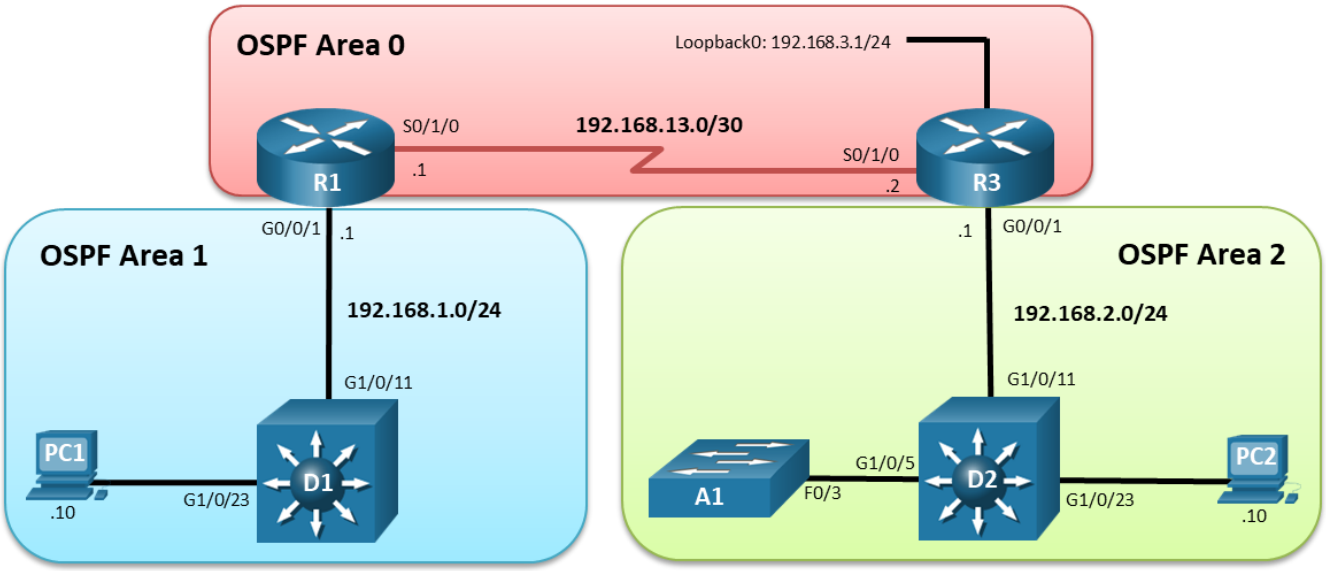
Lab - Implement IPv4 ACLs (Instructor Version)

Instructor Note: Red font color or gray highlights indicate text that appears in the instructor copy only.

Answers: 26.1.2 Lab - Implement IPv4 ACLs

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	S0/1/0	192.168.13.1	255.255.255.252	N/A
	G0/0/1	192.168.1.1	255.255.255.0	
R3	S0/1/0	192.168.13.2	255.255.255.252	N/A
	G0/0/1	192.168.2.1	255.255.255.0	
	Loopback0	192.168.3.1	255.255.255.0	
D1	VLAN 1	192.168.1.2	255.255.255.0	N/A
D2	VLAN 1	192.168.2.2	255.255.255.0	N/A
A1	VLAN 1	192.168.2.3	255.255.255.0	N/A
PC1	NIC	192.168.1.10	255.255.255.0	192.168.1.1
PC2	NIC	192.168.2.10	255.255.255.0	192.168.2.1

Objectives

Part 1: Build the Network and Configure Basic Device Settings

Part 2: Verify Initial Connectivity

Part 3: Implement Standard ACLs on R3

Part 4: Implement a Named Extended ACL from Area 1 to Area 2

Part 5: Implement a Named Extended ACL from Area 2 to Area 1

Part 6: Implement a Port ACL on D2

Part 7: Implement a VLAN ACL on D2

Background / Scenario

Access control lists (ACLs) are sequential lists of individual access control entries (ACEs) that permit or deny packets based on predefined conditional matching statements. Finding a match in an ACL starts at the top with the lowest sequence number and proceeds down the list (higher sequence numbers) until a matching ACE is found. When a match is found, the preset action (permit or deny) is applied and processing stops. At the end of every ACL is an implicit deny “any” ACE, which denies all packets that did not match prior ACEs.

ACLs can be used for packet classification with quality of service (QoS), Network Address Translations (NAT), and numerous other services.

In this lab, you will configure three different types of ACLs. Router ACL (RACL) is the most common ACL is the IP-based ACL that are applied to routed interface. The ACL that applies to traffic entering and leaving a VLAN is a VLAN ACL (VACL). The VACLs can filter traffic based on MAC addresses, IP addresses, and port numbers. A VACL that is applied to an individual port inside a VLAN is a port-based ACL (PACL).

The focus of this lab is using IPv4 ACLs for packet filtering.

Note: This lab is an exercise in configuring various types of access control lists and does not necessarily reflect network troubleshooting best practices.

Note: The routers used with CCNP 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 3650s with Cisco IOS XE Release 16.9.4 (universalk9 image) and Cisco Catalyst 2960s 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. If you are unsure, contact your instructor.

Instructor Note: Refer to the Instructor Lab Manual for the procedures to initialize and reload devices.

Required Resources

- 2 Routers (Cisco 4221 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 2 Switches (Cisco 3650 with Cisco IOS XE Release 16.9.4 universal image or comparable)
- 1 Switch (Cisco 2960 with Cisco IOS Release 15.2(2) lanbasek9 image or comparable)
- 3 PC (Choice of operating system with a terminal emulation program and a packet capture utility installed)
- Console cables to configure the Cisco IOS devices via the console ports
- Ethernet cables as shown in the topology

Instructions

Part 1: Build the Network and Configure Basic Device Settings

In Part 1, you will set up the network topology and configure basic settings and interface addressing on devices.

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: Configure basic settings for each device.

a. Console into each device, enter global configuration mode, and apply the basic settings. The startup configurations for each device are provided below.

Router R1

```
hostname R1
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # R1, Lab Access Control Lists #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface g0/0/1
 ip address 192.168.1.1 255.255.255.0
 no shutdown
 exit
interface Serial 0/1/0
 ip address 192.168.13.1 255.255.255.252
 no shutdown
 exit
router ospf 1
  router-id 0.0.0.1
  network 192.168.1.0 0.0.0.255 area 1
  network 192.168.13.0 0.0.0.3 area 0
  exit
line vty 0 4
 login local
 transport input telnet
end
```

Router R3

```
hostname R3
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # R3, Lab Access Control Lists #
line con 0
```

```
 exec-timeout 0 0
 logging synchronous
 exit
interface Loopback0
 ip address 192.168.3.1 255.255.255.0
 exit
interface g0/0/1
 ip address 192.168.2.1 255.255.255.0
 no shutdown
 exit
interface Serial 0/1/0
 ip address 192.168.13.2 255.255.255.252
 no shutdown
 exit
router ospf 1
  router-id 0.0.0.3
  network 192.168.2.0 0.0.0.255 area 2
  network 192.168.3.0 0.0.0.255 area 0
  network 192.168.13.0 0.0.0.3 area 0
  exit
line vty 0 4
 login local
 transport input telnet
end
```

Switch D1

```
hostname D1
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # D1, Lab Access Control Lists #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface range g1/0/11, g1/0/23
 switchport mode access
 no shutdown
 exit
interface vlan 1
 ip address 192.168.1.2 255.255.255.0
 no shut
 exit
 ip default-gateway 192.168.1.1
```

```
 line vty 0 15
 login local
 transport input telnet
 end
```

Switch D2

```
hostname D2
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
banner motd # D2, Lab Access Control Lists #
line con 0
 exec-timeout 0 0
 logging synchronous
 exit
interface range g1/0/1-24, g1/1/1-4, g0/0
 shutdown
 exit
interface range g1/0/5, g1/0/11, g1/0/23
switchport mode access
 no shutdown
 exit
interface vlan 1
 ip address 192.168.2.2 255.255.255.0
 no shut
 exit
 ip default-gateway 192.168.2.1
 line vty 0 15
 login local
 transport input telnet
 end
```

Switch A1

```
hostname A1
no ip domain lookup
username admin privilege 15 algorithm-type scrypt secret cisco123
ip http authentication local
ip domain name CCNP.ACL.LAB
banner motd # A1, Lab Access Control Lists #
spanning-tree mode rapid-pvst
 line con 0
 exec-timeout 0 0
 logging synchronous
 exit
line vty 0 15
 login local
 transport input telnet ssh
```

```
interface range f0/1-24, g0/1-2
 shutdown
 exit
interface range f0/3, f0/23
switchport mode access
 no shutdown
 exit
interface vlan 1
 ip address 192.168.2.3 255.255.255.0
 no shut
 exit
 ip default-gateway 192.168.2.1
 crypto key generate rsa modulus 1024
 end
```

b. Set the clock on each device to UTC time.

c. Save the running configuration to startup-config.

d. Configure and verify the IP address on PC1 and PC2.

e. Verify ICMP connectivity between all devices and PCs.

Part 2: Verify Initial Connectivity

It is always advisable to test network connectivity and services before applying ACL filtering. This ensures that the network is fully functional, and that the loss of connectivity or functionality is due to the applied ACLs and not a pre-existing network issue.

When testing TCP connectivity some services will prompt for a username/password. The pre-configured username is **admin** and password is **cisco123**.

Step 1: Test Telnet connectivity.

From both PCs test Telnet connectivity to all devices. You should be successful. Troubleshoot as needed.

Step 2: Test SSH connectivity.

From both PCs test SSH to A1. You should be successful. Troubleshoot as needed.

Step 3: Test HTTP and HTTPS connectivity.

a. From PC1, open a web browser and HTTP to A1. You will be prompted for a username/password. You should be successful. Troubleshoot as needed.

b. From PC1, open a web browser and access A1 via HTTPS. A warning will appear in the web browser when connecting via HTTPS because A1 generated a self-signed certificate. Continue to access A1 via HTTPS. You should be successful. Troubleshoot as needed.

Note: Ignoring a web browser warning on the internet could expose your computer to risk. But it is safe to do so in a lab environment where the equipment has no access to production networks or the internet.

Part 3: Implement Standard ACLs on R3

A standard ACL is able to match traffic based on the source IP address. When filtering data traffic using a standard ACL it is considered best practice to apply the ACL as close to the destination as possible. ACLs can be configured using a name or number. The standard ACL number range is 1-99 and 1300-1999.

Step 1: Configure a numbered standard ACL on R3 and block data traffic from the 192.168.1.0 /24 network.

a. Create a numbered standard ACL using the number 99 to deny the source network of 192.168.1.0/24.

```
R3(config)# access-list 99 deny 192.168.1.0 0.0.0.255
```

b. Then add a second ACE to permit all other traffic.

Note: 0.0.0.0 255.255.255.255 will be converted to the keyword "**any"**.

```
R3(config)# access-list 99 permit 0.0.0.0 255.255.255.255
```

Step 2: Apply the numbered standard ACL to the correct interface and in the correct direction.

a. Apply the ACL to the G0/0/1 interface.

```
R3(config)# interface g0/0/1
R3(config-if)# ip access-group 99 ?
  in   inbound packets
  out  outbound packets
```

b. Specify outbound on G0/0/1 because the data traffic is originating from R1 and exiting G/0/1 to access the 192.168.2.0/24 network.

```
R3(config-if)# ip access-group 99 out
R3(config-if)# end
```

Step 3: Verify that the numbered standard ACL is working properly.

a. From PC1, ping D2, A1, and PC2. The pings should fail. Notice the messages on PC1 is from R3.

b. Issue the **show access-lists** command on R3. The output shows packets matching the sequence number 10.

```
R3# show access-lists
Standard IP access list 99
    10 deny   192.168.1.0, wildcard bits 0.0.0.255 (12 matches)
    20 permit any
```

c. Next, from PC1, ping the loopback on R3 at 192.168.3.1. The ping should be successful. This verifies connectivity to resources on R3. However, ACL 99 denies access to the 192.168.2.0/24 network.

d. Remove the access list from R3 and remove the **ip access-group** command from interface G0/0/1 to provide connectivity to 192.168.2.0/24 network.

```
R3(config)# no access-list 99
R3(config)# interface g0/0/1
R3(config-if)# no ip access-group 99 out
```

e. Verify that PC1 can ping devices on the 192.168.2.0/24 network.

Step 4: Configure a named standard ACL to limit management traffic to R3.

Create a named standard ACL using the name MGMT-TRAFFIC. The ACL should only permit Telnet access to the management plane on R3 from PC1. Explicitly deny access from any other IPv4 destination and log the attempts.

```
R3(config)# ip access-list standard MGMT-TRAFFIC
R3(config-std-nacl)# permit host 192.168.1.10
R3(config-std-nacl)# deny any log
```

```
R3(config-std-nacl)# exit
```

Step 5: Apply the named standard ACL to the correct interface and in the correct direction.

Use the **access-class** command to apply the MGMT-TRAFFIC ACL to all inbound vty lines on R3. Outbound Telnet connections from R3 will still be allowed.

```
R3(config)# line vty 0 4
R3(config-line)# access-class MGMT-TRAFFIC in
R3(config-line)# end
```

Step 6: Verify that the named standard ACL is working properly.

a. Test the ACL by initiating a Telnet session from PC1 to the Loopback address (192.168.3.1) on R3. The results should be successful. Repeat the test from PC2. Telnet connectivity from PC2 should be denied.

b. On R3, you should see the console messages shown below. The first message shows a login success and the second message shows a denial.

```
R3#
*Feb 28 20:36:29.495: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: admin] [Source: 192.168.1.10] [localport: 23] at 20:36:29 UTC Fri Feb 28 2020
*Feb 28 20:37:10.647: %SEC-6-IPACCESSLOGS: list MGMT-TRAFFIC denied 192.168.2.10 1 packet
```

c. Issue the **show access-lists** on R3 to view the packet "matches" from each ACE.

```
R3# show access-lists
Standard IP access list MGMT-TRAFFIC
    10 permit 192.168.1.10 (2 matches)
    20 deny   any log (3 matches)
```

Note: During testing, the first packet in a flow will trigger a syslog message. Enabling logging with the **log** option in the **deny any** statement provides insight into the amount of denied traffic. Unfortunately, ACL logging can be CPU-intensive and can negatively affect other functions of the network device. There are two primary factors that contribute to the CPU load increase from ACL logging: process switching of packets that match log-enabled access control entries (ACEs), and the generation and transmission of log messages. Care should be taken when using the **log** option in a production network.

Part 4: Implement a Named Extended ACL from Area 1 to Area 2

Extended ACLs can filter traffic based on more than just source address. Extended ACLs can filter on protocol, source, and destination IP addresses, and source and destination port numbers. Extended ACLs may also be used to filter IP packets with header options.

When filtering data traffic using an extended ACL, it is considered best practice to apply the ACL as close to the source as possible. You can configure both numbered and named extended ACLs. Part 4 of this lab uses a named extended ACL. Review the ACL policy that will be implemented using a named extended ACL.

On R1, use the following requirements to create a named extended ACL that will filter traffic originating from OSPF area 1 destined to OSPF area 2. As a result, all traffic from area 2 destined to area 1 will also be filtered:

- Name the extended ACL AREA1_TO_AREA2.
- Deny any IP traffic from PC1 with a time-to-live (TTL) value less than 25.
- Permit PC1 to send ICMP echo packets to 192.168.2.0/24.
- Permit PC1 Telnet access to D2.
- Permit 192.168.1.0/24 network to SSH to A1.

- Permit PC1 to send HTTP traffic to A1.
- Permit PC1 to send HTTPS traffic to A1.
- Explicitly deny all other traffic.

Step 1: Configure the named extended ACL on R1.

a. Enter the following to configure the ACL.

```
R1(config)# ip access-list extended AREA1_TO_AREA2
R1(config-ext-nacl)# deny ip host 192.168.1.10 any ttl lt 25
R1(config-ext-nacl)# permit icmp host 192.168.1.10 192.168.2.0 0.0.0.255 echo
R1(config-ext-nacl)# permit tcp host 192.168.1.10 host 192.168.2.2 eq 23
R1(config-ext-nacl)# permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.3 eq 22
R1(config-ext-nacl)# permit tcp host 192.168.1.10 host 192.168.2.3 eq 80
R1(config-ext-nacl)# permit tcp host 192.168.1.10 host 192.168.2.3 eq 443
R1(config-ext-nacl)# deny ip any any
R1(config-ext-nacl)# end
```

b. Issue the **show access-list** command to verify the creation of the extended named ACL. Notice the sequence numbers and the top (lower sequence number) to bottom (higher sequence number) processing of each ACE. Also notice some of the port numbers have been changed to the IOS keywords. For example, port 80 is now **www** in the IOS.

```
R1# show access-lists
Extended IP access list AREA1_TO_AREA2
    10 deny ip host 192.168.1.10 any ttl lt 25
    20 permit icmp host 192.168.1.10 192.168.2.0 0.0.0.255 echo
    30 permit tcp host 192.168.1.10 host 192.168.2.2 eq telnet
    40 permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.3 eq 22
    50 permit tcp host 192.168.1.10 host 192.168.2.3 eq www
    60 permit tcp host 192.168.1.10 host 192.168.2.3 eq 443
    70 deny ip any any
```

Note: Sequence 70 essentially filters all other traffic that does not originate from area 1.

Step 2: Apply the named extended ACL to the correct interface and in the correct direction.

a. Apply the named ACL to the G0/0/1 interface on R1. Because traffic originates from area 1, apply the ACL inbound to R1

```
R1(config)# interface g0/0/1
R1(config-if)# ip access-group AREA1_TO_AREA2 in
R1(config-if)# end
```

Step 3: Verify that the AREA1_TO_AREA2 named extended ACL is working properly.

a. Test the first two lines of the ACL. From PC1, you should be able to successfully ping PC2.

b. From PC1, ping PC2 again, but this time set the TTL value to 20 inside the IP header on PC1. Use the following command on PC1 to set the TTL to 20 for the ICMP packet.

```
C:\> ping 192.168.2.10 –i 20
```

The ICMP packets with a TTL value of 20 should be dropped by R1, which is the area border router (ABR) for area 1. R1 sends error messages to PC1.

c. Continue to test each individual ACE within the ACL.

- From PC1, you should be able to successfully access D2 via Telnet. However, accessing any other device in area 2 via Telnet should be denied.
- From any device in area 1, you should be able to SSH to A1. However, accessing any other device on area 2 via SSH should be denied.
- From PC1, you should be able to access the web interface on A1 using either HTTP or HTTPS. However, all other attempts to access other devices using HTTP and HTTPS should time out.

d. After testing each ACE, issue the **show access-list** command. Notice that each ACE has matches. Your match counts will be different.

```
R1# show access-lists
Extended IP access list AREA1_TO_AREA2
    10 deny ip host 192.168.1.10 any ttl lt 25 (62 matches)
    20 permit icmp host 192.168.1.10 192.168.2.0 0.0.0.255 echo (12 matches)
    30 permit tcp host 192.168.1.10 host 192.168.2.2 eq telnet (31 matches)
    40 permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.3 eq 22 (21 matches)
    50 permit tcp host 192.168.1.10 host 192.168.2.3 eq www (15 matches)
    60 permit tcp host 192.168.1.10 host 192.168.2.3 eq 443 (92 matches)
    70 deny ip any any (49 matches)
```

Part 5: Implement a Named Extended ACL from Area 2 to Area 1

In this Part, you will configure and verify a named extended ACL to permit specific returning traffic from area 2. To do this, you will now configure the addressing and port numbers from area 2 as the source addresses and source ports. Additionally, the **established** keyword will be set on returning TCP connections to increase security.

Step 1: Configure the named extended ACL on R1.

a. Use this filtered output from the **show run** command below to configure the named extended ACL AREA2_TO_AREA1 on R1.

```
R1# show run | s AREA2_TO_AREA1
ip access-list extended AREA2_TO_AREA1
 permit icmp 192.168.2.0 0.0.0.255 host 192.168.1.10 echo-reply
 permit tcp host 192.168.2.2 eq telnet host 192.168.1.10 established
 permit tcp host 192.168.2.3 eq 22 192.168.1.0 0.0.0.255 established
 permit tcp host 192.168.2.3 eq www host 192.168.1.10 established
 permit tcp host 192.168.2.3 eq 443 host 192.168.1.10 established
 deny   ip any any log
```

b. Verify your configuration by entering the same **show run** command. Your output should be same as shown above.

c. Next, apply the ACL to the G/0/1 interface in the outgoing direction. Because traffic originated from area 1 and is returning from area 2, configure the ACL going outbound on the G/0/1 interface towards area 1.

```
R1(config)# interface g0/0/1
R1(config-if)# ip access-group AREA2_TO_AREA1 out
R1(config-if)# end
```

Step 2: Verify that the AREA2_TO_AREA1 named extended ACL is working properly.

a. Repeat the tests from Part 4, Step 4. The return traffic permitted in the ACL AREA2_TO_AREA1 should be successful.

b. Issue the **show ip access-list AREA2_TO_AREA1** command on R1 to see the matches for the return traffic from area 2.

```
R1# show access-lists AREA2_TO_AREA1
Extended IP access list AREA2_TO_AREA1
    10 permit icmp 192.168.2.0 0.0.0.255 host 192.168.1.10 echo-reply (4 matches)
    20 permit tcp host 192.168.2.2 eq telnet host 192.168.1.10 established (36 matches)
    30 permit tcp host 192.168.2.3 eq 22 192.168.1.0 0.0.0.255 established (23 matches)
    40 permit tcp host 192.168.2.3 eq www host 192.168.1.10 established (13 matches)
    50 permit tcp host 192.168.2.3 eq 443 host 192.168.1.10 established (26 matches)
    60 deny ip any any log
```

Note: The **established** option allows only TCP responses to traffic that originates from area 1 (192.168.1.0/24) to return. A match occurs if the returning TCP segment has either the ACK or reset (RST) bit set. Either of these bits indicates that the packet belongs to an established connection. Therefore, when filtering return traffic, the source port number must be checked.

c. To verify that both ACLs are applied to the G0/0/1 interface on R1, issue the following filtered **show ip interface** command.

```
R1# show ip interface g0/0/1 | s AREA
  Outgoing access list is AREA2_TO_AREA1
  Inbound  access list is AREA1_TO_AREA2
```

Part 6: Implement a Port ACL on D2

Port ACLs (PACLs) are similar to the router ACLs (RACLs) configured previously in this lab. However, PACLs are supported on Layer 2 switchports. PACLs can be implemented using standard or extended ACLs but can only be applied inbound. The processing of a PACL is done before a VLAN ACL (VACL) and RACL.

Step 1: Configure a PACL on D2 using the following requirements:

On D2, create an extended numbered PACL on port 23 which has the following requirements:

- Deny all ICMP messages sent to 192.168.2.3.
- Deny Telnet access to 192.168.2.2.
- Permit all other traffic.

The configuration is as follows:

```
D2(config)# access-list 123 deny icmp any host 192.168.2.3
D2(config)# access-list 123 deny tcp any host 192.168.2.2 eq 23
D2(config)# access-list 123 permit ip any any
```

Step 2: Apply the PACL to an interface on D2.

On D2, apply the ACL inbound on G1/0/23.

```
D2(config)# interface g1/0/23
D2(config-if)# ip access-group 123 in
D2(config-if)# end
```

Step 3: Verify that the PACL is working properly.

From PC2 test each ACE within the PACL.

a. A ping from PC2 to A1 should not be successful. However, a ping from PC2 to D2 should be successful.

b. Access via Telnet from PC2 to D2 should fail. However, access via Telnet and SSH from PC2 to A1 should be successful.

c. On D2 issue the **show run | s access-list** command. Notice the numbered ACL 123 configured earlier is now converted into the format for an extended named ACL.

```
D2# show run | s access-list
ip access-list extended 123
 deny   icmp any host 192.168.2.3 echo
 deny   tcp any host 192.168.2.2 eq telnet
 permit ip any any
```

Part 7: Implement a VLAN ACL on D2

VLAN ACLs (VACLs) control access to the VLAN of all packets, both bridged and routed. Packets can enter the VLAN via a Layer 2 switchport or through a Layer 3 interface after being routed. Similar to PACLs, VACLs can filter traffic within the same VLAN.

Step 1: Configure a VACL on D2 using the following requirements:

Create an extended VACL to support the following requirements:

- Deny pings from 192.168.2.2 to 192.168.2.3.
- Deny Telnet from 192.168.2.2 to 192.168.2.3.
- Permit all other traffic.

a. On D2 create an extended ACL named **D2ACL**. Notice that in this first configuration step, the traffic to be blocked is defined in the ACL with a **permit** statement. That is because in Step1c, you will configure a **vlan access-map** to match this permitted traffic and then drop it. All denied traffic will not match the first **vlan access -map** statement, but will match the second **vlan access -map** statement in Step1d. This second statement will forward all other traffic.

```
D2(config)# ip access-list extended D2ACL
D2(config-ext-nacl)# permit icmp host 192.168.2.2 host 192.168.2.3
D2(config-ext-nacl)# permit tcp host 192.168.2.2 host 192.168.2.3 eq 22
D2(config-ext-nacl)# deny ip any any
D2(config-ext-nacl)# exit
```

b. On D2, configure the VLAN access map with the name VACL and sequence 10. Within the VLAN access map match the access list named D2ACL. Then set the action to drop any packets being permitted by the D2ACL.

```
D2(config)# vlan access-map VACL 10
D2(config-access-map)# match ip address D2ACL
D2(config-access-map)# action drop
D2(config-access-map)# exit
```

c. Like all ACLs, there is an implied “deny any” with a VLAN ACL. To prevent all traffic from being dropped, create a sequence 20 for the same VLAN access map. Next, set the action to forward.

```
D2(config)# vlan access-map VACL 20
D2(config-access-map)# action forward
D2(config-access-map)# exit
```

Step 2: Apply the VACL to a VLAN filter.

To apply a VACL, use the **vlan filter** *vlan-access-map-name* **vlan-list** *vlan-list* command. The *vlan-list* can be a single VLAN, a contiguous range of VLANs (7–10), or a comma separated list of multiple VLANs (4,9–11,17). Notice a direction is not required because the VLAN filter applies to both intra-VLAN in inter-VLAN traffic.

```
D2(config)# vlan filter VACL vlan-list 1
```

Step 3: Verify that the VACL is working properly.

From D2, test each ACE within the VACL.

a. From D2, ping A1. The ping should fail.

```
D2# ping 192.168.2.3
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.10, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)
```

b. From D2, use Telnet to access A1. The connection should time out.

```
D2# telnet 192.168.2.3
Trying 192.168.2.3 ...
% Connection timed out; remote host not responding
```

c. From D2, ping PC10. The ping should be successful.

```
D2# ping 192.168.2.10
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.2.10, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 2/2/3 ms
```

d. From D2, SSH to A1. The connection should be successful. Exit the SSH session.

```
D2# ssh -l admin 192.168.2.3
Password:
 A1, Lab Access Control Lists A1#
A1# exit

[Connection to 192.168.2.3 closed by foreign host]
D2#
```

e. Use the **show vlan access-map** and **show access-list commands** to view the VACL configuration.

```
D2# show vlan access-map
Vlan access-map "VACL"  10
  Match clauses:
    ip   address: D2ACL
  Action:
    drop
Vlan access-map "VACL"  20
  Match clauses:
  Action:
    forward
```

```
D2# show access-lists D2ACL
Extended IP access list D2ACL
    10 permit icmp host 192.168.2.2 host 192.168.2.3
    20 permit tcp host 192.168.2.2 host 192.168.2.3 eq telnet
    30 deny ip any any
```

Router Interface Summary Table

Router Model	Ethernet Interface #1	Ethernet Interface #2	Serial Interface #1	Serial Interface #2
1800	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/0/0 (S0/0/0)	Serial 0/0/1 (S0/0/1)
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)
2801	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	Serial 0/1/0 (S0/1/0)	Serial 0/1/1 (S0/1/1)
2811	Fast Ethernet 0/0 (F0/0)	Fast Ethernet 0/1 (F0/1)	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)

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.

Device Configs – Final

Instructor Note: ACL 99 created in Part 3 was removed and is not shown in final config for R3.

Router R1

```
R1# show running-config
Building configuration...


Current configuration : 2355 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
```

```
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$hvvdMpWq40xmg0$dcldDqEfm1cWDhtbJm60nqJ7s1n3akPNBQl5Jo0vI/k
!
redundancy
 mode none
!
interface GigabitEthernet0/0/0
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.1.1 255.255.255.0
 ip access-group AREA1_TO_AREA2 in
 ip access-group AREA2_TO_AREA1 out
 negotiation auto
!
interface Serial0/1/0
 ip address 192.168.13.1 255.255.255.252
!
interface Serial0/1/1
 no ip address
!
router ospf 1
 router-id 0.0.0.1
 network 192.168.1.0 0.0.0.255 area 1
 network 192.168.13.0 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
ip access-list extended AREA1_TO_AREA2
 deny   ip host 192.168.1.10 any ttl lt 25
```

```
 permit icmp host 192.168.1.10 192.168.2.0 0.0.0.255 echo
 permit tcp host 192.168.1.10 host 192.168.2.2 eq telnet
 permit tcp 192.168.1.0 0.0.0.255 host 192.168.2.3 eq 22
 permit tcp host 192.168.1.10 host 192.168.2.3 eq www
 permit tcp host 192.168.1.10 host 192.168.2.3 eq 443
 deny   ip any any
ip access-list extended AREA2_TO_AREA1
 permit icmp 192.168.2.0 0.0.0.255 host 192.168.1.10 echo-reply
 permit tcp host 192.168.2.2 eq telnet host 192.168.1.10 established
 permit tcp host 192.168.2.3 eq 22 192.168.1.0 0.0.0.255 established
 permit tcp host 192.168.2.3 eq www host 192.168.1.10 established
 permit tcp host 192.168.2.3 eq 443 host 192.168.1.10 established
 deny   ip any any log
!
control-plane
!
banner motd ^C R1, Lab Access Control Lists ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input telnet
!
end
```

Router R3

```
R3# show running-config
Building configuration...


Current configuration : 1649 bytes
!
version 16.9
service timestamps debug datetime msec
service timestamps log datetime msec
platform qfp utilization monitor load 80
no platform punt-keepalive disable-kernel-core
!
hostname R3
!
boot-start-marker
boot-end-marker
!
```

```
no aaa new-model
!
no ip domain lookup
!
login on-success log
!
subscriber templating
!
multilink bundle-name authenticated
!
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$tDPbCeTe5wv3..$xGjyZS1UXPsbn1RHas.36.naR/NPTT3F9qOB.IBxuGY
!
redundancy
 mode none
!
interface Loopback0
 ip address 192.168.3.1 255.255.255.0
!
interface GigabitEthernet0/0/0
 no ip address
 negotiation auto
!
interface GigabitEthernet0/0/1
 ip address 192.168.2.1 255.255.255.0
 negotiation auto
!
interface Serial0/1/0
 ip address 192.168.13.2 255.255.255.252
!
interface Serial0/1/1
 no ip address
!
router ospf 1
 router-id 0.0.0.3
 network 192.168.2.0 0.0.0.255 area 2
 network 192.168.3.0 0.0.0.255 area 0
 network 192.168.13.0 0.0.0.3 area 0
!
ip forward-protocol nd
no ip http server
ip http secure-server
!
ip access-list standard MGMT-TRAFFIC
 permit 192.168.1.10
 deny   any log
!
control-plane
```

```
!
banner motd ^C R3, Lab Access Control Lists ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 transport input none
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 access-class MGMT-TRAFFIC in
 login local
 transport input telnet
!
end
```

Switch D1

```
D1# show running-config
Building configuration...

Current configuration : 4863 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D1
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
!
no aaa new-model
switch 1 provision ws-c3650-24ps
!
no ip domain lookup
!
login on-success log
!
```

```
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$L5ulQyyHG0iozE$zXAEobs86JoglKn2p/xilCqGQLzcAaE/b6HWbT6gaZ6
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception,
NFLSAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
```

```
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 shutdown
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 switchport mode access
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
```

```
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 switchport mode access
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 ip address 192.168.1.2 255.255.255.0
!
ip default-gateway 192.168.1.1
ip forward-protocol nd
```

```
ip http server
ip http secure-server
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D1, Lab Access Control Lists ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input telnet
line vty 5 15
 login local
 transport input telnet
!
end
```

Switch D2

```
D2# show running-config
Building configuration...

Current configuration : 5312 bytes
!
version 16.9
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
! Call-home is enabled by Smart-Licensing.
service call-home
no platform punt-keepalive disable-kernel-core
!
hostname D2
!
vrf definition Mgmt-vrf
 !
 address-family ipv4
 exit-address-family
 !
 address-family ipv6
 exit-address-family
!
no aaa new-model
switch 1 provision ws-c3650-24ps
```

```
!
no ip domain lookup
!
login on-success log
!
license boot level ipservicesk9
!
diagnostic bootup level minimal
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
username admin privilege 15 secret 9
$9$HEkaAR1.pGOn6.$x/d/YVdPMABEPyjrajuUe2IIMBetxO8vBxSJPWGWlRM
!
redundancy
 mode sso
!
transceiver type all
 monitoring
!
vlan access-map VACL 10
 match ip address D2ACL
 action drop
vlan access-map VACL 20
 action forward
!
vlan filter VACL vlan-list 1
!
class-map match-any system-cpp-police-topology-control
  description Topology control
class-map match-any system-cpp-police-sw-forward
  description Sw forwarding, L2 LVX data, LOGGING
class-map match-any system-cpp-default
  description Inter FED, EWLC control, EWLC data
class-map match-any system-cpp-police-sys-data
  description Learning cache ovfl, High Rate App, Exception, EGR Exception,
NFLSAMPLED DATA, RPF Failed
class-map match-any system-cpp-police-punt-webauth
  description Punt Webauth
class-map match-any system-cpp-police-l2lvx-control
  description L2 LVX control packets
class-map match-any system-cpp-police-forus
  description Forus Address resolution and Forus traffic
class-map match-any system-cpp-police-multicast-end-station
  description MCAST END STATION
class-map match-any system-cpp-police-multicast
  description Transit Traffic and MCAST Data
class-map match-any system-cpp-police-l2-control
  description L2 control
```

```
class-map match-any system-cpp-police-dot1x-auth
  description DOT1X Auth
class-map match-any system-cpp-police-data
  description ICMP redirect, ICMP_GEN and BROADCAST
class-map match-any system-cpp-police-stackwise-virt-control
  description Stackwise Virtual
class-map match-any non-client-nrt-class
class-map match-any system-cpp-police-routing-control
  description Routing control and Low Latency
class-map match-any system-cpp-police-protocol-snooping
  description Protocol snooping
class-map match-any system-cpp-police-dhcp-snooping
  description DHCP snooping
class-map match-any system-cpp-police-system-critical
  description System Critical and Gold Pkt
!
policy-map system-cpp-policy
!
interface GigabitEthernet0/0
 vrf forwarding Mgmt-vrf
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet1/0/1
 shutdown
!
interface GigabitEthernet1/0/2
 shutdown
!
interface GigabitEthernet1/0/3
 shutdown
!
interface GigabitEthernet1/0/4
 shutdown
!
interface GigabitEthernet1/0/5
 switchport mode access
!
interface GigabitEthernet1/0/6
 shutdown
!
interface GigabitEthernet1/0/7
 shutdown
!
interface GigabitEthernet1/0/8
 shutdown
!
interface GigabitEthernet1/0/9
 shutdown
```

```
!
interface GigabitEthernet1/0/10
 shutdown
!
interface GigabitEthernet1/0/11
 switchport mode access
!
interface GigabitEthernet1/0/12
 shutdown
!
interface GigabitEthernet1/0/13
 shutdown
!
interface GigabitEthernet1/0/14
 shutdown
!
interface GigabitEthernet1/0/15
 shutdown
!
interface GigabitEthernet1/0/16
 shutdown
!
interface GigabitEthernet1/0/17
 shutdown
!
interface GigabitEthernet1/0/18
 shutdown
!
interface GigabitEthernet1/0/19
 shutdown
!
interface GigabitEthernet1/0/20
 shutdown
!
interface GigabitEthernet1/0/21
 shutdown
!
interface GigabitEthernet1/0/22
 shutdown
!
interface GigabitEthernet1/0/23
 switchport mode access
 ip access-group 123 in
!
interface GigabitEthernet1/0/24
 shutdown
!
interface GigabitEthernet1/1/1
 shutdown
!
```

```
interface GigabitEthernet1/1/2
 shutdown
!
interface GigabitEthernet1/1/3
 shutdown
!
interface GigabitEthernet1/1/4
 shutdown
!
interface Vlan1
 ip address 192.168.2.2 255.255.255.0
!
ip default-gateway 192.168.2.1
ip forward-protocol nd
ip http server
ip http secure-server
!
ip access-list extended D2ACL
 permit icmp host 192.168.2.2 host 192.168.2.3
 permit tcp host 192.168.2.2 host 192.168.2.3 eq telnet
 deny   ip any any
!
ip access-list extended 123
 deny   icmp any host 192.168.2.3 echo
 deny   tcp any host 192.168.2.2 eq telnet
 permit ip any any
!
control-plane
 service-policy input system-cpp-policy
!
banner motd ^C D2, Lab Access Control Lists ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
 stopbits 1
line aux 0
 stopbits 1
line vty 0 4
 login local
 transport input telnet
line vty 5 15
 login local
 transport input telnet
!
end
```

Switch A1

```
A1# show running-config
```

```
Building configuration...

Current configuration : 1992 bytes
!
version 15.2
no service pad
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname A1
!
boot-start-marker
boot-end-marker
!
username admin privilege 15 secret 9
$9$b3yUg9rBvRaLFq$9MUWdShuqqLQy6U9DnDVT2EzGH1n4ZOUfv8p3WGtZTg
no aaa new-model
system mtu routing 1500
!
no ip domain-lookup
ip domain-name CCNP.ACL.LAB
!
spanning-tree mode rapid-pvst
spanning-tree extend system-id
!
vlan internal allocation policy ascending
!
interface FastEthernet0/1
 shutdown
!
interface FastEthernet0/2
 shutdown
!
interface FastEthernet0/3
 switchport mode access
!
interface FastEthernet0/4
 shutdown
!
interface FastEthernet0/5
 shutdown
!
interface FastEthernet0/6
 shutdown
!
interface FastEthernet0/7
 shutdown
!
interface FastEthernet0/8
```

```
 shutdown
!
interface FastEthernet0/9
 shutdown
!
interface FastEthernet0/10
 shutdown
!
interface FastEthernet0/11
 shutdown
!
interface FastEthernet0/12
 shutdown
!
interface FastEthernet0/13
 shutdown
!
interface FastEthernet0/14
 shutdown
!
interface FastEthernet0/15
 shutdown
!
interface FastEthernet0/16
 shutdown
!
interface FastEthernet0/17
 shutdown
!
interface FastEthernet0/18
 shutdown
!
interface FastEthernet0/19
 shutdown
!
interface FastEthernet0/20
 shutdown
!
interface FastEthernet0/21
 shutdown
!
interface FastEthernet0/22
 shutdown
!
interface FastEthernet0/23
 switchport mode access
!
interface FastEthernet0/24
 shutdown
!
```

```
interface GigabitEthernet0/1
 shutdown
!
interface GigabitEthernet0/2
 shutdown
!
interface Vlan1
 ip address 192.168.2.3 255.255.255.0
!
ip default-gateway 192.168.2.1
ip http server
ip http authentication local
ip http secure-server
!
banner motd ^C A1, Lab Access Control Lists ^C
!
line con 0
 exec-timeout 0 0
 logging synchronous
line vty 0 4
 login local
 transport input telnet ssh
line vty 5 15
 login local
 transport input telnet ssh
!
end
```