Packet Tracer - Configuring Extended ACLs - Scenario 1

Topology



1. Addressing Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Device | Interface | IP Address | Subnet Mask | Default Gateway |
| R1 | G0/0 | 172.22.34.65 | 255.255.255.224 | N/A |
| G0/1 | 172.22.34.97 | 255.255.255.240 | N/A |
| G0/2 | 172.22.34.1 | 255.255.255.192 | N/A |
| Server | NIC | 172.22.34.62 | 255.255.255.192 | 172.22.34.1 |
| PC1 | NIC | 172.22.34.66 | 255.255.255.224 | 172.22.34.65 |
| PC2 | NIC | 172.22.34.98 | 255.255.255.240 | 172.22.34.97 |

1. Objectives

Part 1: Configure, Apply and Verify an Extended Numbered ACL

Part 2: Configure, Apply and Verify an Extended Named ACL

1. Background / Scenario

Two employees need access to services provided by the server. **PC1** only needs FTP access while **PC2** only needs web access. Both computers are able to ping the server, but not each other.

1. Configure, Apply and Verify an Extended Numbered ACL
   1. Configure an ACL to permit FTP and ICMP.
      1. From global configuration mode on **R1**, enter the following command to determine the first valid number for an extended access list.

R1(config)# **access-list ?**

<1-99> IP standard access list

<100-199> IP extended access list

* + 1. Add **100** to the command, followed by a question mark.

R1(config)# **access-list 100 ?**

deny Specify packets to reject

permit Specify packets to forward

remark Access list entry comment

* + 1. To permit FTP traffic, enter **permit,** followed by a question mark.

R1(config)# **access-list 100 permit ?**

ahp Authentication Header Protocol

eigrp Cisco's EIGRP routing protocol

esp Encapsulation Security Payload

gre Cisco's GRE tunneling

icmp Internet Control Message Protocol

ip Any Internet Protocol

ospf OSPF routing protocol

tcp Transmission Control Protocol

udp User Datagram Protocol

* + 1. This ACL permits FTP and ICMP. ICMP is listed above, but FTP is not, because FTP uses TCP. So you enter TCP. Enter **tcp** to further refine the ACL help.

R1(config)# **access-list 100 permit tcp ?**

A.B.C.D Source address

any Any source host

host A single source host

* + 1. Notice that we could filter just for **PC1** by using the **host** keywordor we could allow **any** host. In this case, any device is allowed that has an address belonging to the 172.22.34.64/27 network. Enter the network address, followed by a question mark.

R1(config)# **access-list 100 permit tcp 172.22.34.64 ?**

A.B.C.D Source wildcard bits

* + 1. Calculate the wildcard mask determining the binary opposite of a subnet mask.

**11111111.11111111.11111111.111**00000 = 255.255.255.224

00000000.00000000.00000000.000**11111** = 0.0.0.31

* + 1. Enter the wildcard mask, followed by a question mark.

R1(config)# **access-list 100 permit tcp 172.22.34.64 0.0.0.31 ?**

A.B.C.D Destination address

any Any destination host

eq Match only packets on a given port number

gt Match only packets with a greater port number

host A single destination host

lt Match only packets with a lower port number

neq Match only packets not on a given port number

range Match only packets in the range of port numbers

* + 1. Configure the destination address. In this scenario, we are filtering traffic for a single destination, the server. Enter the **host** keywordfollowed by the server’s IP address.

R1(config)# **access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 ?**

dscp Match packets with given dscp value

eq Match only packets on a given port number

established established

gt Match only packets with a greater port number

lt Match only packets with a lower port number

neq Match only packets not on a given port number

precedence Match packets with given precedence value

range Match only packets in the range of port numbers

<cr>

* + 1. Notice that one of the options is **<cr>** (carriage return). In other words, you can press **Enter** and the statement would permit all TCP traffic. However, we are only permitting FTP traffic; therefore, enter the **eq** keyword, followed by a question mark to display the available options. Then, enter **ftp** and press **Enter**.

R1(config)# **access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ?**

<0-65535> Port number

ftp File Transfer Protocol (21)

pop3 Post Office Protocol v3 (110)

smtp Simple Mail Transport Protocol (25)

telnet Telnet (23)

www World Wide Web (HTTP, 80)

R1(config)# **access-list 100 permit tcp 172.22.34.64 0.0.0.31 host 172.22.34.62 eq ftp**

* + 1. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC1** to **Server**. Note that the access list number remains the same and a specific type of ICMP traffic does not need to be specified.

R1(config)# **access-list 100 permit icmp 172.22.34.64 0.0.0.31 host 172.22.34.62**

* + 1. All other traffic is denied, by default.
  1. Apply the ACL on the correct interface to filter traffic.

From **R1**’s perspective, the traffic that ACL 100 applies to is inbound from the network connected to Gigabit Ethernet 0/0 interface. Enter interface configuration mode and apply the ACL.

R1(config)# **interface gigabitEthernet 0/0**

R1(config-if)# **ip access-group 100 in**

* 1. Verify the ACL implementation.
     1. Ping from **PC1** to **Server**. If the pings are unsuccessful, verify the IP addresses before continuing.
     2. FTP from **PC1** to **Server**. The username and password are both **cisco**.

PC> **ftp 172.22.34.62**

* + 1. Exit the FTP service of the **Server**.

ftp> **quit**

* + 1. Ping from **PC1** to **PC2**. The destination host should be unreachable, because the traffic was not explicitly permitted.

1. Configure, Apply and Verify an Extended Named ACL
   1. Configure an ACL to permit HTTP access and ICMP.
      1. Named ACLs start with the **ip** keyword. From global configuration mode of **R1**, enter the following command, followed by a question mark.

R1(config)# **ip access-list ?**

extended Extended Access List

standard Standard Access List

* + 1. You can configure named standard and extended ACLs. This access list filters both source and destination IP addresses; therefore, it must be extended. Enter **HTTP\_ONLY** as the name. (For Packet Tracer scoring, the name is case-sensitive.)

R1(config)# **ip access-list extended HTTP\_ONLY**

* + 1. The prompt changes. You are now in extended named ACL configuration mode. All devices on the **PC2** LAN need TCP access. Enter the network address, followed by a question mark.

R1(config-ext-nacl)# **permit tcp 172.22.34.96 ?**

A.B.C.D Source wildcard bits

* + 1. An alternative way to calculate a wildcard is to subtract the subnet mask from 255.255.255.255.

255.255.255.255

- 255.255.255.240

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

= 0. 0. 0. 15

R1(config-ext-nacl)# **permit tcp 172.22.34.96 0.0.0.15 ?**

* + 1. Finish the statement by specifying the server address as you did in Part 1 and filtering **www** traffic.

R1(config-ext-nacl)# **permit tcp 172.22.34.96 0.0.0.15 host 172.22.34.62 eq www**

* + 1. Create a second access list statement to permit ICMP (ping, etc.) traffic from **PC2** to **Server**. Note: The prompt remains the same and a specific type of ICMP traffic does not need to be specified.

R1(config-ext-nacl)# **permit icmp 172.22.34.96 0.0.0.15 host 172.22.34.62**

* + 1. All other traffic is denied, by default. Exit out of extended named ACL configuration mode.
  1. Apply the ACL on the correct interface to filter traffic.

From **R1**’s perspective, the traffic that access list **HTTP\_ONLY** applies to is inbound from the network connected to Gigabit Ethernet 0/1 interface. Enter the interface configuration mode and apply the ACL.

R1(config)# **interface gigabitEthernet 0/1**

R1(config-if)# **ip access-group HTTP\_ONLY in**

* 1. Verify the ACL implementation.
     1. Ping from **PC2** to **Server**. If the pings unsuccessful, verify the IP addresses before continuing.
     2. FTP from **PC2** to **Server**. The connection should fail.
     3. Open the web browser on **PC2** and enter the IP address of **Server** as the URL. The connection should be successful.