# CCNA 3 v7.0 Curriculum: Module 4 – ACL Concepts

itexamanswers.net/ccna-3-v7-0-curriculum-module-4-acl-concepts.html

April 8, 2020

## 4.0. Introduction

### 4.0.1. Why should I take this module?

Welcome to ACL Concepts!

You have arrived at your grandparents' residence. It is a beautiful gated community with walking paths and gardens. For the residents safety, no one is permitted to get into the community without stopping at the gate and presenting the guard with identification. You provide your ID and the guard verifies that you are expected as a visitor. He documents your information and lifts the gate. Imagine if the guard had to do this for the many staff members that entered each day. They have simplified this process by assigning a badge for each employee to automatically raise the gate once the badge is scanned. You greet your grandparents who are anxiously awaiting you at the front desk. You all get back into the car to go down the street for dinner. As you exit the parking lot, you must again stop and show your identification so that the guard will lift the gate. Rules have been put in place for all incoming and outgoing traffic.

Much like the guard in the gated community, network traffic passing through an interface configured with an access control list (ACL) has permitted and denied traffic. The router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering. Let's learn more!

### 4.0.2.What will I learn to do in this module?

**Module Title:** ACL Concepts

**Module Objective:** Explain how ACLs are used as part of a network security policy.

| Topic Title | Topic Objective |
|---|---|
| **Purpose of ACLs** | Explain how ACLs filter traffic. |
| **Wildcard Masks in ACLs** | Explain how ACLs use wildcard masks. |
| **Guidelines for ACL Creation** | Explain how to create ACLs. |
| **Types of IPv4 ACLs** | Compare standard and extended IPv4 ACLs. |

## 4.1. Purpose of ACLs

### 4.1.1. What is an ACL?

Routers make routing decisions based on information in the packet header. Traffic entering a router interface is routed solely based on information within the routing table. The router compares the destination IP address with routes in the routing table to find the best match and then forwards the packet based on the best match route. That same process can be used to filter traffic using an access control list (ACL).

An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. By default, a router does not have any ACLs configured. However, when an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded.

An ACL uses a sequential list of permit or deny statements, known as access control entries (ACEs).

**Note:** ACEs are also commonly called ACL statements.

When network traffic passes through an interface configured with an ACL, the router compares the information within the packet against each ACE, in sequential order, to determine if the packet matches one of the ACEs. This process is called packet filtering.

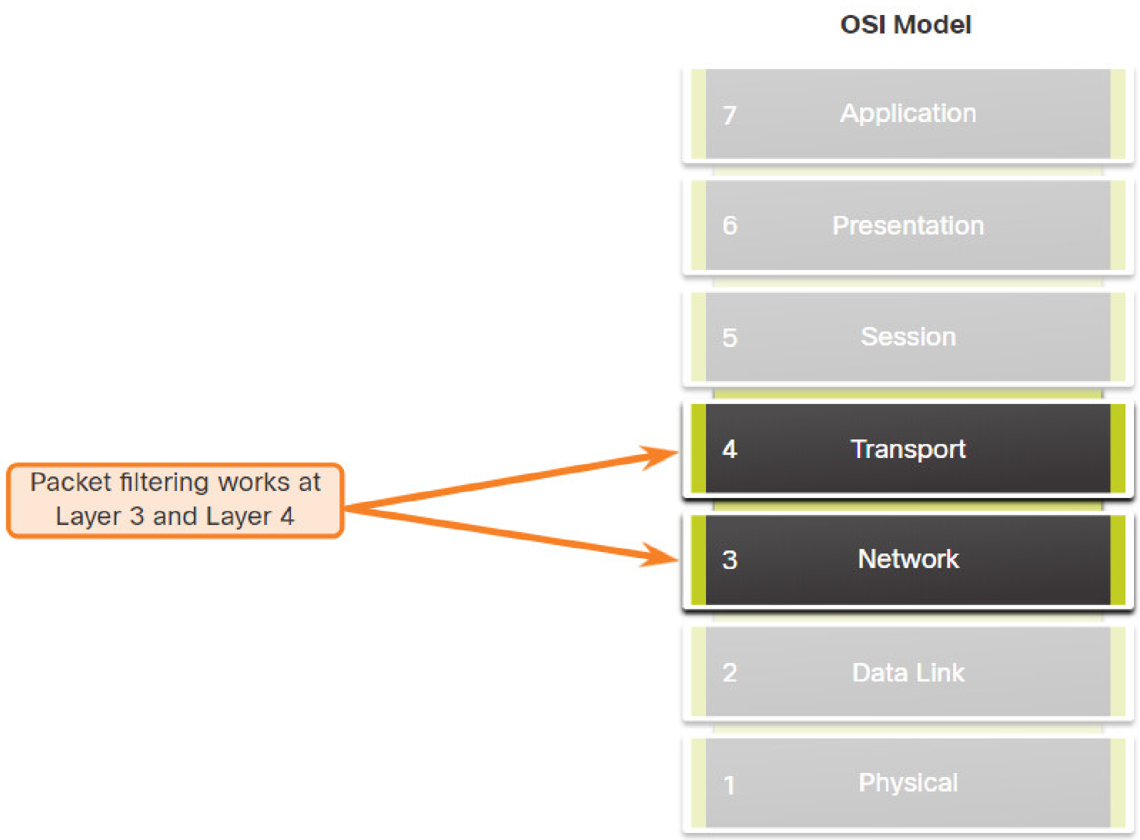
Several tasks performed by routers require the use of ACLs to identify traffic. The table lists some of these tasks with examples.

| Task | Example |
|---|---|
| Limit network traffic to increase network performance | • A corporate policy prohibits video traffic on the network to reduce the network load.<br>• 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.<br>• A policy can be implemented using ACLs to restrict the delivery of routing updates to only those that come from a known source. |

| Task | Example |
|---|---|
| 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.<br>• 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.<br>• 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.<br>• 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.<br>• A policy can be implemented using ACLs and QoS services to identify voice traffic and process it immediately. |

### 4.1.2. 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, as shown in the figure.



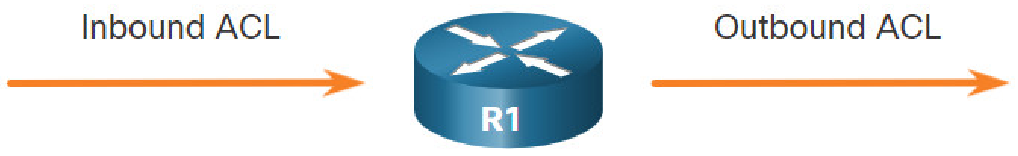
Cisco routers support two types of ACLs:

- **Standard ACLs** – ACLs only filter at Layer 3 using the source IPv4 address only.
- **Extended ACLs** – ACLs filter at Layer 3 using the source and / or destination IPv4 address. They can also filter at Layer 4 using TCP, UDP ports, and optional protocol type information for finer control.

## 4.1.3. 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.

ACLs can be configured to apply to inbound traffic and outbound traffic, as shown in the figure.



**Note**: ACLs do not act on packets that originate from the router itself.

An inbound ACL filters packets before they are routed to the outbound interface. An inbound ACL is efficient because it saves the overhead of routing lookups if the packet is discarded. 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.

When an ACL is applied to an interface, it follows a specific operating procedure. For example, here are the operational steps used when traffic has entered a router interface with an inbound standard IPv4 ACL configured.

1. The router extracts the source IPv4 address from the packet header.
2. The router starts at the top of the ACL and compares the source IPv4 address to each ACE in a sequential order.
3. When a match is made, the router carries out the instruction, either permitting or denying the packet, and the remaining ACEs in the ACL, if any, are not analyzed.
4. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded because there is an implicit deny ACE automatically applied to all ACLs.

The last ACE statement of an ACL is always an implicit deny that blocks all traffic. By default, this statement is automatically implied at the end of an ACL even though it is hidden and not displayed in the configuration.

**Note**: An ACL must have at least one permit statement otherwise all traffic will be denied due to the implicit deny ACE statement.

### 4.1.4. Packet Tracer – ACL Demonstration

In this activity, you will observe how an access control list (ACL) can be used to prevent a ping from reaching hosts on remote networks. After removing the ACL from the configuration, the pings will be successful.

**4.1.4 Packet Tracer – ACL Demonstration**

## 4.2. Wildcard Masks in ACLs

### 4.2.1. Wildcard Mask Overview

In the previous topic, you learned about the purpose of ACL. This topic explains how ACL uses wildcard masks. An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match. Wildcard masks are also used by the Open Shortest Path First (OSPF) routing protocol.

A wildcard mask is similar to 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 masks use the following rules to match binary 1s and 0s:

- **Wildcard mask bit 0** – Match the corresponding bit value in the address
- **Wildcard mask bit 1** – Ignore the corresponding bit value in the address

The table lists some examples of wildcard masks and what they would identify.

| 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<br>• Match the two left most bits of the last octet<br>• Ignore the last 6 bits |
| `0.0.0.15` | `00001111` | • Match the first three octets<br>• Match the four left most bits of the last octet<br>• Ignore the last 4 bits of the last octet |
| `0.0.0.248` | `11111100` | • Match the first three octets<br>• Ignore the six left most bits of the last octet<br>• Match the last two bits |
| `0.0.0.255` | `11111111` | • Match the first three octet<br>• Ignore the last octet |

## 4.2.2. Wildcard Mask Types

Using wildcard masks will take some practice. Refer to the examples to learn how the wildcard mask is used to filter traffic for one host, one subnet, and a range IPv4 addresses.

Click each button to see how the wildcard mask is used in ACLs.

- Wildcard to Match a Host
- Wildcard Mask to Match an IPv4 Subnet
- Wildcard Mask to Match an IPv4 Address Range

### Wildcard to Match a Host

In this example, the wildcard mask is used to match a specific host IPv4 address. Assume ACL 10 needs an ACE that only permits the host with IPv4 address 192.168.1.1. Recall that "0" equals a match and "1" equals ignore. To match a specific host IPv4 address, a wildcard mask consisting of all zeroes (i.e., 0.0.0.0) is required.

The table lists in binary, the host IPv4 address, the wildcard mask, and the permitted IPv4 address.

The 0.0.0.0 wildcard mask stipulates that every bit must match exactly. Therefore, when the ACE is processed, the wildcard mask will permit only the 192.168.1.1 address. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.1 0.0.0.0.**

| | 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

In this example, ACL 10 needs an ACE that permits all hosts in the 192.168.1.0/24 network. The wildcard mask 0.0.0.255 stipulates that the very first three octets must match exactly but the fourth octet does not.

The table lists in binary, the host IPv4 address, the wildcard mask, and the permitted IPv4 addresses.

When processed, the wildcard mask 0.0.0.255 permits all hosts in the 192.168.1.0/24 network. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.1.0 0.0.0.255.**

| | Decimal | Binary |
|---|---|---|
| IPv4 address | 192.168.1.1 | `11000000.10101000.00000001.00000001` |
| Wildcard Mask | 0.0.0.255 | `00000000.00000000.00000000.11111111` |

| | Decimal | Binary |
|---|---|---|
| **Permitted IPv4 Address** | **192.168.1.0/24** | **`11000000.10101000.00000001.00000000`** |

**Wildcard Mask to Match an IPv4 Address Range**

In this example, ACL 10 needs an ACE that permits all hosts in the 192.168.16.0/24, 192.168.17.0/24, …, 192.168.31.0/24 networks. The wildcard mask 0.0.15.255 would correctly filter that range of addresses.

The table lists in binary the host IPv4 address, the wildcard mask, and the permitted IPv4 addresses.

The highlighted wildcard mask bits identify which bits of the IPv4 address must match. When processed, the wildcard mask 0.0.15.255 permits all hosts in the 192.168.16.0/24 to 192.168.31.0/24 networks. The resulting ACE in ACL 10 would be **access-list 10 permit 192.168.16.0 0.0.15.255.**

| | 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`** |

## 4.2.3. Wildcard Mask Calculation

Calculating wildcard masks can be challenging. One shortcut method is to subtract the subnet mask from 255.255.255.255. Refer to the examples to learn how to calculate the wildcard mask using the subnet mask.

Click each button to see how to calculate each wildcard mask.

- Example 1
- Example 2
- Example 3
- Example 4

**Example 1**

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.

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.**

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

**Example 2**

In this example, assume you wanted an ACE in ACL 10 to permit network access for the 14 users in the subnet 192.168.3.32/28. Subtract the subnet (i.e., 255.255.255.240) from 255.255.255.255, as shown in the table.

This solution produces the wildcard mask 0.0.0.15. Therefore, the ACE would be **access-list 10 permit 192.168.3.32 0.0.0.15.**

| | |
|---|---|
| Starting value | `255.255.255.255` |
| Subtract the subnet mask | `- 255.255.255.240` |
| **Resulting wildcard mask** | `0. 0. 0. 15` |

**Example 3**

In this example, assume you needed an ACE in ACL 10 to permit only networks 192.168.10.0 and 192.168.11.0. These two networks could be summarized as 192.168.10.0/23 which is a subnet mask of 255.255.254.0. Again, you subtract 255.255.254.0 subnet mask from 255.255.255.255, as shown in the table.

This solution produces the wildcard mask 0.0.1.255. Therefore, the ACE would be **access-list 10 permit 192.168.10.0 0.0.1.255.**

| | |
|---|---|
| Starting value | `255.255.255.255` |
| Subtract the subnet mask | `- 255.255.254. 0` |
| **Resulting wildcard mask** | `0. 0. 1.255` |

## 4.2.4. Wildcard Mask Keywords

Working with decimal representations of binary wildcard mask bits can be tedious. To simplify this task, the Cisco IOS provides two keywords to identify the most common uses of wildcard masking. Keywords reduce ACL keystrokes and make it easier to read the ACE.

The two keywords are:

- **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, in the command output, two ACLs are configured. The ACL 10 ACE permits only the 192.168.10.10 host and the ACL 11 ACE permits all hosts.

```
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)#
```

Alternatively, the keywords **host** and **any** could have been used to replace the highlighted output.

The following commands accomplishes the same task as the previous commands.

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

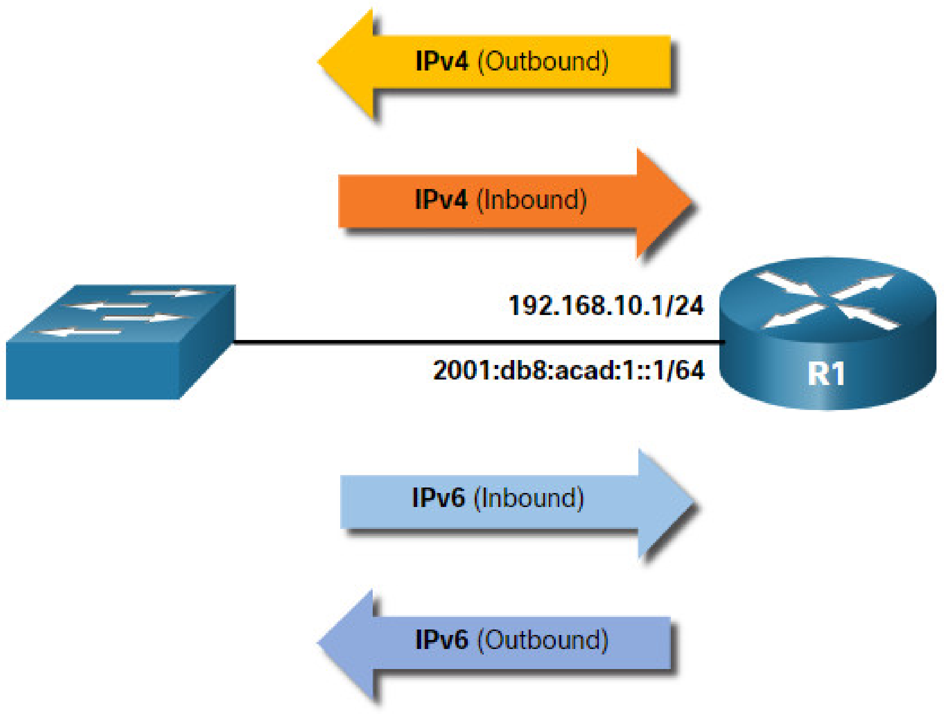
## 4.3. Guidelines for ACL Creation

### 4.3.1. Limited Number of ACLs per Interface

In a previous topic, you learned about how wildcard masks are used in ACLs. This topic will focus on the guidelines for ACL creation. There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e, IPv4 and IPv6) router interface can have up to four ACLs applied, as shown in the figure.

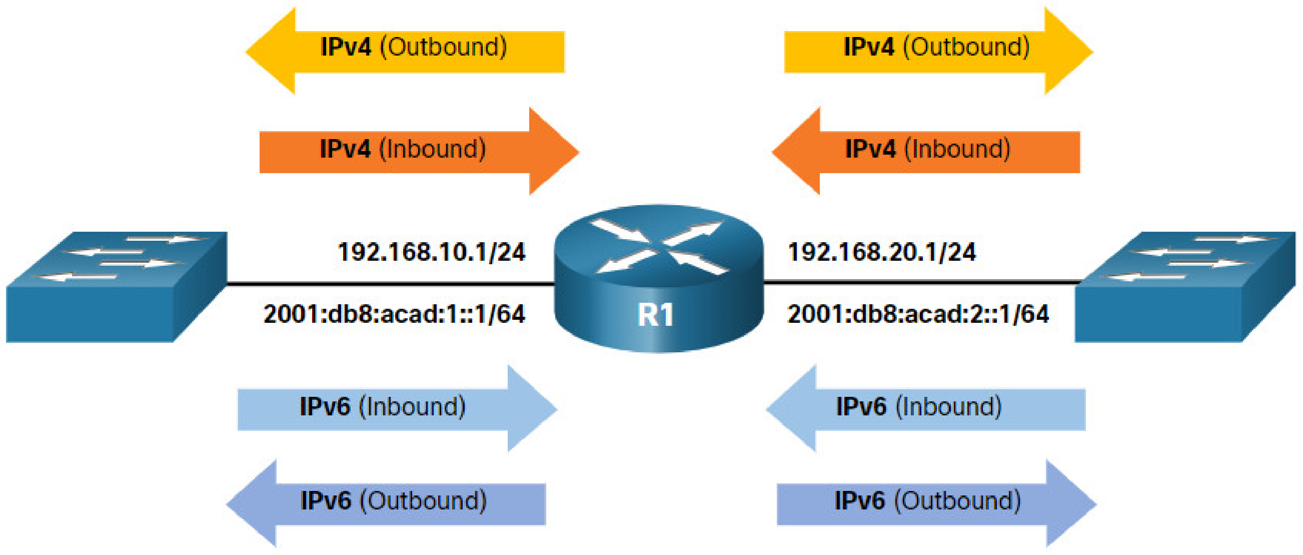
Specifically, a router interface can have:

- one outbound IPv4 ACL
- one inbound IPv4 ACL
- one inbound IPv6 ACL
- one outbound IPv6 ACL



Assume R1 has two dual-stacked interfaces that require inbound and outbound IPv4 and IPv6 ACLs applied. As shown in the figure, R1 could have up to 8 ACLs configured and applied to interfaces. Each interface would have four ACLs; two ACLs for IPv4 and two ACLs for IPv6. For each protocol, one ACL is for inbound traffic and one for outbound traffic.

**Note**: ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization.



## 4.3.2. ACL Best Practices

Using ACLs requires attention to detail and great care. Mistakes can be costly in terms of downtime, troubleshooting efforts, and poor network service. Basic planning is required before configuring an ACL.

The table presents guidelines that form the basis of an ACL best practices list.

| Guideline | Benefit |
|---|---|
| Base ACLs on the organizational security policies. | This will ensure you implement organizational security guidelines. |
| Write out what you want the ACL to do. | This will help you avoid inadvertently creating potential access problems. |
| Use a text editor to create, edit, and save all of your ACLs. | This will help you create a library of reusable ACLs. |
| Document the ACLs using the **remark** command. | This will help you (and others) understand the purpose of an ACE. |
| Test the ACLs on a development network before implementing them on a production network. | This will help you avoid costly errors. |

## 4.4. Types of IPv4 ACLs

### 4.4.1. Standard and Extended ACLs

The previous topics covered the purpose of ACL and the guidelines for ACL creation. This topic will cover standard and extended ACLs, named and numbered ACLs, and the examples of placement of these ACLs.

There are two types of IPv4 ACLs:

- **Standard ACLs** – These permit or deny packets based only on the source IPv4 address.
- **Extended ACLs** – These permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more.

For example, refer to the following standard ALC command.

```
R1(config)# access-list 10 permit 192.168.10.0 0.0.0.255
R1(config)#
```

ACL 10 permits hosts on the source network 192.168.10.0/24. Because of the implied “deny any” at the end, all traffic except for traffic coming from the 192.168.10.0/24 network is blocked with this ACL.

In the next example, an extended ACL 100 permits traffic originating from any host on the 192.168.10.0/24 network to any IPv4 network if the destination host port is 80 (HTTP).

```
R1(config)# access-list 100 permit tcp 192.168.10.0 0.0.0.255 any eq www
R1(config)#
```

Notice how the standard ACL 10 is only capable of filtering by source address while the extended ACL 100 is filtering on the source, and destination Layer 3, and Layer 4 protocol (i.e., TCP) information.

**Note:** Full ACL configuration is discussed in another module.

## 4.4.2. 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. For example, naming an extended ACL FTP-FILTER is far better than having a numbered ACL 100.

The **ip access-list** global configuration command is used to create a named ACL, as shown in the following example.

```
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)#
```

The following summarizes the rules to follow for named ACLs.

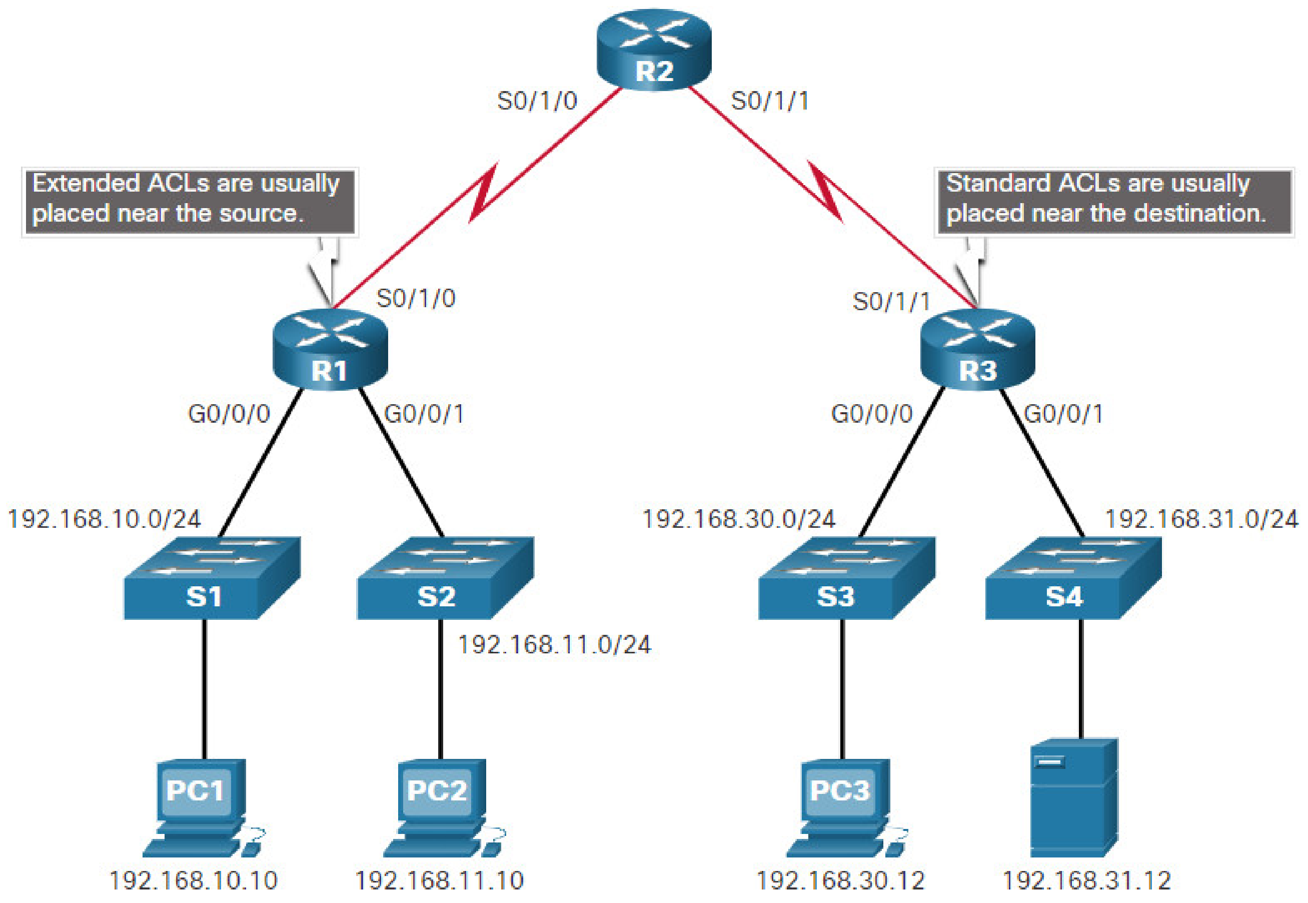
- Assign a name to identify the purpose of the ACL.

- Names can contain alphanumeric characters.
- Names cannot contain spaces or punctuation.
- It is suggested that the name be written in CAPITAL LETTERS.
- Entries can be added or deleted within the ACL.

### 4.4.3. Where to Place ACLs

Every ACL should be placed where it has the greatest impact on efficiency.

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



Extended ACLs should be located as close as possible to the source of the traffic to be filtered. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure.

Standard ACLs should be located as close to the destination as possible. If a standard ACL was placed at the source of the traffic, the "permit" or "deny" will occur based on the given source address no matter where the traffic is destined.

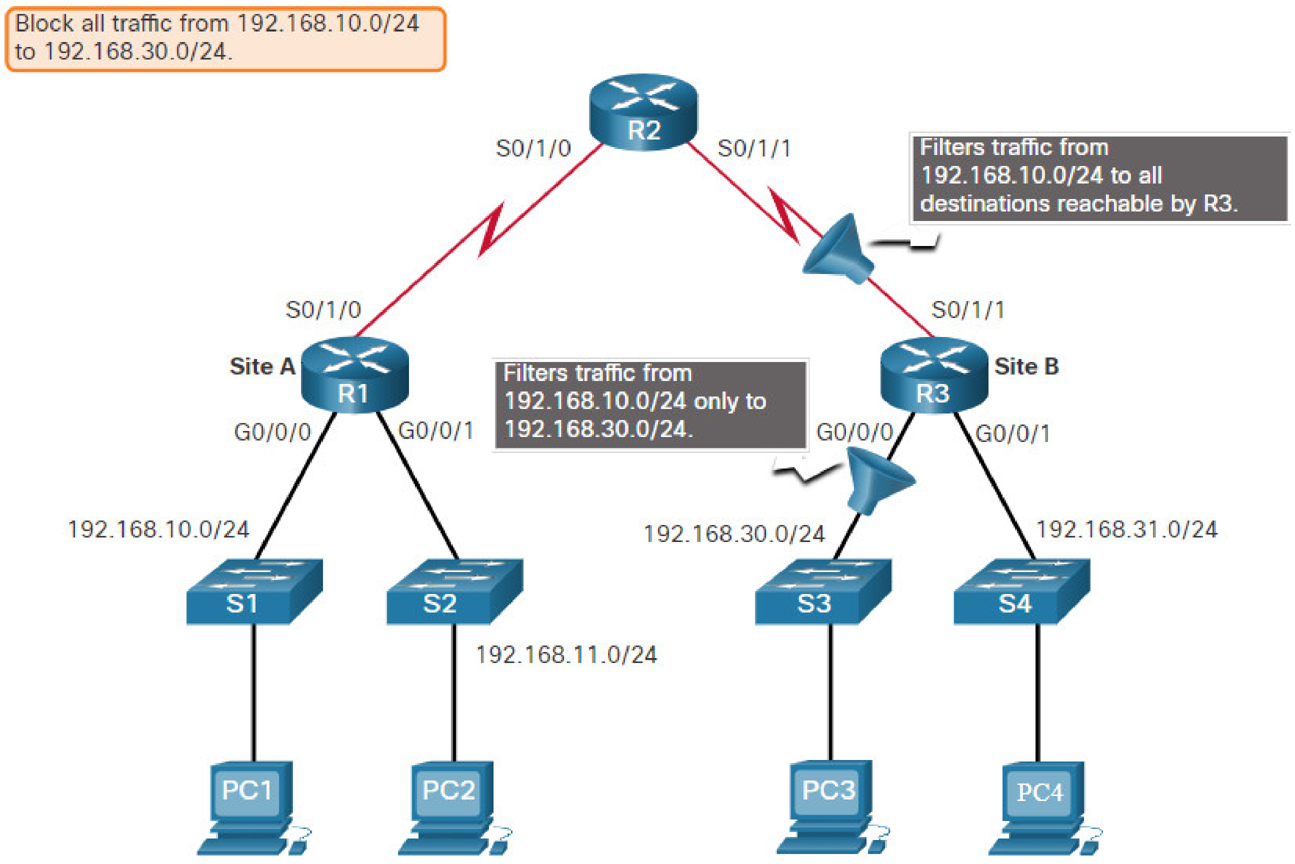
Placement of the ACL and therefore, the type of ACL used, may also depend on a variety of factors as listed in the table.

| 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.<br>• 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. |

## 4.4.4. 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.


Block all traffic from 192.168.10.0/24 to 192.168.30.0/24.
R2
S0/1/0
S0/1/1
Filters traffic from 192.168.10.0/24 to all destinations reachable by R3.
S0/1/0
S0/1/1
Site A
R1
G0/0/0
G0/0/1
Filters traffic from 192.168.10.0/24 only to 192.168.30.0/24.
R3
Site B
G0/0/0
G0/0/1
192.168.10.0/24
192.168.30.0/24
192.168.31.0/24
S1
S2
S3
S4
192.168.11.0/24
PC1
PC2
PC3
PC4

Following the basic placement guidelines, the administrator would place a standard ACL on router R3. There are two possible interfaces on R3 to apply the standard ACL:

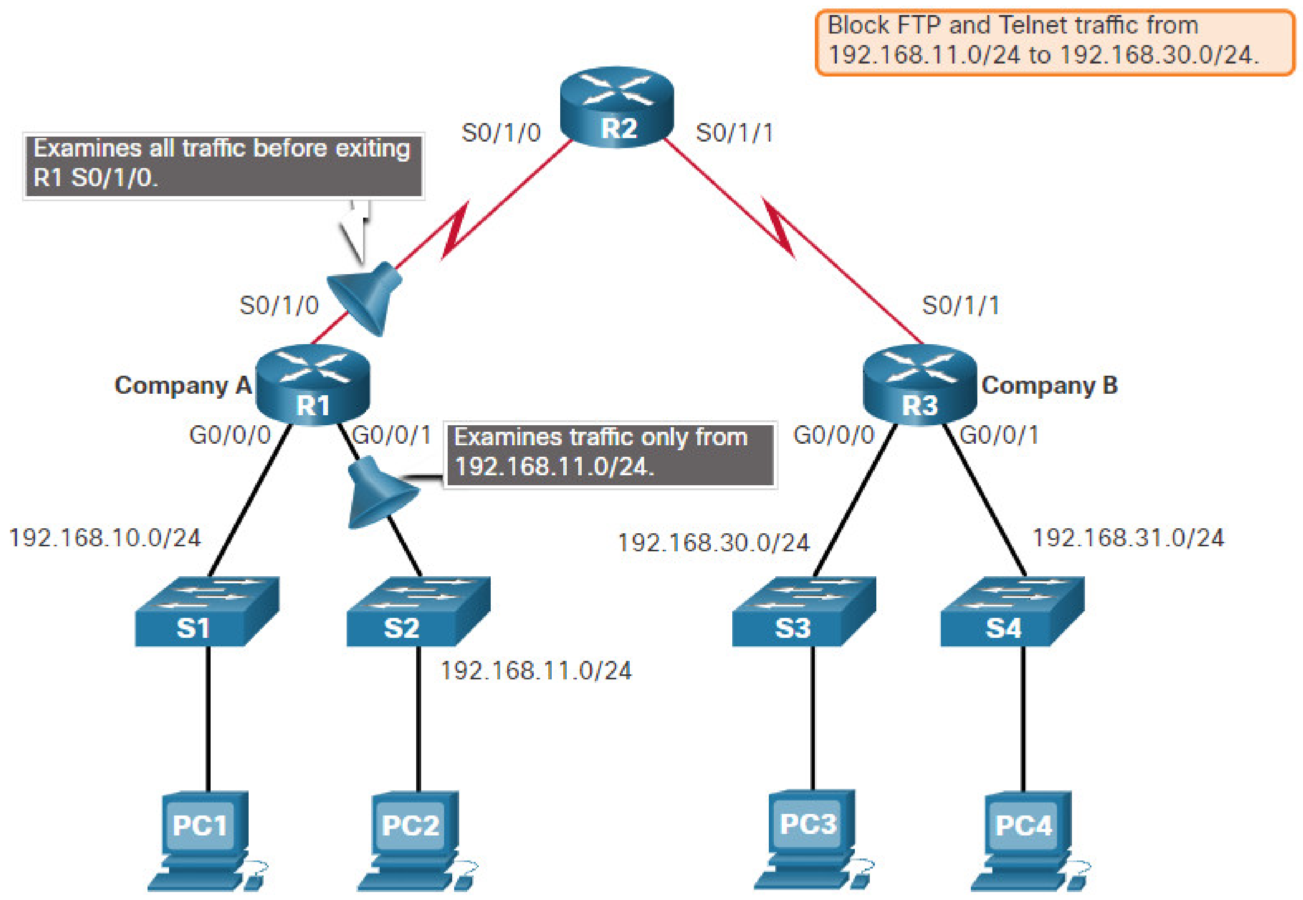
- **R3 S0/1/1 interface (inbound)** – The standard ACL can be applied inbound on the R3 S0/1/1 interface to deny traffic from .10 network. However, it would also filter .10 traffic to the 192.168.31.0/24 (.31 in this example) network. Therefore, the standard ACL should not be applied to this interface.
- **R3 G0/0 interface (outbound)** – The standard ACL can be applied outbound on the R3 G0/0/0 interface. This will not affect other networks that are reachable by R3. Packets from .10 network will still be able to reach the .31 network. This is the best interface to place the standard ACL to meet the traffic requirements.

## 4.4.5. Extended ACL Placement Example

Extended ACL 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.

In the figure, for example, 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.


Block FTP and Telnet traffic from 192.168.11.0/24 to 192.168.30.0/24.
Examines all traffic before exiting R1 S0/1/0.
S0/1/0
R2
S0/1/1
S0/1/0
S0/1/1
Company A
R1
G0/0/0
G0/0/1
Examines traffic only from 192.168.11.0/24.
Company B
R3
G0/0/0
G0/0/1
192.168.10.0/24
192.168.30.0/24
192.168.31.0/24
S1
S2
S3
S4
192.168.11.0/24
PC1
PC2
PC3
PC4

There are several ways to accomplish these goals. An extended ACL on R3 would accomplish the task, but the administrator does not control R3. In addition, this solution allows unwanted traffic to cross the entire network, only to be blocked at the destination. This affects overall network efficiency.

The solution is to place an extended ACL on R1 that specifies both source and destination addresses.

There are two possible interfaces on R1 to apply the extended ACL:

- **R1 S0/1/0 interface (outbound)** – The extended ACL can be applied outbound on the S0/1/0 interface. However, this solution will process all packets leaving R1 including packets from 192.168.10.0/24.
- **R1 G0/0/1 interface (inbound)** – The extended ACL can be applied inbound on the G0/0/1 and only packets from the 192.168.11.0/24 network are subject to ACL processing on R1. Because the filter is to be limited to only those packets leaving the 192.168.11.0/24 network, applying the extended ACL to G0/1 is the best solution.

## 4.5. Module Practice and Quiz

### 4.5.1. What did I learn in this module?

**Purpose of ACLs**

Several tasks performed by routers require the use of ACLs to identify traffic. An ACL is a series of IOS commands that are used to filter packets based on information found in the packet header. A router does not have any ACLs configured by default. However, when an ACL is applied to an interface, the router performs the additional task of evaluating all network packets as they pass through the interface to determine if the packet can be forwarded. An ACL uses a sequential list of permit or deny statements, known as ACEs. Cisco routers support two types of ACLs: standard ACLs and extended ACLs. 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. An outbound ACL filters packets after being routed, regardless of the inbound interface. When an ACL is applied to an interface, it follows a specific operating procedure:

1. The router extracts the source IPv4 address from the packet header.
2. The router starts at the top of the ACL and compares the source IPv4 address to each ACE in a sequential order.
3. When a match is made, the router carries out the instruction, either permitting or denying the packet, and the remaining ACEs in the ACL, if any, are not analyzed.
4. If the source IPv4 address does not match any ACEs in the ACL, the packet is discarded because there is an implicit deny ACE automatically applied to all ACLs.

**Wildcard Masks**

An IPv4 ACE uses a 32-bit wildcard mask to determine which bits of the address to examine for a match. Wildcard masks are also used by the Open Shortest Path First (OSPF) routing protocol. A wildcard mask is similar to 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. **Wildcard mask bit 0** matches the corresponding bit value in the address. **Wildcard mask bit 1** ignores the corresponding bit value in the address. A wildcard mask is used to filter traffic for one host, one subnet, and a range IPv4 addresses. A shortcut to calculating a wildcard mask is to subtract the subnet mask from 255.255.255.255. Working with decimal representations of binary wildcard mask bits can be simplified by using the Cisco IOS keywords **host** and **any** to identify the most common uses of wildcard masking. Keywords reduce ACL keystrokes and make it easier to read the ACE.

**Guidelines for ACL creation**

There is a limit on the number of ACLs that can be applied on a router interface. For example, a dual-stacked (i.e, IPv4 and IPv6) router interface can have up to four ACLs applied. Specifically, a router interface can have one outbound IPv4 ACL, one inbound IPv4 ACL, one inbound IPv6 ACL , and one outbound IPv6 ACL. ACLs do not have to be configured in both directions. The number of ACLs and their direction applied to the interface will depend on the security policy of the organization. Basic planning is required before configuring an ACL and includes the following best practices:

- Base ACLs on the organizational security policies.
- Write out what you want the ACL to do.
- Use a text editor to create, edit, and save all of your ACLs.
- Document the ACLs using the **remark** command.
- Test the ACLs on a development network before implementing them on a production network.

**Types of IPv4 ACLs**

There are two types of IPv4 ACLs: standard ACLs and Extended ACLs. Standard ACLs permit or deny packets based only on the source IPv4 address. Extended ACLs permit or deny packets based on the source IPv4 address and destination IPv4 address, protocol type, source and destination TCP or UDP ports and more. ACLs number 1 -to 99, or 1300 to 1999, are standard ACLs. ACLs number 100-199, or 2000 to 2699, are extended 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 following summarizes the rules to follow for named ACLs:

- Assign a name to identify the purpose of the ACL.
- Names can contain alphanumeric characters.
- Names cannot contain spaces or punctuation.
- It is suggested that the name be written in CAPITAL LETTERS.
- Entries can be added or deleted within the ACL.

Every ACL should be placed where it has the greatest impact on efficiency. Extended ACLs should be located as close as possible to the source of the traffic to be filtered. This way, undesirable traffic is denied close to the source network without crossing the network infrastructure. Standard ACLs should be located as close to the destination as possible. If a standard ACL was placed at the source of the traffic, the “permit” or “deny” will occur based on the given source address no matter where the traffic is destined. Placement of the ACL may depend on the extent of organizational control, bandwidth of the networks, and ease of configuration.

## 4.5.2 Module Quiz – ACL Concepts

## Download Slide Powerpoint (PPT)

CCNA 3 v7.0 Curriculum: Module 4 - ACL Concepts.pptx

1 file(s) 1.57 MB

Download

Tags:ccna 3 v7 modules