-pair name INSIDE_TO_INTERNET
    Source-Zone INSIDE Destination-Zone INTERNET
    service-policy INSIDE_TO_INTERNET
Zone-pair name CONFROOM_TO_INTERNET
    Source-Zone CONFROOM  Destination-Zone INTERNET
    service-policy CONFROOM_TO_INTERNET
Zone-pair name INSIDE
    Source-Zone INSIDE Destination-Zone INSIDE
    service-policy inside
```

```
R3# show zone-pair security
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
2900	Gigabit Ethernet 0/0 (G0/0)	Gigabit Ethernet 0/1 (G0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
4221	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
4300	Gigabit Ethernet 0/0/0 (G0/0/0)	Gigabit Ethernet 0/0/1 (G0/0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)

Blank Line, No additional information

Note: To find out how the router is configured, look at the interfaces to identify the type of router and how many interfaces the router has. There is no way to effectively list all the combinations of configurations for each router class. This table includes identifiers for the possible combinations of Ethernet and Serial interfaces in the device. The table does not include any other type of interface, even though a specific router may contain one. An example of this might be an ISDN BRI interface. The string in parenthesis is the legal abbreviation that can be used in Cisco IOS commands to represent the interface.

end of document