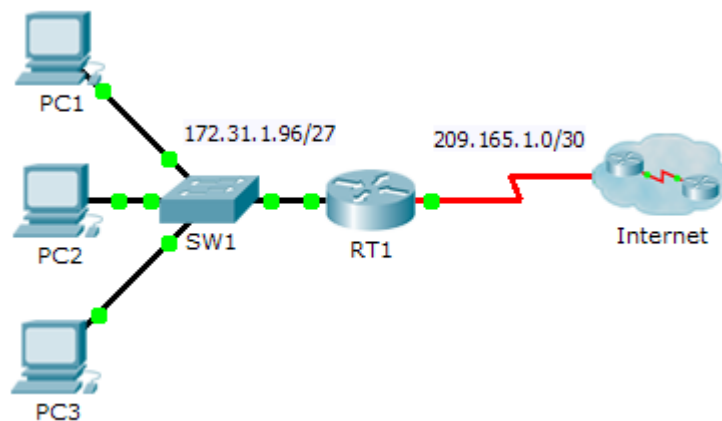


Packet Tracer - Configuring Extended ACLs - Scenario 3

Topology



Addressing Table

Device	Interface	IP Address	Subnet Mask	Default Gateway
RT1	G0/0	172.31.1.126	255.255.255.224	N/A
	S0/0/0	209.165.1.2	255.255.255.252	N/A
PC1	NIC	172.31.1.101	255.255.255.224	172.31.1.126
PC2	NIC	172.31.1.102	255.255.255.224	172.31.1.126
PC3	NIC	172.31.1.103	255.255.255.224	172.31.1.126
Server1	NIC	64.101.255.254	255.254.0.0	64.100.1.1
Server2	NIC	64.103.255.254	255.254.0.0	64.102.1.1

Objectives

Part 1: Configure a Named Extended ACL

Part 2: Apply and Verify the Extended ACL

Background / Scenario

In this scenario, specific devices on the LAN are allowed to various services on servers located on the Internet.

Part 1: Configure a Named Extended ACL

Use one named ACL to implement the following policy:

- Block HTTP and HTTPS access from **PC1** to **Server1** and **Server2**. The servers are inside the cloud and you only know their IP addresses.
- Block FTP access from **PC2** to **Server1** and **Server2**.
- Block ICMP access from **PC3** to **Server1** and **Server2**.

Note: For scoring purposes, you must configure the statements in the order specified in the following steps.

Step 1: Deny PC1 to access HTTP and HTTPS services on Server1 and Server2.

- Create an extended IP access list named ACL which will deny **PC1** access to the HTTP and HTTPS services of **Server1** and **Server2**. Because it is impossible to directly observe the subnet of servers on the Internet, four rules are required.

What is the command to begin the named ACL?

```
ip access-list extended ACL
```

- Record the statement that denies access from **PC1** to **Server1**, only for HTTP (port 80).

```
deny tcp host 172.31.1.101 host 64.101.255.254 eq 80
```

- Record the statement that denies access from **PC1** to **Server1**, only for HTTPS (port 443).

```
deny tcp host 172.31.1.101 host 64.101.255.254 eq 443
```

- Record the statement that denies access from **PC1** to **Server2**, only for HTTP.

```
deny tcp host 172.31.1.101 host 64.103.255.254 eq 80
```

- Record the statement that denies access from **PC1** to **Server2**, only for HTTPS.

```
deny tcp host 172.31.1.101 host 64.103.255.254 eq 443
```

Step 2: Deny PC2 to access FTP services on Server1 and Server2.

- Record the statement that denies access from **PC2** to **Server1**, only for FTP (port 21 only).

```
deny tcp host 172.31.1.102 host 64.101.255.254 eq 21
```

- Record the statement that denies access from **PC2** to **Server2**, only for FTP (port 21 only).

```
deny tcp host 172.31.1.102 host 64.103.255.254 eq 21
```

Step 3: Deny PC3 to ping Server1 and Server2.

- Record the statement that denies ICMP access from **PC3** to **Server1**.

```
deny icmp host 172.31.1.103 host 64.101.255.254
```

- Record the statement that denies ICMP access from **PC3** to **Server2**.

```
deny icmp host 172.31.1.103 host 64.103.255.254
```

Step 4: Permit all other IP traffic.

By default, an access list denies all traffic that does not match any rule in the list. What command permits all other traffic?

```
permit ip any any
```

Part 2: Apply and Verify the Extended ACL

The traffic to be filtered is coming from the 172.31.1.96/27 network and is destined for remote networks. Appropriate ACL placement also depends on the relationship of the traffic with respect to **RT1**.

Step 1: Apply the ACL to the correct interface and in the correct direction.

- What are the commands you need to apply the ACL to the correct interface and in the correct direction?

```
interface g0/0
```

```
ip access-group ACL in
```

Step 2: Test access for each PC.

- Access the websites of **Server1** and **Server2** using the Web Browser of **PC1** and using both HTTP and HTTPS protocols.
- Access FTP of **Server1** and **Server2** using **PC1**. The username and password is "**cisco**".
- Ping **Server1** and **Server2** from **PC1**.
- Repeat Step 2a to Step 2c with **PC2** and **PC3** to verify proper access list operation.