Module 8: Access Control Lists

Instructor Materials

Networking Security v1.0
(NETSEC)

Module Objectives

Module Title: Access Control Lists

Module Objective: Implement access control lists (ACLs) to filter traffic and mitigate network attacks on a network.

Topic Title	Topic Objective
Introduction to Access Control Lists	Describe standard and extended ACLs.
Wildcard Masks	Explain how ACLs use wildcard masks.
Configure ACLs	Explain how to configure ACLs.
Modify ACLs	Use sequence numbers to edit existing standard IPv4 ACLs.
Implement ACLs	Implement ACLs.
Mitigate Attacks with ACLs	Use ACLs to mitigate common network attacks.
IPv6 ACLs	Configure IPv6 ACLs using CLI.

8.1 Introduction to Access Control Lists

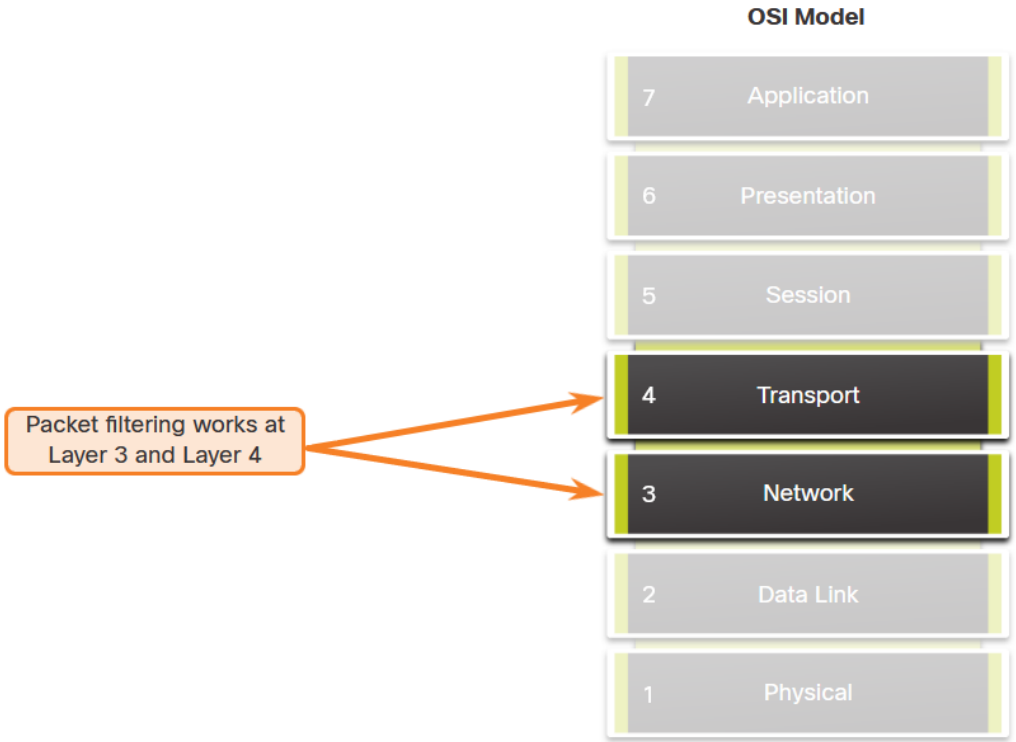
What is an ACL?

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header.

Task	Example
Limit network traffic to increase network performance	• A corporate policy prohibits video traffic on the network to reduce the network load. • A policy can be enforced using ACLs to block video traffic.
Provide traffic flow control	• A corporate policy requires that routing protocol traffic be limited to certain links only. • A policy can be implemented using ACLs to restrict the delivery of routing updates to only those that come from a known source.
Provide a basic level of security for network access	• Corporate policy demands that access to the Human Resources network be restricted to authorized users only. • A policy can be enforced using ACLs to limit access to specified networks.
Filter traffic based on traffic type	• Corporate policy requires that email traffic be permitted into a network, but that Telnet access be denied. • A policy can be implemented using ACLs to filter traffic by type.
Screen hosts to permit or deny access to network services	• Corporate policy requires that access to some file types (e.g., FTP or HTTP) be limited to user groups. • A policy can be implemented using ACLs to filter user access to services.
Provide priority to certain classes of network traffic	• Corporate traffic specifies that voice traffic be forwarded as fast as possible to avoid any interruption. • A policy can be implemented using ACLs and QoS services to identify voice traffic and process it immediately.

Packet Filtering

Packet filtering controls access to a network by analyzing the incoming and/or outgoing packets and forwarding them or discarding them based on given criteria. Packet filtering can occur at Layer 3 or Layer 4. Cisco routers support standard and extended ACLs.



Numbered and Named ACLs

Numbered ACLs - ACLs number 1 to 99, or 1300 to 1999 are standard ACLs while ACLs number 100 to 199, or 2000 to 2699 are extended ACLs, as shown in the output.

```
R1(config)# access-list ?
  <1-99>       IP standard access list
  <100-199>    IP extended access list
  <1100-1199>  Extended 48-bit MAC address access list
  <1300-1999>  IP standard access list (expanded range)
  <200-299>    Protocol type-code access list
  <2000-2699>  IP extended access list (expanded range)
  <700-799>    48-bit MAC address access list
  rate-limit   Simple rate-limit specific access list
  template     Enable IP template acls
Router(config)# access-list
```

Named ACLs - Named ACLs is the preferred method to use when configuring ACLs. Specifically, standard and extended ACLs can be named to provide information about the purpose of the ACL. The **ip access-list** global configuration command is used to create a named ACL.

```
R1(config)# ip access-list extended FTP-FILTER
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq ftp-data
R1(config-ext-nacl)#
```

ACL Operation

ACLs define the set of rules that give added control for packets that enter inbound interfaces, packets that relay through the router, and packets that exit outbound interfaces of the router.

An inbound ACL filters packets before they are routed to the outbound interface. If the packet is permitted by the ACL, it is then processed for routing. Inbound ACLs are best used to filter packets when the network attached to an inbound interface is the only source of packets that need to be examined.

An outbound ACL filters packets after being routed, regardless of the inbound interface. Incoming packets are routed to the outbound interface and then they are processed through the outbound ACL. Outbound ACLs are best used when the same filter will be applied to packets coming from multiple inbound interfaces before exiting the same outbound interface.

8.2 Wildcard Masking

Wildcard Mask Overview

A wildcard mask is like a subnet mask in that it uses the ANDing process to identify which bits in an IPv4 address to match. However, they differ in the way they match binary 1s and 0s. Unlike a subnet mask, in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask, the reverse is true.

Wildcard Mask	Last Octet (in Binary)	Meaning (0 - match, 1 - ignore)
0.0.0.0	00000000	Match all octets.
0.0.0.63	00111111	• Match the first three octets • Match the two left most bits of the last octet • Ignore the last 6 bits
0.0.0.15	00001111	• Match the first three octets • Match the four left most bits of the last octet • Ignore the last 4 bits of the last octet
0.0.0.252	11111100	• Match the first three octets • Ignore the six left most bits of the last octet • Match the last two bits
0.0.0.255	11111111	• Match the first three octet • Ignore the last octet

Wildcard Mask Types

Wildcard Mask to Match a Host

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.0	00000000.00000000.00000000.00000000
Permitted IPv4 Address	**192.168.1.1**	11000000.10101000.00000001.00000001

Wildcard Mask to Match an IPv4 Subnet

	Decimal	Binary
IPv4 address	192.168.1.1	11000000.10101000.00000001.00000001
Wildcard Mask	0.0.0.255	00000000.00000000.00000000.11111111
Permitted IPv4 Address	**192.168.1.0/24**	11000000.10101000.00000001.00000000

Wildcard Mask to Match an IPv4 Address Range

	Decimal	Binary
IPv4 address	192.168.16.0	11000000.10101000.00010000.00000000
Wildcard Mask	0.0.15.255	00000000.00000000.00001111.11111111
Permitted IPv4 Address	**192.168.16.0/24 to 192.168.31.0/24**	11000000.10101000.00010000.00000000 11000000.10101000.00011111.00000000

Wildcard Mask Calculation

Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255.

Assume you wanted an ACE in ACL 10 to permit access to all users in the 192.168.3.0/24 network. To calculate the wildcard mask, subtract the subnet mask (i.e., 255.255.255.0) from 255.255.255.255, as shown in the table.

Starting value	255.255.255.255
Subtract the subnet mask	- 255.255.255. 0
Resulting wildcard mask	0. 0. 0.255

The solution produces the wildcard mask 0.0.0.255. Therefore, the ACE would be **access-list 10 permit 192.168.3.0 0.0.0.255**.

Wildcard Mask Keywords

Keywords reduce ACL keystrokes and make it easier to read the ACE:

- **host** - This keyword substitutes for the 0.0.0.0 mask. This mask states that all IPv4 address bits must match to filter just one host address.
- **any** - This keyword substitutes for the 255.255.255.255 mask. This mask says to ignore the entire IPv4 address or to accept any addresses.

For example, these ACL commands…

```
R1(config)# access-list 10 permit 192.168.10.10  0.0.0.0
R1(config)# access-list 11 permit  0.0.0.0 255.255.255.255
R1(config)#
```

…can be rewritten as follows:

```
R1(config)# access-list 10 permit  host  192.168.10.10
R1(config)# access-list 11 permit  any
R1(config)#
```

8.3 Configure ACLs

Create an ACL

When configuring a complex ACL, it is suggested that you:

- Use a text editor and write out the specifics of the policy to be implemented.
- Add the IOS configuration commands to accomplish those tasks.
- Include remarks to document the ACL.
- Copy and paste the commands onto the device.
- Always thoroughly test an ACL to ensure that it correctly applies the desired policy.

Numbered Standard IPv4 ACL Syntax

To create a numbered standard ACL, use the following global configuration command:

```
Router(config)# access-list access-list-number {deny | permit | remark text} source [source-
wildcard] [log]
```

Use the **no access-list** *access-list-number* global configuration command to remove a numbered standard ACL.

Numbered Standard IPv4 ACL Syntax (Cont.)

This table provides a detailed explanation of the syntax for a standard ACL.

Parameter	Description
access-list-number	• This is the decimal number of the ACL. • Standard ACL number range is 1 to 99 or 1300 to 1999.
deny	This denies access if the condition is matched.
permit	This permits access if the condition is matched.
remark *text*	• (Optional) This adds a text entry for documentation purposes. • Each remark is limited to 100 characters.
source	• This identifies the source network or host address to filter. • Use the **any** keyword to specify all networks. • Use the **host** *ip-address* keyword or simply enter an ip-address (without the host keyword) to identify a specific IP address.
source-wildcard	(Optional) This is a 32-bit wildcard mask that is applied to the . If omitted, a default 0.0.0.0 mask is assumed.
log	• (Optional) This keyword generates and sends an informational message whenever the ACE is matched. • Message includes ACL number, matched condition (i.e., permitted or denied), source address, and number of packets.{`{" "}`} • This message is generated for the first matched packet. • This keyword should only be implemented for troubleshooting or security reasons.

Named Standard IPv4 ACL Syntax

ACL names are alphanumeric, case sensitive, and must be unique. Capitalizing ACL names is recommended.

To create a named standard ACL, use the following global configuration command:

```
Router(config)# ip access-list standard access-list-name
```

In the example, a named standard IPv4 ACL called **NO-ACCESS** is created. Notice that the prompt changes to named standard ACL configuration mode. Use the help facility to view all the named standard ACL ACE options.

```
R1(config)# ip access-list standard NO-ACCESS
R1(config-std-nacl)# ?
Standard Access List configuration commands:
  <1-2147483647>  Sequence Number
  default         Set a command to its defaults
  deny            Specify packets to reject
  exit            Exit from access-list configuration mode
  no              Negate a command or set its defaults
  permit          Specify packets to forward
  remark          Access list entry comment
R1(config-std-nacl)#
```

Numbered Extended IPv4 ACL Syntax

The procedural steps for configuring extended ACLs are the same as for standard ACLs. The extended ACL is first configured, and then it is activated on an interface. However, the command syntax and parameters are more complex to support the additional features provided by extended ACLs.

To create a numbered extended ACL, use the following global configuration command:

```
Router(config)# access-list access-list-number {deny | permit | remark text} protocol source
source-wildcard [ operator {port}] destination destination-wildcard [operator {port}]
[established] [log]
```

The parameters are reviewed on the next two slides.

The command to apply an extended IPv4 ACL to an interface is the same as the command used for standard IPv4 ACLs.

```
Router(config-if)# ip access-group {access-list-number | access-list-name} {in | out}
```

Numbered Extended IPv4 ACL Syntax (Cont.)

Although there are many keywords and parameters for extended ACLs, it is not necessary to use all of them when configuring an extended ACL. The table provides a detailed explanation of the syntax for an extended ACL.

Parameter	Description
access-list-number	This is the decimal number of the ACL. Extended ACL number range is 100 to 199 and 2000 to 2699.
deny	This denies access if the condition is matched.
permit	This permits access if the condition is matched.
remark *text*	• (Optional) Adds a text entry for documentation purposes. • Each remark is limited to 100 characters.
protocol	• Name or number of an internet protocol. • Common keywords include **ip**, **tcp**, **udp**, and **icmp**. • The **ip** keyword matches all IP protocols.
source	• This identifies the source network or host address to filter. • Use the **any** keyword to specify all networks. • Use the **host** *ip-address* keyword or simply enter an *ip-address* (without the **host** keyword) to identify a specific IP address.
source-wildcard	(Optional) A 32-bit wildcard mask that is applied to the source.

(table continued on next slide)

Numbered Extended IPv4 ACL Syntax (Cont.)

Parameter	Description
destination	• This identifies the destination network or host address to filter. • Use the **any** keyword to specify all networks. • Use the **host** *ip-address* keyword or *ip-address*.
destination-wildcard	(Optional) This is a 32-bit wildcard mask that is applied to the destination.
operator	• (Optional) This compares source or destination ports. • Some operators include **lt** (less than), **gt** (greater than), **eq** (equal), and **neq** (not equal).
port	(Optional) The decimal number or name of a TCP or UDP port.
established	• (Optional) For the TCP protocol only. • This is a 1st generation firewall feature.
log	• (Optional) This keyword generates and sends an informational message whenever the ACE is matched. • This message includes ACL number, matched condition (i.e., permitted or denied), source address, and number of packets. • This message is generated for the first matched packet. • This keyword should only be implemented for troubleshooting or security reasons.

Protocols and Port Numbers

Protocol Options - The four highlighted protocols are the most popular options. Use the **?** to get help when entering a complex ACE. If an internet protocol is not listed, then the IP protocol number could be specified. For instance, the ICMP protocol number 1, TCP is 6, and UDP is 17.

```
R1(config)# access-list 100 permit ?
  <0-255>        An IP protocol number
  ahp            Authentication Header Protocol
  dvmrp          dvmrp
  eigrp          Cisco's EIGRP routing protocol
  esp            Encapsulation Security Payload
  gre            Cisco's GRE tunneling
  icmp           Internet Control Message Protocol
  igmp           Internet Gateway Message Protocol
  ip             Any Internet Protocol
  ipinip         IP in IP tunneling
  nos            KA9Q NOS compatible IP over IP tunneling
  object-group   Service object group
  ospf           OSPF routing protocol
  pcp            Payload Compression Protocol
  pim            Protocol Independent Multicast
  tcp            Transmission Control Protocol
  udp            User Datagram Protocol
R1(config)# access-list 100 permit
```

Protocols and Port Numbers (Cont.)

Port Keyword Options - Selecting a protocol influences port options. For instance, selecting the:

- **tcp** protocol would provide TCP related ports options
- **udp** protocol would provide UDP specific ports options
- **icmp** protocol would provide ICMP related ports (i.e., message) options

Notice how many TCP port options are available. The highlighted ports are popular options. Port names or number can be specified. However, port names make it easier to understand the purpose of an ACE. Notice how some common ports names (e.g., SSH and HTTPS) are not listed. For these protocols, port numbers will have to be specified.

```
R1(config)# access-list 100 permit tcp any any eq ?
  <0-65535>     Port number
  bgp           Border Gateway Protocol (179)
  chargen       Character generator (19)
  cmd           Remote commands (rcmd, 514)
  daytime       Daytime (13)
  discard       Discard (9)
  domain        Domain Name Service (53)
  echo          Echo (7)
  exec          Exec (rsh, 512)
  finger        Finger (79)
  ftp           File Transfer Protocol (21)
  ftp-data      FTP data connections (20)
  gopher        Gopher (70)
  hostname      NIC hostname server (101)
  ident         Ident Protocol (113)
  irc           Internet Relay Chat (194)
  klogin        Kerberos login (543)
  kshell        Kerberos shell (544)
  login         Login (rlogin, 513)
  lpd           Printer service (515)
  msrpc         MS Remote Procedure Call (135)
  nntp          Network News Transport Protocol (119)
  onep-plain    Onep Cleartext (15001)
  onep-tls      Onep TLS (15002)
  pim-auto-rp   PIM Auto-RP (496)
  pop2          Post Office Protocol v2 (109)
  pop3          Post Office Protocol v3 (110)
  smtp          Simple Mail Transport Protocol (25)
  sunrpc        Sun Remote Procedure Call (111)
  syslog        Syslog (514)
  tacacs        TAC Access Control System (49)
  talk          Talk (517)
  telnet        Telnet (23)
  time          Time (37)
  uucp          Unix-to-Unix Copy Program (540)
  whois         Nicname (43)
  www           World Wide Web (HTTP, 80)
```

Protocols and Port Numbers Configuration Examples

Extended ACLs can filter on different port number and port name options. This example configures an extended ACL 100 to filter HTTP traffic. The first ACE uses the www port name. The second ACE uses the port number 80. Both ACEs achieve exactly the same result.

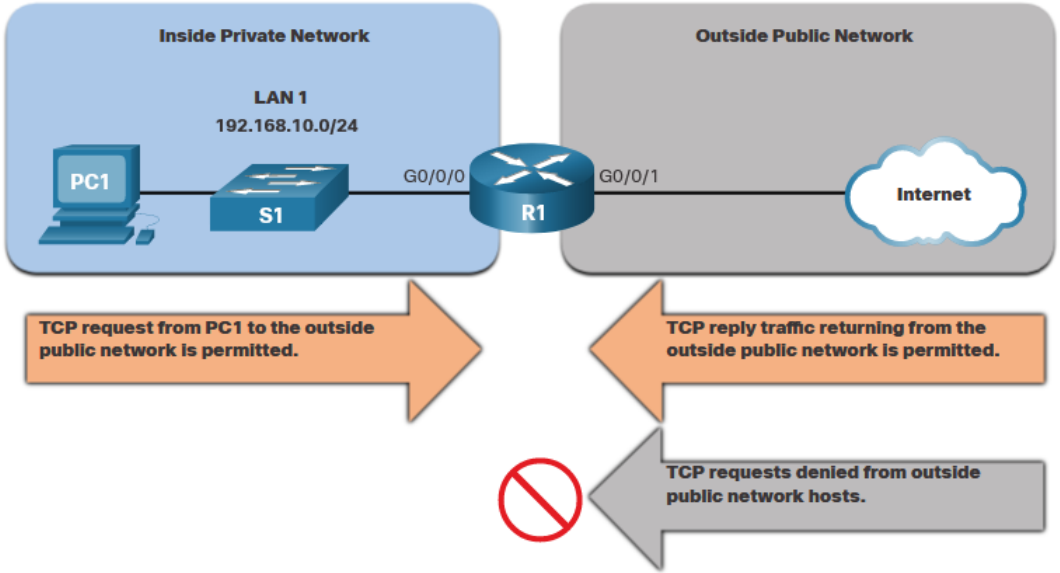
```
R1(config)# access-list 100 permit tcp any any eq www !or...
R1(config)# access-list 100 permit tcp any any eq 80
```

Configuring the port number is required when there is not a specific protocol name listed such as SSH (port number 22) or an HTTPS (port number 443)

```
R1(config)# access-list 100 permit tcp any any eq 22 R1(config)# access-list 100 permit tcp any any
eq 443
R1(config)#
```

TCP Established Extended ACL

TCP can also perform basic stateful firewall services using the TCP **established** keyword. The keyword enables inside traffic to exit the inside private network and permits the returning reply traffic to enter the inside private network. However, TCP traffic generated by an outside host and attempting to communicate with an inside host is denied. The established keyword can be used to permit only the return HTTP traffic from requested websites, while denying all other traffic.



TCP Established Extended ACL (Cont.)

In this example, ACL 120 is configured to only permit returning web traffic to the inside hosts. The new ACL is then applied outbound on the R1 G0/0/0 interface. The show access-lists command displays both ACLs. Notice from the match statistics that inside hosts have been accessing the secure web resources from the internet.

```
R1(config)# access-list 120 permit tcp any 192.168.10.0 0.0.0.255 established
R1(config)# interface g0/0/0
R1(config-if)# ip access-group 120 out
R1(config-if)# end
R1# show access-lists
Extended IP access list 110
     10 permit tcp 192.168.10.0 0.0.0.255 any eq www
     20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (657 matches)
Extended IP access list 120
     10 permit tcp any 192.168.10.0 0.0.0.255 established (1166 matches)
R1#
```

Named Extended IPv4 ACL Syntax

Naming an ACL makes it easier to understand its function. This command enters the named extended configuration mode. Recall that ACL names are alphanumeric, case sensitive, and must be unique. To create a named extended ACL, use the following global configuration command:

```
Router(config)# ip access-list extended access-list-name
```

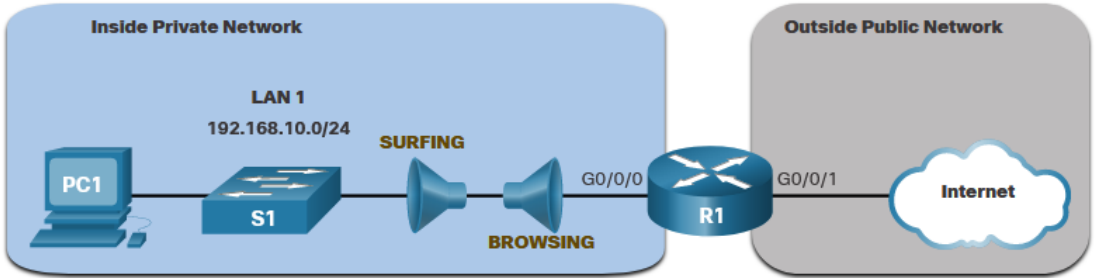
In the example, a named extended ACL called NO-FTP-ACCESS is created and the prompt changed to named extended ACL configuration mode. ACE statements are entered in the named extended ACL sub configuration mode.

```
R1(config)# ip access-list extended NO-FTP-ACCESS
R1(config-ext-nacl)#
```

Named Extended IPv4 ACL Example

Named extended ACLs are created in essentially the same way that named standard ACLs are created. The topology in the figure is used to demonstrate configuring and applying two named extended IPv4 ACLs to an interface:

- SURFING - This will permit inside HTTP and HTTPS traffic to exit to the internet.
- BROWSING - This will only permit returning web traffic to the inside hosts while all other traffic exiting the R1 G0/0/0 interface is implicitly denied.



Named Extended IPv4 ACL Example (Cont.)

The SURFING ACL permits HTTP and HTTPS traffic from inside users to exit the G0/0/1 interface connected to the internet. Web traffic returning from the internet is permitted back into the inside private network by the BROWSING ACL.
The SURFING ACL is applied inbound and the BROWSING ACL applied outbound on the R1 G0/0/0 interface.

Inside hosts have been accessing the secure web resources from the internet. The **show access-lists** command is used to verify the ACL statistics.

```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# Remark Permits inside HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)#
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# Remark Only permit returning HTTP and HTTPS traffic
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
R1(config-if)# end
R1# show access-lists
Extended IP access list SURFING
    10 permit tcp 192.168.10.0 0.0.0.255 any eq www
    20 permit tcp 192.168.10.0 0.0.0.255 any eq 443 (124 matches)
Extended IP access list BROWSING
    10 permit tcp any 192.168.10.0 0.0.0.255 established (369 matches)
R1#
```

8.4 Modify ACLs

Two Methods to Modify an ACL

After an ACL is configured, it may need to be modified. ACLs with multiple ACEs can be complex to configure. Sometimes the configured ACE does not yield the expected behaviors. For these reasons, ACLs may initially require a bit of trial and error to achieve the desired filtering result. There are two methods to use when modifying an ACL:

- Use a Text Editor
- Use Sequence Numbers

Text Editor Method

ACLs with multiple ACEs should be created in a text editor. This allows you to plan the required ACEs, create the ACL, and then paste it into the router interface. It also simplifies the tasks to edit and fix an ACL. To modify an ACL using a text editor:

- Copy the ACL from the running configuration and paste it into the text editor.
- Make the necessary edits changes.
- Remove the previously configured ACL on the router otherwise, pasting the edited ACL commands will only append (i.e., add) to the existing ACL ACEs on the router.
- Copy and paste the edited ACL back to the router.

Sequence Number Method

An ACL ACE can also be deleted or added using the ACL sequence numbers. Sequence numbers are automatically assigned when an ACE is entered. These numbers are listed in the **show access-lists** command. The **show running-config** command does not display sequence numbers.

Use the **ip access-list standard** command to edit an ACL. Statements cannot be overwritten using the same sequence number as an existing statement. Therefore, the current statement must be deleted first with the **no 10** command. Then the correct ACE can be added using sequence number 10 as configured. Verify the changes using the **show access-lists** command.

```
R1# conf t
R1(config)# ip access-list standard 1
R1(config-std-nacl)# no 10
R1(config-std-nacl)# 10 deny host 192.168.10.10
R1(config-std-nacl)# end
R1# show access-lists
Standard IP access list 1
    10 deny   192.168.10.10
    20 permit 192.168.10.0, wildcard bits 0.0.0.255
R1#
```

8.5 Implement ACLs

ACL Configuration Guidelines

An ACL is made up of one or more access control entries (ACEs) or statements. When configuring and applying an ACL, be aware of the guidelines summarized in this list:

- Create an ACL globally and then apply it.
- Ensure the last statement is an implicit **deny any** or **deny ip any any**.
- Remember that statement order is important because ACLs are processed top-down.
- As soon as a statement is matched the ACL is exited.
- Ensure that the most specific statements are at the top of the list.
- Remember that only one ACL is allowed per interface, per protocol, per direction.
- Remember that new statements for an existing ACL are added to the bottom of the ACL by default.
- Remember that router-generated packets are not filtered by outbound ACLs.
- Place standard ACLs as close to the destination as possible.
- Place extended ACLs as close to the source as possible.

Apply an ACL

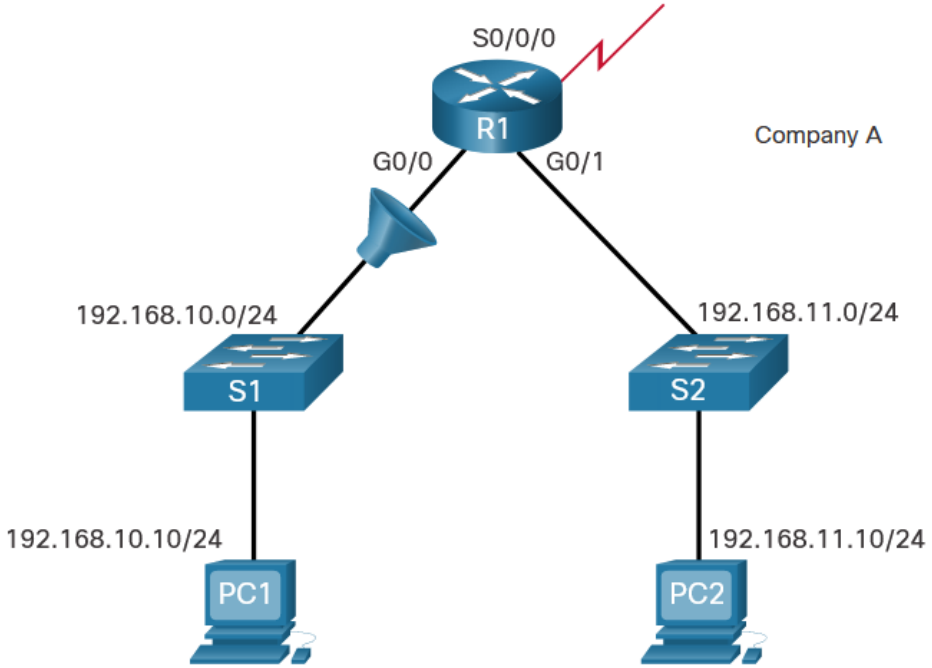
After creating an ACL, the administrator can apply it in a number of different ways. The following shows the command syntax to apply an ACL to an interface or to the vty lines.

```
Router(config-if)# ip access-group {acl-# | name} {in | out}
```

```
Router(config-line)# ip access-class {acl-# | name} {in | out}
```

Apply an ACL (Cont.)

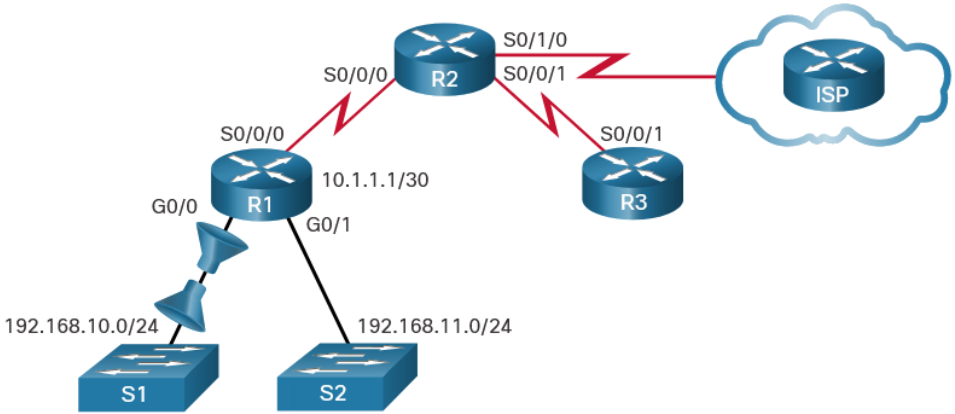
The figure below shows a named standard ACL applied to outbound traffic.




```
R1(config)# ip access-list standard NO_ACCESS
R1(config-std-nacl)# deny host 192.168.11.10
R1(config-std-nacl)# permit any
R1(config-std-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group NO_ACCESS out
```

Apply an ACL (Cont.)

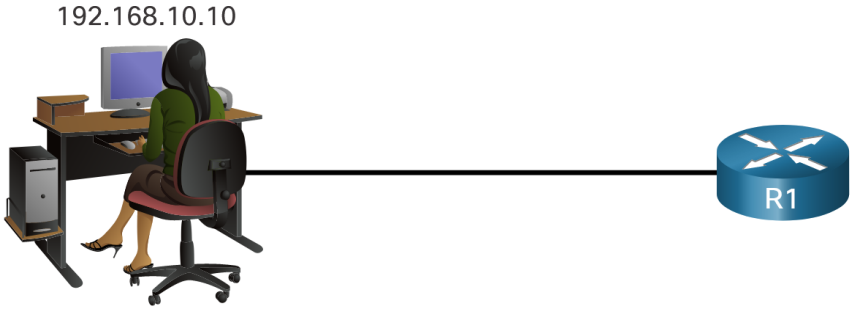
This figure shows two named extended ACLs. The SURFING ACL is applied to inbound traffic and the BROWSING ACL is applied to outbound traffic.




```
R1(config)# ip access-list extended SURFING
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 80
R1(config-ext-nacl)# permit tcp 192.168.10.0 0.0.0.255 any eq 443
R1(config-ext-nacl)# exit
R1(config)# ip access-list extended BROWSING
R1(config-ext-nacl)# permit tcp any 192.168.10.0 0.0.0.255 established
R1(config-ext-nacl)# exit
R1(config)# interface g0/0
R1(config-if)# ip access-group SURFING in
R1(config-if)# ip access-group BROWSING out
```

Apply an ACL (Cont.)

This example shows an ACL applied to the vty lines.

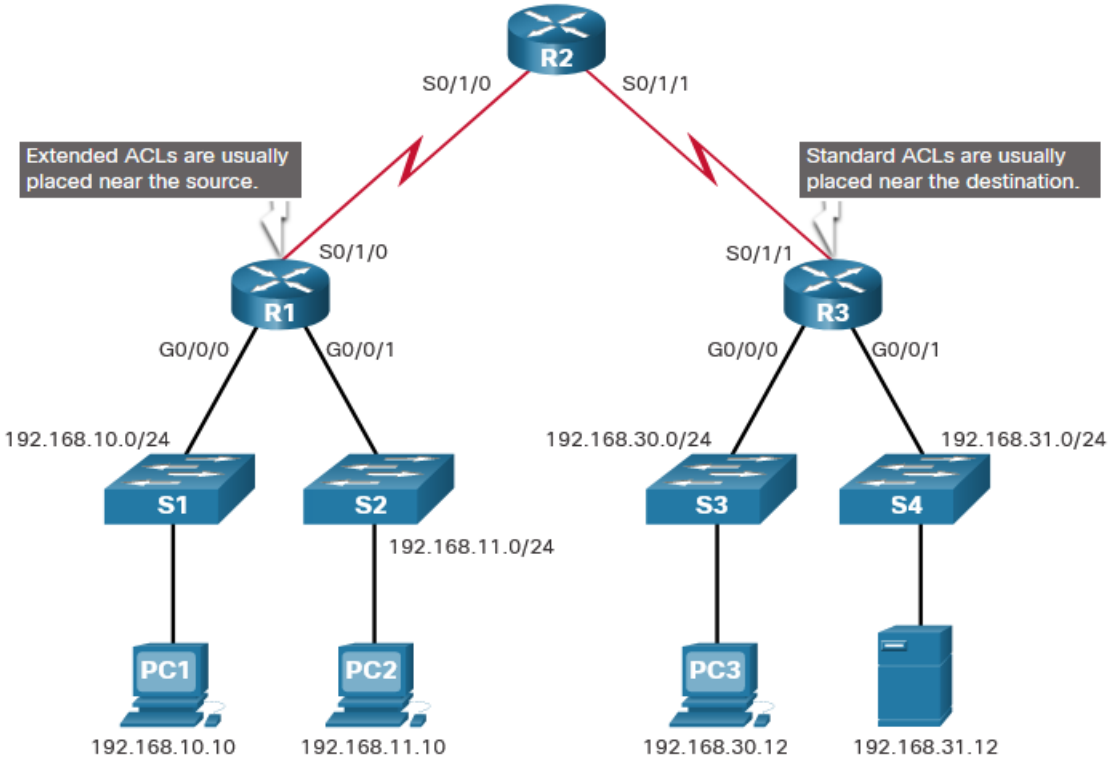



```
R1(config)# ip access-list standard VTY_ACCESS
R1(config-std-nacl)# permit 192.168.10.10 log
R1(config-std-nacl)# deny any
R1(config-std-nacl)# exit
R1(config)# line vty 0 4
R1(config-line)# access-class VTY_ACCESS in
R1(config-line)# end
R1#
R1# !The administrator accesses the vty lines from 192.168.10.10
R1#
*Feb 26 18:58:30.579: %SEC-6-IPACCESSLOGNP: list VTY_ACCESS permitted 0
192.168.10.10 -> 0.0.0.0, 5 packets
R1# show access-lists
Standard IP access list VTY_ACCESS
    10 permit 192.168.10.10 log (6 matches)
    20 deny any
```

Where to Place ACLs

Every ACL should be placed where it is the most efficient.

The figure illustrates where standard and extended ACLs should be located in an enterprise network. Assume the objective is to prevent traffic that originates in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.



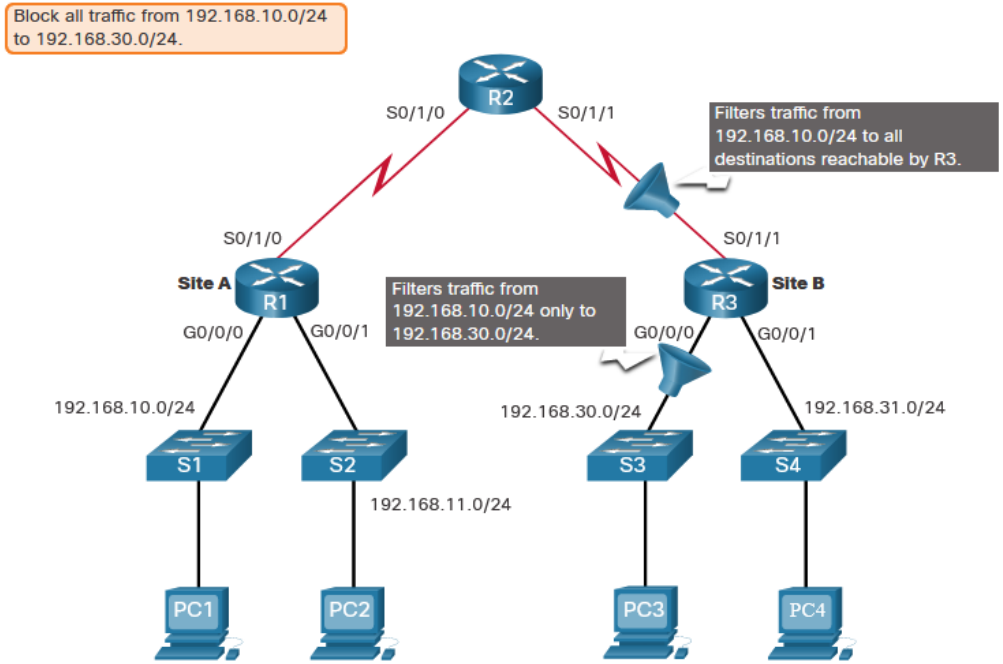
Where to Place ACLs (Cont.)

Placement of the ACL and therefore, the type of ACL used, may also depend on a variety of factors as listed in the table.

Factors Influencing ACL Placement	Explanation
The extent of organizational control	Placement of the ACL can depend on whether or not the organization has control of both the source and destination networks.
Bandwidth of the networks involved	It may be desirable to filter unwanted traffic at the source to prevent transmission of bandwidth-consuming traffic.
Ease of configuration	• It may be easier to implement an ACL at the destination, but traffic will use bandwidth unnecessarily. • An extended ACL could be used on each router where the traffic originated. This would save bandwidth by filtering the traffic at the source, but it would require creating extended ACLs on multiple routers.

Standard ACL Placement Example

Following the guidelines for ACL placement, standard ACLs should be located as close to the destination as possible. In the figure, the administrator wants to prevent traffic originating in the 192.168.10.0/24 network from reaching the 192.168.30.0/24 network.



Packet Tracer - Configure Named Standard IPv4 ACLs

The senior network administrator has asked you to create a named standard ACL to prevent access to a file server. All clients from one network and one specific workstation from a different network should be denied access.

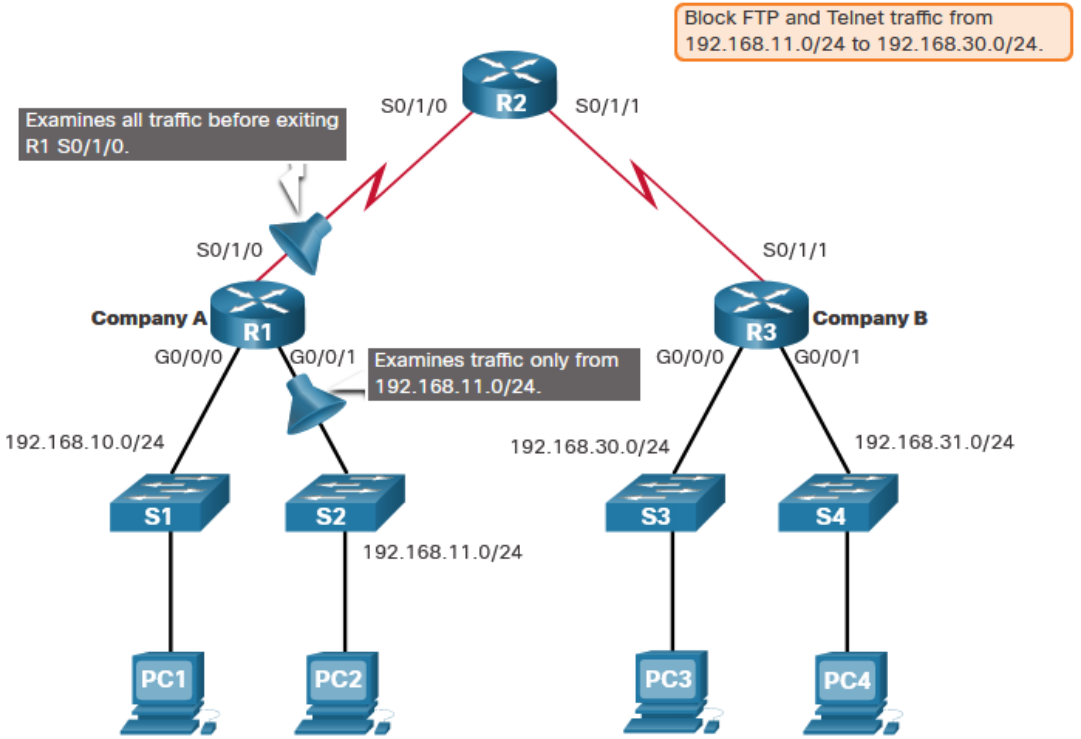
Packet Tracer - Configure Numbered Standard IPv4 ACLs

Standard access control lists 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.

Extended ACL Placement Example

Extended ACLs should be located as close to the source as possible. This prevents unwanted traffic from being sent across multiple networks only to be denied when it reaches its destination. However, the organization can only place ACLs on devices that they control. Therefore, the extended ACL placement must be determined in the context of where organizational control extends.

Company A wants to deny Telnet and FTP traffic to Company B's 192.168.30.0/24 network from their 192.168.11.0/24 network while permitting all other traffic.



Packet Tracer- Configuring Extended ACLs Scenario 1

In this Packet Tracer activity, you will complete the following objectives:

- Part 1: Configure, Apply, and Verify an Extended Numbered IPv4 ACL
- Part 2: Configure, Apply, and Verify an Extended Named IPv4 ACL

Packet Tracer - Configuring Extended ACLs Scenario 2

In this Packet Tracer activity, you will complete the following objectives:

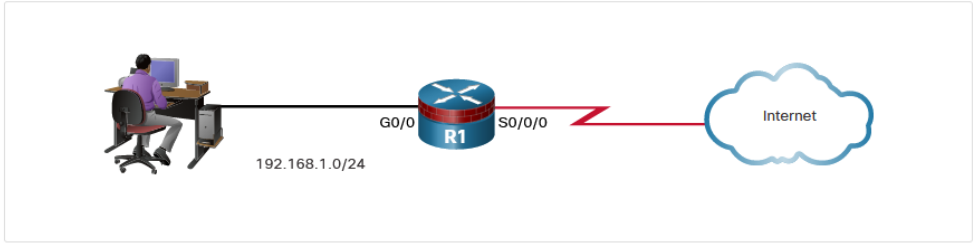
- Part 1: Configure a Named Extended IPv4 ACL
- Part 2: Apply and Verify the Extended IPv4 ACL

8.6 Mitigate Attacks with ACLs

Mitigate Spoofing Attacks

IP address spoofing overrides the normal packet creation process by inserting a custom IP header with a different source IP address. There are many well-known classes of IP addresses that should never be source IP addresses for traffic entering an organization's network. The S0/0/0 interface is attached to the internet and should never accept inbound packets from the following addresses:

- All zeros addresses
- Broadcast addresses
- Local host addresses (127.0.0.0/8)
- Automatic Private IP Addressing (APIPA) addresses (169.254.0.0/16)
- Reserved private addresses (RFC 1918)
- IP multicast address range (224.0.0.0/4)



Inbound on S0/0/0:

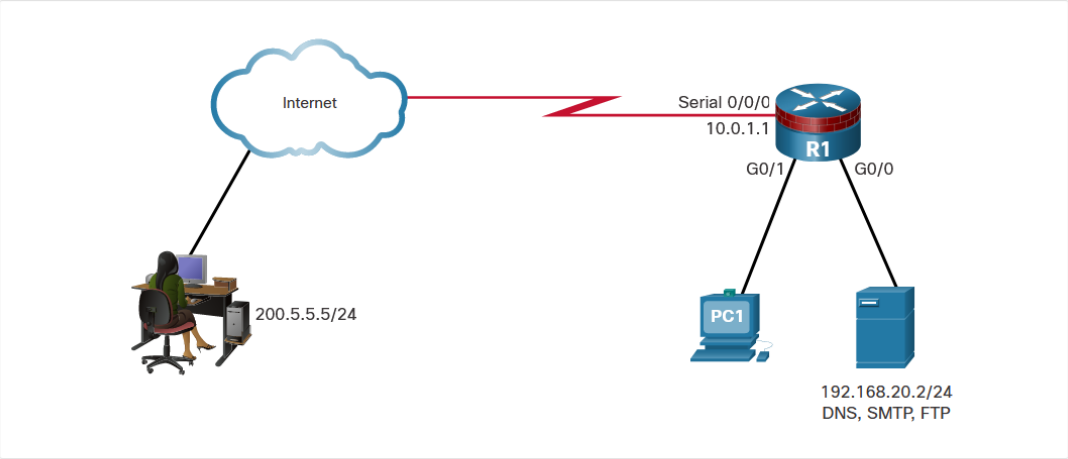
```
R1(config)# access-list 150 deny ip host 0.0.0.0 any
R1(config)# access-list 150 deny ip 10.0.0.0 0.255.255.255 any
R1(config)# access-list 150 deny ip 127.0.0.0 0.255.255.255 any
R1(config)# access-list 150 deny ip 172.16.0.0 0.15.255.255 any
R1(config)# access-list 150 deny ip 192.168.0.0 0.0.255.255 any
R1(config)# access-list 150 deny ip 224.0.0.0 15.255.255.255 any
R1(config)# access-list 150 deny ip host 255.255.255.255 any
```

Inbound on G0/0:

```
R1(config)# access-list 105 permit ip 192.168.1.0 0.0.0.255 any
```

Permit Necessary Traffic through a Firewall

An effective strategy for mitigating attacks is to explicitly permit only certain types of traffic through a firewall. For example, Domain Name System (DNS), Simple Mail Transfer Protocol (SMTP), and File Transfer Protocol (FTP) are services that often must be allowed through a firewall. Secure Shell (SSH), syslog, and Simple Network Management Protocol (SNMP) are examples of services that a router may need to include. The figure shows an example topology with ACL configurations to permit specific services on the Serial 0/0/0 interface.



Inbound on Serial 0/0/0

```
R1(config)# access-list 180 permit udp any host 192.168.20.2 eq domain
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq smtp
R1(config)# access-list 180 permit tcp any host 192.168.20.2 eq ftp
R1(config)# access-list 180 permit tcp host 200.5.5.5 host 10.0.1.1 eq 22
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq syslog
R1(config)# access-list 180 permit udp host 200.5.5.5 host 10.0.1.1 eq snmptrap
```

Mitigate ICMP Attacks

Both ICMP echo and redirect messages should be blocked inbound by the router. Several ICMP messages are recommended for proper network operation and should be allowed into the internal network:

- **Echo reply** - Allows users to ping external hosts.
- **Source quench** - Requests that the sender decrease the traffic rate of messages.
- **Unreachable** - Generated for packets that are administratively denied by an ACL.

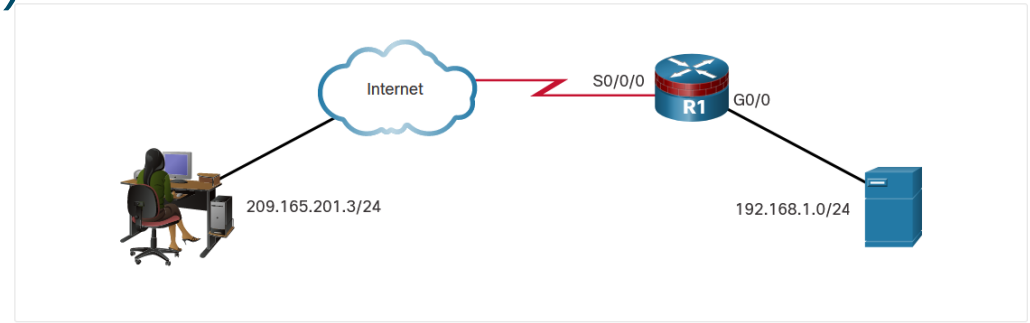
Several ICMP messages are required for proper network operation and should be allowed to exit the network:

- **Echo** - Allows users to ping external hosts.
- **Parameter problem** - Informs the host of packet header problems.
- **Packet too big** - Enables packet maximum transmission unit (MTU) discovery.
- **Source quench** - Throttles down traffic when necessary.

As a rule, block all other ICMP message types outbound.

Mitigate ICMP Attacks (Cont.)

The example shows a sample topology and possible ACL configurations to permit specific ICMP services on the G0/0 and S0/0/0 interfaces.



Inbound on S0/0/0:

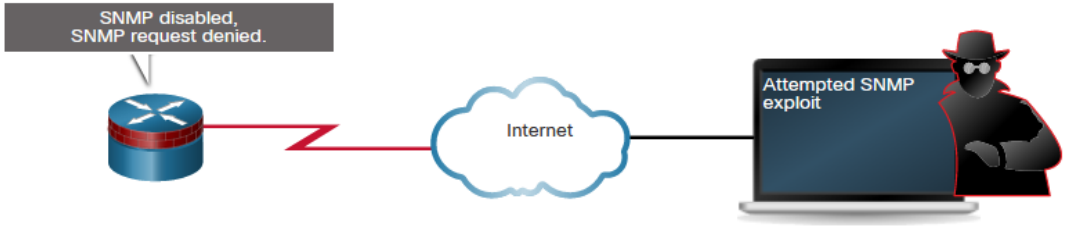
```
R1(config)# access-list 112 permit icmp any any echo-reply
R1(config)# access-list 112 permit icmp any any source-quench
R1(config)# access-list 112 permit icmp any any unreachable
R1(config)# access-list 112 deny icmp any any
R1(config)# access-list 112 permit ip any any
```

Inbound on G0/0:

```
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any echo
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any parameter-problem
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any packet-too-big
R1(config)# access-list 114 permit icmp 192.168.1.0 0.0.0.255 any source-quench
R1(config)# access-list 114 deny icmp any any
R1(config)# permit ip any any
```

Mitigate SNMP Attacks

Exploitation of SNMP vulnerabilities can be mitigated by applying interface ACLs to filter SNMP packets from non-authorized systems. These security measures are helpful, but the most effective means of exploitation prevention is to disable the SNMP server on IOS devices for which it is not required. Use the command **no snmp-server** to disable SNMP services on Cisco IOS devices.




```
Router(config)# no snmp-server
```

Packet Tracer - Configure IP ACLs to Mitigate Attacks

In this Packet Tracer, you will complete the following objectives:

- Verify connectivity among devices before firewall configuration.
- Use ACLs to ensure remote access to the routers is available from only management station PC-C.
- Configure ACLs on R1 and R3 to mitigate attacks.
- Verify ACL functionality.

8.8 Access Control Lists Summary

What Did I Learn in this Module?

- An ACL uses a sequential list of permit or deny statements, known as ACEs.
- The packet filtering process occurs when network traffic passes through an interface configured with an ACL.
- Packet filtering can occur at Layer 3 or Layer 4.
- Named ACLs are the preferred method to use when configuring ACLs.
- An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match.
- Unlike a subnet mask, in which binary 1 is equal to a match and binary 0 is not a match, in a wildcard mask, the reverse is true.
- Subtract the subnet mask from 255.255.255.255 to calculate the wildcard mask.
- Two keywords, **host** and **any**, can be used to simplify the most common uses of wildcard masking.
- Use a text editor to configure more complex ACLs, and then copy and paste the commands onto the device.

What Did I Learn in this Module?

- To create a numbered standard ACL, use the command **access-list** *access-list-number* {**deny** | **permit** | **remark** *text*} *source* [*source-wildcard*] [**log**].
- To create a named standard ACL, use the command **ip access-list standard** *access-list-name*.
- To apply a standard or extended IPv4 ACL to an interface use the command **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}.
- ACLs with multiple ACEs should be created in a text editor.
- An ACL ACE can also be deleted or added using the ACL sequence numbers.
- Extended ACLs should be located as close as possible to the source of the traffic to be filtered.
- Standard ACLs should be located as close to the destination as possible.
- Explicitly permit only certain types of traffic through a firewall.
- Both ICMP echo and redirect messages should be blocked inbound by the router. Apply interface ACLs to filter SNMP packets from non-authorized systems.
- Several ICMP messages are recommended for proper network operation and should be allowed into the internal network including echo reply, source quench, and unreachable.

What Did I Learn in this Module?

- Several ICMP messages should be allowed to exit the network including echo, parameter problem, packet too big, and source quench. As a rule, block all other ICMP message types outbound.
- Attackers can accomplish stealth attacks that result in trust exploitation by using dual-stacked hosts, rogue NDP messages, and tunneling techniques.
- To mitigate attacks against IPv6 infrastructures and protocols, the strategy should include filtering at the edge using various techniques, such as IPv6 ACLs.
- IPv6 ACLs allow filtering based on source and destination addresses that are traveling inbound and outbound to a specific interface.
- They also support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control, similar to extended ACLs in IPv4.

New Terms and Commands

- access control list (ACL)
- access control entry (ACE)
- packet filtering
- wildcard mask
- ANDing
- **access-list** *access-list-number* {**deny** | **permit** | **remark** *text*} *protocol source source-wildcard* [*operator* {*port*}] *destination destination-wildcard* [*operator* {*port*}] [**established**] [**log**]
- **ip access-list** {**standard** | **extended**} *name*
- **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
- **access-class** {*access-list-number* | *access-list-name*} {**in** | **out**}
- **show access-list**
- **ipv6 access-list** *access-list-name*
- **deny** | **permit** *protocol* {*source-ipv6-prefix* **/** *prefix-length* | **any** | **host** *source-ipv6-address*} [*operator* [*port-number*]] { *destination-ipv6-prefix* **/** *prefix-length* | **any** | **host** *destination-ipv6-address* } [*operator* [*port-number*]] [**dscp** *value*] [**fragments**] [**log**] [**log-input**] [**sequence** *value*] [**time-range** *name*]
- **show ipv6 access-list**


CISCO