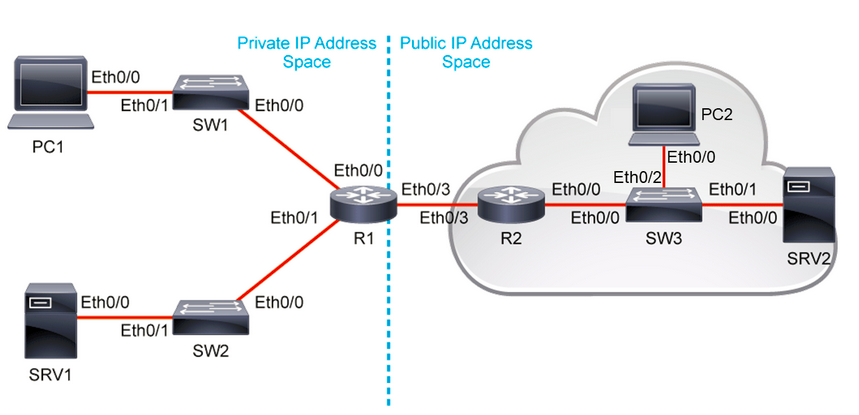
**Discovery 25: Configure Standard and Extended ACLs**

**Introduction**

Through this discovery, you will review the implementation of standard and extended ACLs using both numbered and named configuration methods. SRV2 is a public DNS server. The domain names of the devices are the same as their hostnames. SRV1 and SRV2 are also HTTP/HTTPS servers.

****

In this discovery you will configure ACLs on router R1 to accomplish the following objectives:

* Use a numbered standard ACL to block host PC1 from accessing the 10.10.2.0/24 network but allow all other traffic.
* Use a named standard ACL to only allow PC1 access to the VTY lines for remote management.
* Use a named extended ACL that will specifically:
  + Allow HTTP and HTTPS access to SRV1 from any public device
  + Allow Ping replies from any public device to the 10.10.1.0/24 network
  + Allow DNS replies from SRV2 to the 10.10.1.0/24 network
  + Allow HTTP and HTTPS replies from any public devices to the 10.10.1.0/24 network.

The virtual lab environment is prepared with the devices that are represented in the topology diagram and configured according to the information in the Job Aids section.

**Task 1: Configure and Verify ACLs**

**Activity**

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.

**Step 1:** Using Telnet, perform the following tests:

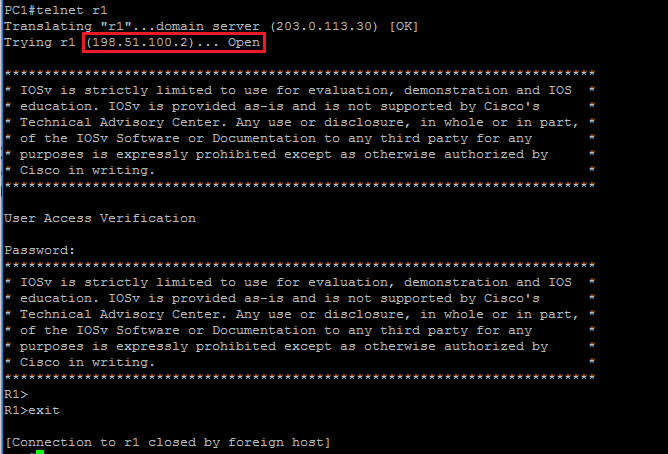
* PC1 can successfully Telnet to R1 using port 23
* PC1 can successfully Telnet to SRV1 using port 80 and 443
* PC1 can successfully Telnet to SRV2 using port 80 and 443
* PC2 can successfully Telnet to SRV1 using port 80 and 443

**Note**

When telnetting to port 80 or 443 you might need to press Enter to get the system to respond after the session is established. Use the password (root@123).

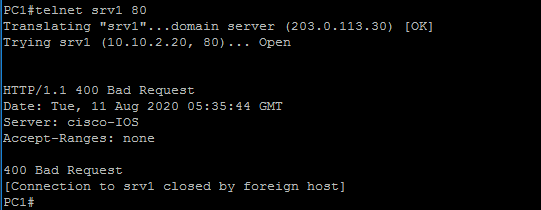
On PC1 and PC2, enter the following commands:

PC1# telnet r1

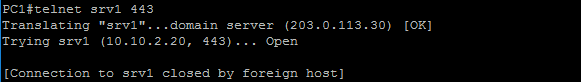


After the Telnet session opens, you can use any character key followed by the Enter key to close the connection.

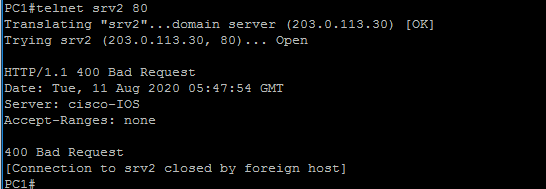
PC1#telnet srv1 80



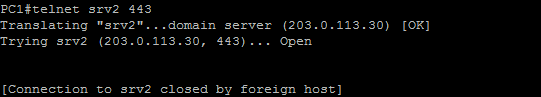
PC1#telnet srv1 443

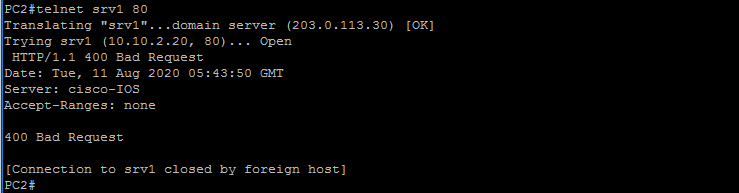


PC1# telnet srv2 80

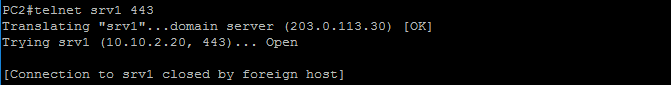


PC1# telnet srv2 443



PC2#telnet srv1 80

PC2#telnet srv1 443



**Step 2:** All tests are successful. Telnet is used to verify port 80 and port 443 functionality on SRV1 and SRV2. After the Telnet session opens, you can use any character key followed by the Enter key to close the connection.

**Step 3:** Standard ACLs can filter traffic based on source IP address only. A typical best practice is to configure a standard ACL as close to the destination as possible. In this step, you are configuring a numbered standard ACL on R1. The ACL is designed to block traffic from host 10.10.1.10 to the 10.10.2.0/24 network on R1. This ACL will be applied outbound on the R1 GigabitEthernet 0/1 interface.

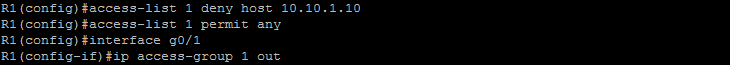
On R1, enter the following commands:

R1(config)# access-list 1 deny host 10.10.1.10

R1(config)# access-list 1 permit any

R1(config)# interface GigabitEthernet 0/1

R1(config-if)# ip access-group 1 out



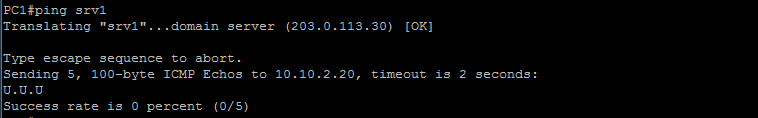
The purpose of this ACL is to block host PC1 from accessing the 10.10.2.0/24 network but allow all other traffic. Remember that every ACL has an implicit “deny any” that causes all traffic that has not matched a statement in the ACL to be blocked. For this reason, the permit any statement was added to the end of the ACL.

The ACL is applied in the out direction on interface GigabitEthernet 0/1 since that is the closest interface to the destination you are trying block.

**Step 4:** Test the ACL by pinging from PC1 to SRV1. Since the ACL is designed to block traffic with source addresses from the 10.10.1.10 host, PC1 should not be able to ping SRV1. Perform a similar test from PC2 to ensure that other devices still have access to SRV1. Verify the ACL using the show access-list command.

On PC1, enter the following command:

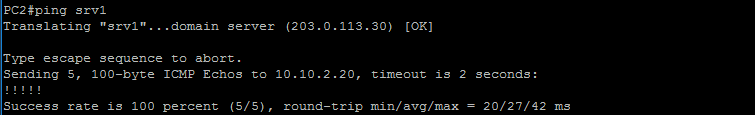
PC1# ping srv1



First notice that the DNS requests that to SRV2 are still functional. SRV2 has resolved the “srv1” domain name and returned the 10.10.2.20 address to PC1. PC1 then sends five echo-request messages to SRV1 but R1 drops the packets and returns ICMP destination unreachable messages to PC1 because of the deny statement in the ACL.

On PC2, enter the following command:

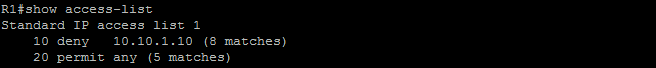
PC2# ping srv1



Connectivity to SRV1 from other sources still works because of the permit any statement in the ACL.

On R1, enter the following command:

R1# show access-list

****

Standard IP access list 1 10 deny 10.10.1.10 (8 matches) 20 permit any (5 matches)

You should see matches for both access control entries (ACEs) as packets from PC1 were denied and packets from PC2 were permitted through the ACL.

**Step 5:** As a network administrator, it is important to have remote access to your routers, switches, and firewalls. This access should not be available to other users of the network. Therefore, you will configure and apply a named standard ACL that allows PC1 access to the vty lines on R1, but explicitly denies all other source IP addresses. Use the log keyword to generate a syslog message at the console when a connection attempt is blocked.

On R1, enter the following commands:

R1(config)# ip access-list standard VTY-ACCESS

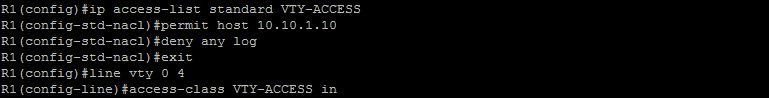
R1(config-std-nacl)# permit host 10.10.1.10

R1(config-std-nacl)# deny any log

R1(config-std-nacl)# exit

R1(config)# line vty 0 4

R1(config-line)# access-class VTY-ACCESS in



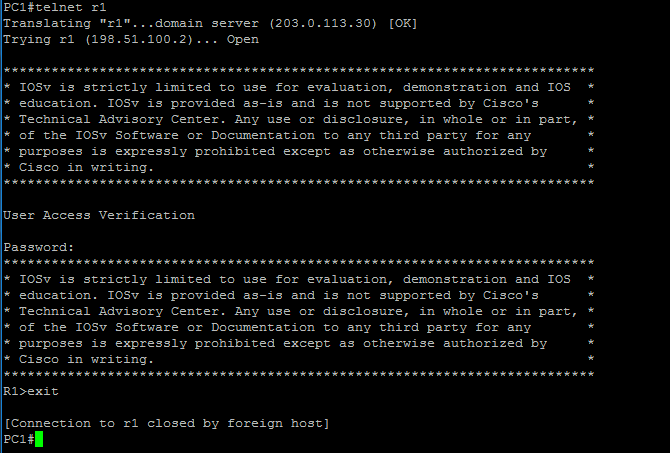
A named standard ACL called VTY-ACCESS is configured with one permit statement for PC1’s IP address and an explicit deny any statement allowing you to apply the log keyword. During testing, the first packet in a flow will trigger a syslog message. Logging-enabled ACLs provide insight into traffic as it traverses the network or is dropped by network devices. 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.

The ACL is then applied to all router vty lines in the in direction since the connection attempts will be viewed as inbound from the perspective of the router.

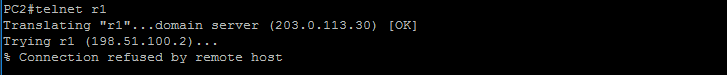
**Step 6:** Test the ACL by initiating a Telnet session from PC1 to R1. Repeat the test from PC2 To R1. On R1, verify the syslog messages and the ACL matches.

On PC1, P2, and R1, enter the following commands:

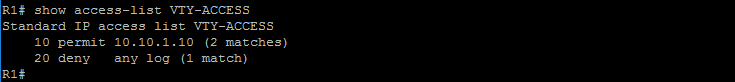
PC1# telnet r1



PC2# telnet r1



R1# show access-list VTY-ACCESS



The Telnet connection from PC1 to R1 is successful but not from PC2. Because of the log keyword, a syslog message is triggered by the explicit deny any statement at the end of the ACL. The message confirms the source of the failed connection attempt (PC2). The show access-list VTY-ACCESS command indicates matches for both ACEs since packets were permitted from PC1 and denied from PC2.

When greater granularity is required, you can use an extended ACL. 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.

**Step 7:** In this step, you will configure a complex named extended ACL on R1 that will act as a traffic filter for request or replies that originate in the public IP address space on the right of the topology, and that are destined for the private IP address space on the left of the topology.

The access list will be applied on R1 to interface GigabitEthernet 0/3 in the inbound direction with the following objectives in mind:

* Allowing any public devices HTTP and HTTPS access to SRV1
* Allowing Ping replies to the 10.10.1.0/24 network from any public devices
* Allowing DNS replies to the 10.10.1.0/24 network from SRV2
* Allowing HTTP and HTTPS replies to the 10.10.1.0/24 network from any public devices.
* Explicitly denying all other traffic.

On R1, enter the following commands:

R1(config)# ip access-list extended TRAFFIC-FILTER

R1(config-ext-nacl)# permit tcp any host 10.10.2.20 eq www

R1(config-ext-nacl)# permit tcp any host 10.10.2.20 eq 443

R1(config-ext-nacl)# permit icmp any 10.10.1.0 0.0.0.255 echo-reply

R1(config-ext-nacl)# permit udp host 203.0.113.30 eq domain 10.10.1.0 0.0.0.255

R1(config-ext-nacl)# permit tcp any eq www 10.10.1.0 0.0.0.255 established

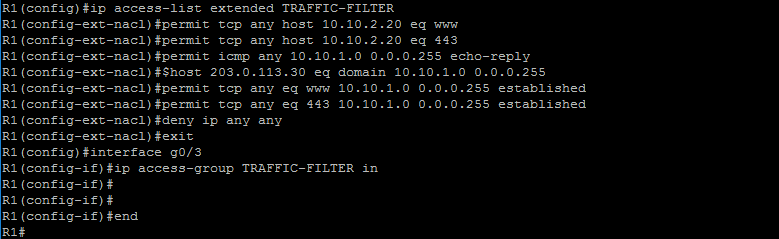
R1(config-ext-nacl)# permit tcp any eq 443 10.10.1.0 0.0.0.255 established

R1(config-ext-nacl)# deny ip any any

R1(config-ext-nacl)# exit

R1(config)# interface GigabitEthernet 0/3

R1(config-if)# ip access-group TRAFFIC-FILTER in



A named extended ACL called TRAFFIC-FILTER is configured with the following ACEs:

* Permit any public device HTTP access to SRV1.
* Permit any public device HTTPS access to SRV1.
* Permit ICMP Ping replies to the PC1 network from any public source.
* Permit DNS replies to the PC1 network from the DNS server SRV2.
* Permit HTTP replies to the PC1 network from any public source.
* Permit HTTPS replies to the PC1 network from any public source.
* Deny all other traffic.

Notice the use of the established parameter for two of the ACEs. The established parameter allows only responses to traffic that originated from the 10.10.1.0/24 network to return to that network from any HTTP or HTTPS public server. A match occurs if the returning TCP segment has the ACK or reset (RST) bits set, which indicates that the packet belongs to a pre-established connection. Without the established parameter in the ACL statement, clients could send traffic to a web server, but not receive traffic returning from that web server.

When filtering return traffic, the source port number must be checked. You can see this being accomplished in lines 4, 5, and 6.

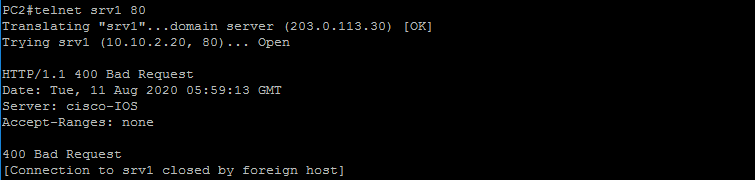
The ACL is applied inbound on R1’s public-facing interface (GigabitEthernet 0/3) since the objective is to filer traffic arriving from the public IP address space.

**Step 8:** Perform the following tests to verify ACL functionality:

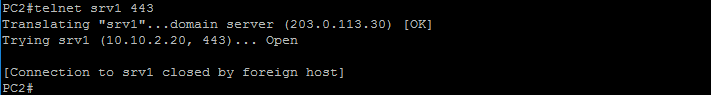
* PC2 can successfully Telnet to SRV1 using port 80
* PC2 can successfully Telnet to SRV1 using port 443
* PC1 can successfully Ping SRV2 using name resolution (this tests ACE #3 and #4)
* PC1 can successfully Telnet to SRV2 using port 80
* PC2 can successfully Telnet to SRV2 using port 443
* All other tests should fail.

On PC1 and PC2, enter the following commands:

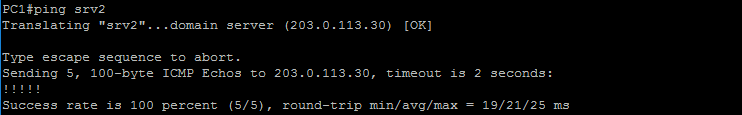
PC2# telnet srv1 80



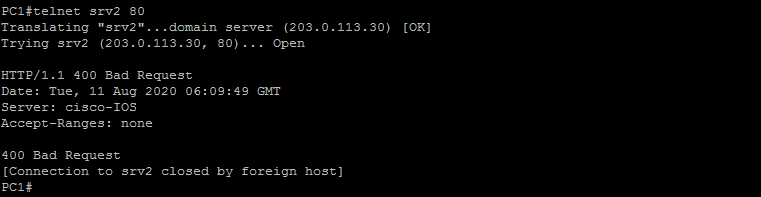
PC2#telnet srv1 443



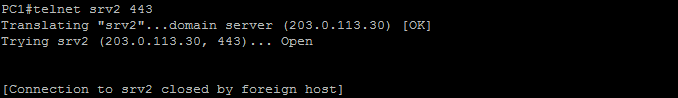
PC1#ping srv2



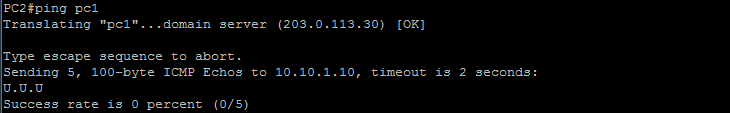
PC1#telnet srv2 80



PC1#telnet srv2 443



PC2#ping pc1



All tests are successful, except the Ping from PC2 to PC1. This type of traffic is not permitted through the ACL. R1 returns ICMP destination unreachable messages to PC2 when these packets are dropped.

**Step 9:** On R1, verify the TRAFFIC-FILTER ACL.

On R1, enter the following command:

R1# show access-lists TRAFFIC-FILTER

Extended IP access list TRAFFIC-FILTER

10 permit tcp any host 10.10.2.20 eq www (13 matches)

20 permit tcp any host 10.10.2.20 eq 443 (12 matches)

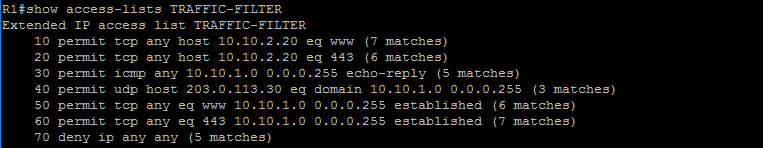
30 permit icmp any 10.10.1.0 0.0.0.255 echo-reply (5 matches)

40 permit udp host 203.0.113.30 eq domain 10.10.1.0 0.0.0.255 (3 matches)

50 permit tcp any eq www 10.10.1.0 0.0.0.255 established (6 matches)

60 permit tcp any eq 443 10.10.1.0 0.0.0.255 established (7 matches)

70 deny ip any any (5 matches)



Each ACE indicates several matches. This shows that packets were either permitted or denied depending on the action configured in each statement.

**Step 10:** Modify the TRAFFIC-FILTER ACL on R1 to allow PC2 SSH access to SRV1. Insert this new ACE at line 65 of the ACL.

On R1, enter the following commands:

R1(config)# ip access-list extended TRAFFIC-FILTER

R1(config-ext-nacl)# 65 permit tcp host 203.0.113.40 host 10.10.2.20 eq 22

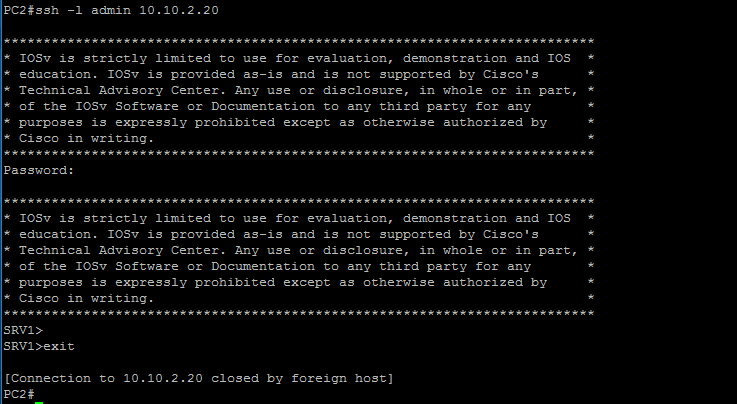


Sequence numbers can be used to delete or insert an ACL statement. The ip access-list extended name command is used to enter named-ACL configuration mode. If the ACL is numbered instead of named, the ACL number is used for the name parameter. ACEs can be inserted or removed using the no keyword. For example, in named-ACL configuration mode, the command no 40 would delete line 40 of the ACL.

Line 65 is configured to only allow PC2 to connect to SRV1 using SSH.

**Step 11:** On PC2, open an SSH connection with SRV1 to test the new ACE entry. Use the username (admin) with password (root@123).

On PC2, enter the following commands:

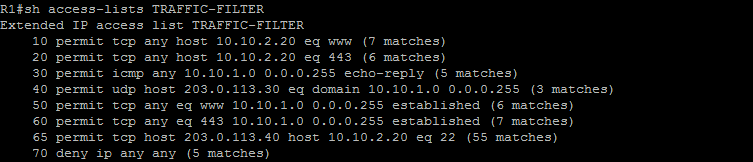
PC2# ssh -l admin 10.10.2.20 Password: (root@123)

The SSH connection is successful.

**Step 12:** On R1, verify the TRAFFIC-FILTER ACL for line 65 matches.

On R1, enter the following command:

R1# sh access-lists TRAFFIC-FILTER

The new ACL at line 65 shows that several packets have matched and were permitted through the ACL.