Network Security: Firewall

- Monitor traffic from and to your server using tables.
- Using rules, called chains, that will filter incoming and outgoing data packets.
- When a packet matches a rule, it is given a target.
    - ACCEPT - will allow the packet to pass through.
    - DROP - will not let the packet pass through.
    - RETURN - stops the packet from traversing through a chain and tell it to go back to the previous chain.

Filters

- INPUT - controls incoming packets to the server.
- FORWARD - filters incoming packets that will be forwarded somewhere else.
- OUTPUT - filter packets that are going out from your server.

Acces Control Lists (ACL)

- One of the most fundamental component of security.
- ACL are network filters or **stateless firewalls**. That only restricts, block or allow packets flowing from source to destination.
- Common in routers or firewalls but also in any network devices as hosts, servers, etc.
- Best located in routers (Email server, Webserver, ...) facing internet or a DMZ (De-Militarized zone) which is a buffer zone between a private network and hte internet.
- Better than a stateful firewall that may compromise a network.

Statelful and stateless firewall

- **Stateful firewall:** Inspect everything inside data packets. They can filter out suspicious data by examining how they behave even if the behavior was not specify by an administrator.

  Can be fooled into allowing a harmful connection. Also more susceptible to man-in-the-middle (MiTM) attacks.

  Cost more than stateless firewall.

  Stateless firewall: Make use of data packet's source, destintion and other parameters (intred by an administrator) to figure out if the data os a threat or not.

ACL Types

- **Standard ACL:** Use only source address, and provide only a **weak** security.

```
access-list access-list-number {permit|deny}
{host|source source-wildcard|any}
```

- **Extended ACL:** Both source and destination can be blocked for single or groups of hosts.

```
access-list access-list-number
[dynamic dynamic-name [timeout minutes]]
{permit|deny} tcp source source-wildcard [operator [port]]
[established] precedence precedence[][tos tos]
[log]|log-input] [time-range time-range-name]
```

- **Dynamic ACL:** Rely upon extended ACL. Also called "Lock and Key".

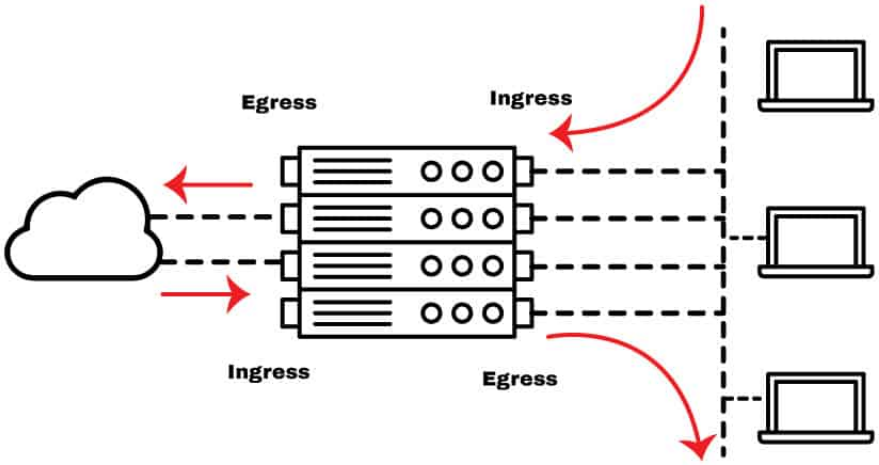
  These lists permit access to a user to a source or destination only if the user authenticates to the device via Telnet.

- **Reflexive ACL:** Reflexive ACLs are also referred to as IP session ACLs. These type of ACLs, filter traffic based on upper layer session information.

  When the session finishes, the entry is removed.

ACL in a router

In/Egress

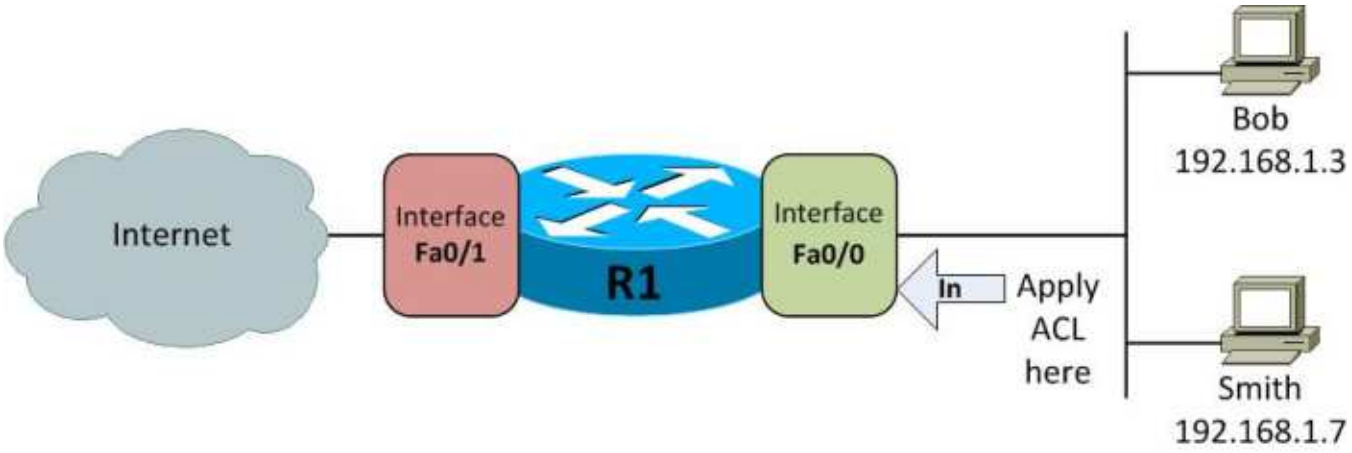


- For an ACL to work, apply it to a router's interface.
- When you create an ACL entry, the source address goes first, and the destination goes after.

```
access-list access-list-number
[dynamic dynamic-name [timeout minutes]]
{permit|deny} protocol source source source-wildcard destinatin-wildcard
        [precedence]
[log]|log-input] [time-range time-range-name]
```

ACL Example/training

Create the following network:



Creating Numbered Standard Access Lists

We will start by configuring a standard access list first in numbered and then in named format. The access list should allow Bob to access the Internet while block all access for Smith also logging unsuccessful attempts by Smith.

Let's see how can we do this using a standard access list in numbered format.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 1 permit host 192.168.1.3
R1(config)#access-list 1 deny host 192.168.1.7
R1(config)#
```

Let's now apply this access list to interface Fa0/0 in the inbound direction.

```
R1(config)#interface Fa0/0
R1(config-if)#ip access-group 1 ?
in    inbound packets
out   outbound packets
R1(config-if)#ip access-group 1 in
R1(config-if)#end
R1#
```

Named access lists have a number from 1 to 99.

In every access list there will be an **implicit deny all** at the end of the ACL even if you don't specify it explicitly. So if you configured your access list like this here is what it would do.

```
show access-list 1
access-list 1 permit host 192.168.1.3
access-list 1 deny host 192.168.1.7
access-list 1 deny any
```

Control using ping that everything works as it should.

Exercise: Create a standard ACL

Create a standard access list apply to interface Fa0/0, in order to achieve the same effect. Here, we would use the inverse mask (0.0.0.0) instead of the host keyword to match individual hosts.

Exercise: Create a standard ACL

Create a standard access list apply to interface Fa0/0, in order to achieve the same effect. Here, we would use the inverse mask (0.0.0.0) instead of the host keyword to match individual hosts.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip access-list standard Filter
R1(config-std-nacl)#permit 192.168.1.3 0.0.0.0
R1(config-std-nacl)#deny 192.168.1.7 0.0.0.0
R1(config-std-nacl)#interface Fa0/0
R1(config-if)#ip access-group Filter in
R1(config-if)#end
R1#
```

Extended ACL example

An extended access control list will allow you to deny or permit traffic from specific IP addresses, and ports.

```
access-list 110 permit tcp 92.128.2.0 0.0.0.255 any eq 80
access-list 111 permit ip 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0
```

Note also that the subnet mask in the ACL configuration in always represented with an inverse mask (i.e instead of using 255.255.255.0 we use 0.0.0.255).

Exercise: Numbered Extended Access Lists

Configure an extended access list first in numbered and then in named format. The access list should allow Bob (from our network diagram above) to access Web servers on the Internet while blocking all Web access for Smith also logging unsuccessful attempts by Smith to open a website.

Exercise: Numbered Extended Access Lists

Configure an extended access list first in numbered and then in named format. The access list should allow Bob (from our network diagram above) to access Web servers on the Internet while blocking all Web access for Smith also logging unsuccessful attempts by Smith to open a website.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#access-list 100 permit tcp host 192.168.1.3 any eq ww
R1(config)#access-list 100 deny tcp host 192.168.1.7 any eq www
R1(config)#interface Fa0/0
R1(config-if)#ip access-group 100 in
R1(config-if)#end
R1#
```

Exercise: Named Extended Access Lists

Configure the same extended access list in the named format.

Exercise: Named Extended Access Lists

Configure the same extended access list in the named format.

```
R1>enable
R1#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
R1(config)#ip access-list extended Filter
R1(config-ext-nacl)#permit tcp 192.168.1.3 0.0.0.0 any eq www
R1(config-ext-nacl)#deny tcp 192.168.1.7 0.0.0.0 any eq www log
R1(config-ext-nacl)#interface Fa0/0
R1(config-if)#ip access-group Filter in
R1(config-if)#end
R1#
```

How to apply ACL?

After you have set the ACL in place you will need to specify which direction you want it to operate on the interface that will be applied (inbound or outbound).

```
Router(config)#interface serial 0
Router(config-if)#ip access-group 111 out
```

References

https://www.cisco.com/c/en/us/support/docs/security/ios-firewall/23602-confaccesslists.html#extacls

https://www.geeksforgeeks.org/access-lists-acl/?ref=lbp

https://www.geeksforgeeks.org/extended-access-list/

Iptable: Install and Configure

1. Connect to your server via SSH

2. Execute the following command.

```
sudo apt-get update
sudo apt-get install iptables
```

3. Check the status of current iptables configuration with,

```
sudo iptables -L -v
```

4. Save the current configuration with,

```
sudo iptables-save > iptables-backup
```

Defining Chain Rules

Insert the **-A** option (**Append**)

```
sudo iptables -A
```

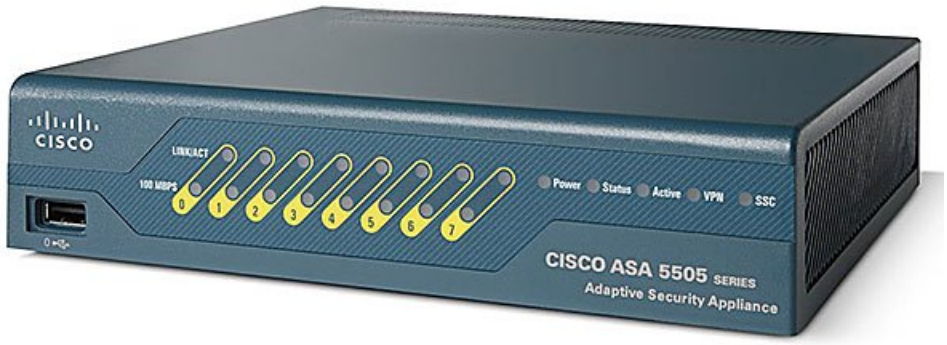
- -i (interface) - the network interface whose traffic you want to filter, such as eth0, lo, ppp0, etc.
- -p (protocol) - the network protocol where your filtering process takes place such as tcp, udp, udplite, icmp, sctp, icmpv6, etc.
- -s (source) - the address from which traffic comes from. You can add a hostname or IP address.
- -dport (destination port) - the destination port number of a protocol, such as 22 (SSH), 443 (https), etc.
- -j (target) - the target name (ACCEPT, DROP, RETURN).

References

https://www.hostinger.com/tutorials/iptables-tutorial

Private Internet Exchange (PIX)

Cisco Systems' Firewall.



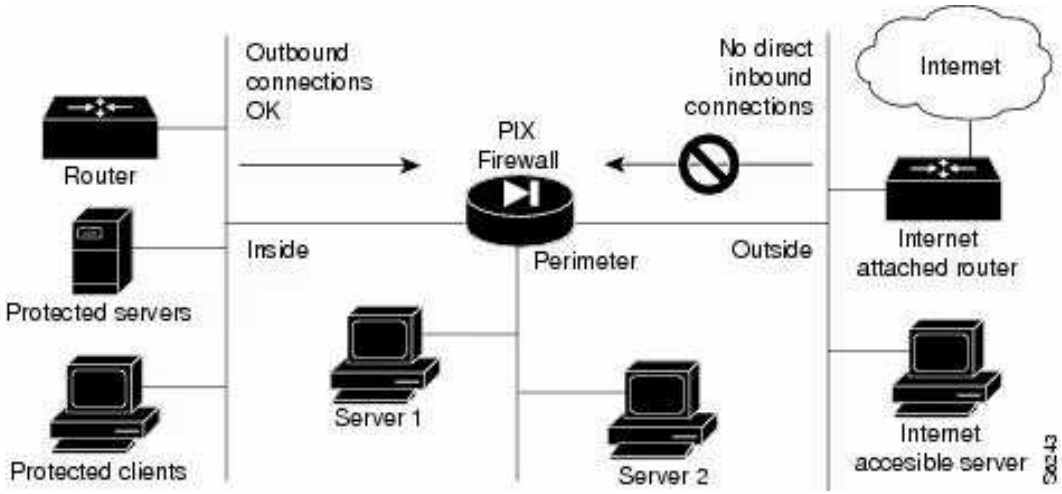
Creating a Security Policy.

Step 1 Draw a map of your complete network. Have paper copy!

Step 2 Identify the devices you need to protect.

Step 3 Identify which inside servers need to be visible on the outside.

Step 4 Identify which router features you need to set the PIX.

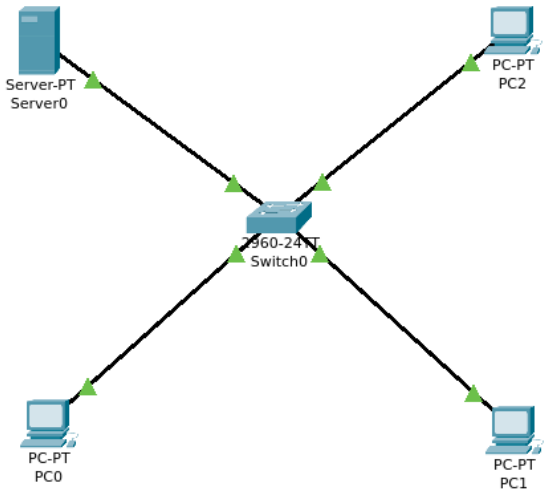

Outbound
connections
OK
No direct
inbound
connections
Internet
PIX
Firewall
Router
Inside
Perimeter
Outside
Internet
attached router
Protected servers
Server 1
Server 2
Internet
accesible server
Protected clients
362143

Adaptive Security Algorithm (ASA)

- **Stateful approach to security**: Every inbound packet is checked against ASA.
- No packets can transverse the PIX Firewall without a connection and a state.
- UDP packets are treated as TCP packets.

Basic configuration

Create the following network:



Using the following addressing table:

Device	IPv4 Address	Subnet Mask
Server	1.0.0.1	255.0.0.0
PC0	1.0.0.2	255.0.0.0
PC1	1.0.0.3	255.0.0.0
PC2	1.0.0.4	255.0.0.0

Blocking packets and allowing web browser.

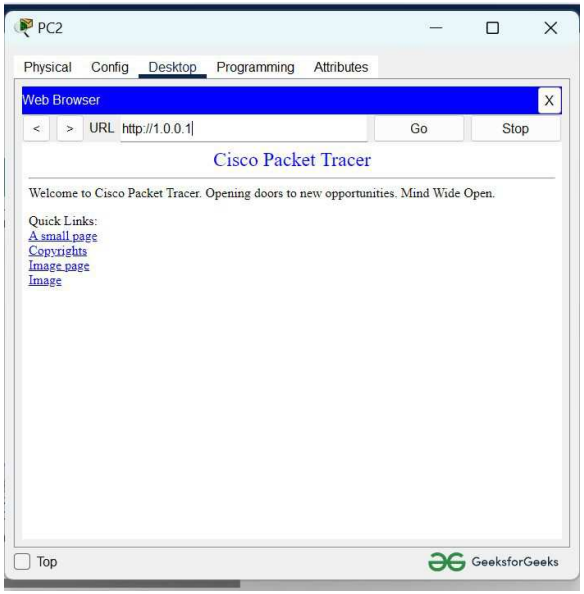
1. Click on Server0 then go to the desktop.
2. Then click on firewall IPv4.
3. Turn on the services.
4. First, Deny the ICMP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
5. Then, allow the IP protocol and set remote IP to 0.0.0.0 and Remote wildcard mask to 255.255.255.255.
6. And add them.

Verifying the network

1. We will use the ping command to do so.
2. First, click on PC2 then Go to the command prompt.
3. Then type ping <IP address of targeted node>.
4. We will ping the IP address of the server0.
5. As we can see in the below image we are getting no replies which means the packets are blocked.

Check the web browser

- Click on PC2 and go to desktop then web browser.
- Enter the IP address 1.0.0.1 in the URL.



References

https://www.cisco.com/en/US/docs/security/pix/pix50/configuration/guide/config.html

Cisco Adaptive Security Appliance (ASA)

The last software for ASA for Cisco ASA family.

Cisco Adaptive Security Appliance (ASA)

- Offers integrated intrusion prevention system (IPS) and VPN
- Helps organizations increase capacity and improve performance through high-performance, multi-site, multi-node clustering
- Delivers high availability for high resiliency applications
- Provides collaboration between physical and virtual devices
- Meets the unique needs of networks and data centers
- Facilitates dynamic routing and site-to-site VPN on a per-context basis
- Supports next-generation encryption standards

Intrusion prevention systems (IPS)

Intrusion prevention systems work by scanning all network traffic. There are a number of different threats that an IPS is designed to prevent, including:

- Denial of Service (DoS) attack
- Distributed Denial of Service (DDoS) attack
- Various types of exploits
- Worms
- Viruses

Types of prevention

- **Signature-Based** - The signature-based approach uses predefined signatures of well-known network threats. When an attack is initiated that matches one of these signatures or patterns, the system takes necessary action.
- **Anomaly-Based** - The anomaly-based approach monitors for any abnormal or unexpected behavior on the network. If an anomaly is detected, the system blocks access to the target host immediately.
- **Policy-Based** - This approach requires administrators to configure security policies according to organizational security policies and the network infrastructure. When an activity occurs that violates a security policy, an alert is triggered and sent to the system administrators.

Next Generation Firewall (NGFW)

From *ForcePoint* provides advanced intrusion prevention and detection for any network, allowing you to respond to threats within minutes, not hours, and protect your most critical data and application assets.

Challenge no 2

IPv4 ACL Implementation with PT

Device	Interface	IP Address
Branch	G0/0/0	192.168.1.1/26
	G0/0/1	192.168.1.65/29
	S0/1/0	192.0.2.1/30
	S0/1/1	192.168.3.1/30
HQ	G0/0/0	192.168.2.1/27
	G0/0/1	192.168.2.33/28
	S0/1/1	192.168.3.2/30
PC-1	NIC	192.168.1.10/26
PC-2	NIC	192.168.1.20/26
PC-3	NIC	192.168.1.30/26
Admin	NIC	192.168.1.67/29
Enterprise Web Server	NIC	192.168.1.70/29
Branch PC	NIC	192.168.2.17/27
Branch Server	NIC	192.168.2.45/28
Internet User	NIC	198.51.100.218/24
External Web Server	NIC	203.0.113.73/24

Objectives

- Configure a router with standard named ACLs.
- Configure a router with extended named ACLs.
- Configure a router with extended ACLs to meet specific communication requirements.
- Configure an ACL to control access to network device terminal lines.
- Configure the appropriate router interfaces with ACLs in the appropriate direction.
- Verify the operation of the configured ACLs.

Background/Scenario

In this activity you will configure extended, standard named, and extended named ACLs to meet specified communication requirements. Instructions

Step 1: Verify Connectivity in the New Company Network

First, test connectivity on the network as it is before configuring the ACLs. All hosts should be able to ping all other hosts.

Step 2: Configure Standard and Extended ACLs per Requirements.

Configure ACLs to meet the following requirements:

Important guidelines:

Do not use explicit deny any statements at the end of your ACLs. Use shorthand (host and any) whenever possible. Write your ACL statements to address the requirements in the order that they are specified here.

Place your ACLs in the most efficient location and direction.

ACL 1 Requirements

Create ACL 101. Explicitly block FTP access to the Enterprise Web Server from the internet. No ICMP traffic from the internet should be allowed to any hosts on HQ LAN 1 Allow all other traffic.

ACL 2 Requirements

Use ACL number 111 No hosts on HQ LAN 1 should be able to access the Branch Server. All other traffic should be permitted.

ACL 3: Requirements

Create a named standard ACL. Use the name vty_block. The name of your ACL must match this name exactly. Only addresses from the HQ LAN 2 network should be able to access the VTY lines of the HQ router.

ACL 4: Requirements

Create a named extended ACL called branch_to_hq. The name of your ACL must match this name exactly. No hosts on either of the Branch LANs should be allowed to access HQ LAN 1. Use one access list statement for each of the Branch LANs. All other traffic should be allowed.

Step 3: Verify ACL Operation.

Perform the following connectivity tests between devices in the topology. Note whether or not they are successful.

Note: Use the show ip access-lists command to verify ACL operation. Use the clear access list counters command to reset the match counters.

Questions:

1. Send a ping request from Branch PC to the Enterprise Web Server. Was it successful? Explain.

2. Which ACL statement permitted or denied the ping between these two devices? List the access list name or number, the router on which it was applied, and the specific line that the traffic matched.

3. Attempt to ping from PC-1 on the HQ LAN 1 to the Branch Server. Was it successful? Explain.

4. Which ACL statement permitted or denied the ping between these two devices?

5. Open a web browser on the External Server and attempt to bring up a web page stored on the Enterprise Web Server. Is it successful? Explain.

6. Which ACL statement permitted or denied the ping between these two devices?

7. Test connections to an internal server from the internet. From the command line on the Internet User PC, attempt to make an FTP connection to the Branch Server. Is the FTP connection successful?

8. Which access list should be modified to prevent users from the Internet to make FTP connections to the Branch Server?

9. Which statement(s) should be added to the access list to deny this traffic?

Configure Numbered Standard IPv4 ACL

Device	Interface	IP Address	Subnet Mask	Default Gateway
R1	G0/0	192.168.10.1	255.255.255.0	N/A
	G0/1	192.168.11.1	255.255.255.0	
	S0/0/0	10.1.1.1	255.255.255.252	
	S0/0/1	10.3.3.1	255.255.255.252	
R2	G0/0	192.168.20.1	255.255.255.0	N/A
	S0/0/0	10.1.1.2	255.255.255.252	
	S0/0/1	10.2.2.1	255.255.255.252	
R3	G0/0	192.168.30.1	255.255.255.0	N/A
	S0/0/0	10.3.3.2	255.255.255.252	
	S0/0/1	10.2.2.2	255.255.255.252	
PC1	NIC	192.168.10.10	255.255.255.0	192.168.10.1
PC2	NIC	192.168.11.10	255.255.255.0	192.168.11.1
PC3	NIC	192.168.30.10	255.255.255.0	192.168.30.1
WebServer	NIC	192.168.20.254	255.255.255.0	192.168.20.1

Create a network with the previous addressing table.

Objectives

Part 1: Plan an ACL Implementation

Part 2: Configure, Apply, and Verify a Standard ACL

Background / Scenario

Standard access control lists (ACLs) are router configuration scripts that control whether a router permits or denies packets based on the source address. This activity focuses on defining filtering criteria, configuring standard ACLs, applying ACLs to router interfaces, and verifying and testing the ACL implementation. The routers are already configured, including IP addresses and Enhanced Interior Gateway Routing Protocol (EIGRP) routing.

Part 1: Plan an ACL Implementation

Step 1: Investigate the current network configuration.

Before applying any ACLs to a network, it is important to confirm that you have full connectivity. Verify that the network has full connectivity by choosing a PC and pinging other devices on the network. You should be able to successfully ping every device.

Step 2: Evaluate two network policies and plan ACL implementations.

The following network policies are implemented on R2:

The 192.168.11.0/24 network is not allowed access to the WebServer on the 192.168.20.0/24 network.

All other access is permitted.

To restrict access from the 192.168.11.0/24 network to the WebServer at 192.168.20.254 without interfering with other traffic, an ACL must be created on R2. The access list must be placed on the outbound interface to the WebServer. A second rule must be created on R2 to permit all other traffic.

The following network policies are implemented on R3:

The 192.168.10.0/24 network is not allowed to communicate with the 192.168.30.0/24 network.

All other access is permitted.

To restrict access from the 192.168.10.0/24 network to the 192.168.30/24 network without interfering with other traffic, an access list will need to be created on R3. The ACL must be placed on the outbound interface to PC3. A second rule must be created on R3 to permit all other traffic.

Part 2: Configure, Apply, and Verify a Standard ACL

Step 1: Configure and apply a numbered standard ACL on R2.

Create an ACL using the number 1 on R2 with a statement that denies access to the 192.168.20.0/24 network from the 192.168.11.0/24 network.

Open configuration window

R2(config)# access-list 1 deny 192.168.11.0 0.0.0.255

By default, an access list denies all traffic that does not match any rules. To permit all other traffic, configure the following statement:

R2(config)# access-list 1 permit any

Before applying an access list to an interface to filter traffic, it is a best practice to review the contents of the access list, in order to verify that it will filter traffic as expected.

```
R2# show access-lists
Standard IP access list 1
10 deny 192.168.11.0 0.0.0.255
20 permit any
```

For the ACL to actually filter traffic, it must be applied to some router operation. Apply the ACL by placing it for outbound traffic on the GigabitEthernet 0/0 interface. Note: In an actual operational network, it is not a good practice to apply an untested access list to an active interface.

R2(config)# interface GigabitEthernet0/0

R2(config-if)# ip access-group 1 out

Step 2: Configure and apply a numbered standard ACL on R3.

Create an ACL using the number 1 on R3 with a statement that denies access to the 192.168.30.0/24 network from the PC1 (192.168.10.0/24) network.

R3(config)# access-list 1 deny 192.168.10.0 0.0.0.255

By default, an ACL denies all traffic that does not match any rules. To permit all other traffic, create a second rule for ACL 1.

R3(config)# access-list 1 permit any

Verify that the access list is configured correctly.

```
R3# show access-lists
Standard IP access list 1
10 deny 192.168.10.0 0.0.0.255
20 permit any
```

Apply the ACL by placing it for outbound traffic on the GigabitEthernet 0/0 interface.

```
R3(config)# interface GigabitEthernet0/0
R3(config-if)# ip access-group 1 out
```

Step 3: Verify ACL configuration and functionality.

Enter the show run or show ip interface gigabitethernet 0/0 command to verify the ACL placements.

With the two ACLs in place, network traffic is restricted according to the policies detailed in Part 1. Use the following tests to verify the ACL implementations:

```
A ping from 192.168.10.10 to 192.168.11.10 succeeds.
A ping from 192.168.10.10 to 192.168.20.254 succeeds.
A ping from 192.168.11.10 to 192.168.20.254 fails.
A ping from 192.168.10.10 to 192.168.30.10 fails.
A ping from 192.168.11.10 to 192.168.30.10 succeeds.
```

A ping from 192.168.30.10 to 192.168.20.254 succeeds.

Issue the show access-lists command again on routers R2 and R3. You should see output that indicates the number of packets that have matched each line of the access list.

Note: The number of matches shown for your routers may be different, due to the number of pings that are sent and received.

```
R2# show access-lists
Standard IP access list 1
10 deny 192.168.11.0 0.0.0.255 (4 match(es))
20 permit any (8 match(es))
R3# show access-lists
Standard IP access list 1
10 deny 192.168.10.0 0.0.0.255 (4 match(es))
20 permit any (8 match(es))
```