Chapter 9 - Access Control Lists

Objectives

- Creating an ACL
- Matching on Addresses: Wildcard Masks
- Activating an ACL
- Configuring Standard Numbered ACLs
- Extended Numbered ACLs
- Creating Named ACLs
- Implementing IPv6 ACLs

1. Introduction

Access Control List (ACL) are filters that enable you to control which routing updates or packets are permitted or denied in or out of a network.

They are specifically used by network administrators to filter traffic and to provide extra security for the network. This can be applied on routers (Cisco).

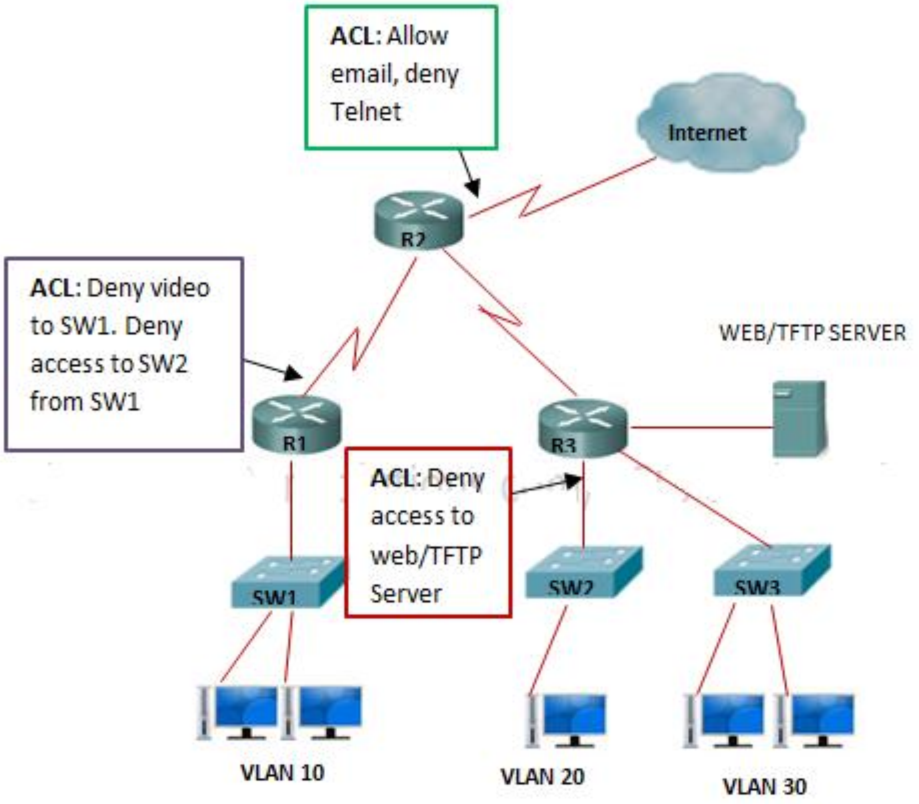
ACLs provide a powerful way to control traffic into and out of your network; this control can be as simple as permitting or denying network hosts or addresses. You can configure ACLs for all routed network protocols.

The most important reason to configure ACLs is to provide security for your network. However, ACLs can also be configured to control network traffic based on the TCP port being used.

How ACLs works

A router acts as a packet filter when it forwards or denies packets according to filtering rules. As a Layer 3 device, a packet-filtering router uses rules to determine whether to permit or deny traffic based on source and destination IP addresses, source port and destination port, and the protocol of the packet. These rules are defined using access control lists or ACLs.

To simplify how ACL or a router uses packet filtering work, imagine a guard stationed at a locked door. The guard's instruction is to allow only people whose names appear on a quest list to pass through the door. The guard is filtering people based on the condition of having their names on the authorized list.


ACL: Allow email, deny Telnet
Internet
R2
ACL: Deny video to SW1. Deny access to SW2 from SW1
WEB/TFTP SERVER
R1
R3
ACL: Deny access to web/TFTP Server
SW1
SW2
SW3
VLAN 10
VLAN 20
VLAN 30

When a packet arrives at the router, the router extracts certain information from the packet header and makes decisions according to the filter rules as to whether the packet can pass through or be dropped. Packet filtering process works at the Network layer of the Open Systems Interconnection (OSI) model, or the Internet layer of TCP/IP.

Why use ACLs

- Limits network traffic to increase network performance.

- ACLs provides traffic flow control by restricting the delivery of routing updates.

- It can be used as additional security.

Routing and Switching Essentials

* Controls which type of traffic are forwarded or blocked by the router.

* Ability to control which areas a client access.

The ACL is one of the most basic building blocks learned first when venturing into Cisco device configuration. Once the basic structure and logic of these ACLs is understood, they are not particularly hard to configure.

There are several different types of ACL that are defined by either the ACL number or by the syntax used to define the ACL when using named ACLs. Table 1 displays a list of the most commonly used ACL numbers and their associated ACL type.

Table 9-1 - ACL Number Ranges

Protocol	**Range**
Standard IP	1–99 and 1300–1999
Extended IP	100–199 and 2000–2699
Ethernet type code	200–299
Ethernet address	700–799
Transparent bridging (protocol type)	200–299
Transparent bridging (vendor code)	700–799
Extended transparent bridging	1100–1199
DECnet and extended DECnet	300–399
Xerox Network Systems (XNS)	400–499
Extended XNS	500–599
AppleTalk	600–699
Source-route bridging (protocol type)	200–299
Source-route bridging (vendor code)	700–799
Internetwork Packet Exchange (IPX)	800–899
Extended IPX	900–999
IPX Service Advertising Protocol (SAP)	1000–1099
Standard Virtual Integrated Network Service (VINES)	1–100
Extended VINES	101–200
Simple VINES	201–300

2. **Standard Access Control Lists** (1 – 99 and 1300 - 1999)

Standard ACLs are the part of Cisco IOS from its beginning. In earlier days simple filtering was sufficient. Standard ACLs are used for normal filtering. Standard ACLs filter the packet based on its source IP address.

Standard ACLs create filters based on source addresses and are used for server based filtering. Address based access lists distinguish routes on a network you want to control by using network address number (IP). Address-based access lists consist of a list of addresses or address ranges and a statement as to whether access to or from that address is permitted or denied.

Example of the command syntax for configuring a standard numbered IP ACL:

R1(config)# access-list {1-99} {permit | deny} source-addr [source-wildcard]

i. The first value {1-99} specifies the standard ACL number range.

ii. The second value specifies whether to permit or deny the configured source IP address traffic.

iii. The third value is the source IP address that must be matched.

iv. The fourth value is the wildcard mask to be applied to the previously configured IP address to indicate the range.

3. **Extended Access Control Lists** (100 – 199 and 2000 - 2699)

Over the time security becomes more challenging. To mitigate current security threats, advance filtering is required. Extended ACLs takes this responsibility. Extended ACLs can filter a packet based on its sources address, destination address, port number, protocol and much more.

Extended access lists create filters based on source addresses, destination addresses, protocol, port number and other features and are used for packet based filtering for packets that traverse the network.

Example of the command syntax for configuring an extended numbered IP ACL:

Router(config)# access-list {100-199} {permit | deny} protocol source-addr [source-wildcard] [operator operand] destination-addr [destination-wildcard] [operator operand] [established]

i. Like the standard ACLs, the first value {100-199 or 2000 – 2699} specifies the ACL number range.

ii. The next value specifies whether to permit or deny according to the criteria that follows.

iii. The third value specifies protocol type (IP, TCP, UDP, or other specific IP sub-protocols). The source IP address and wildcard mask determine traffic source. The destination IP address and its wildcard mask are used to indicate the final destination of the network traffic. When the destination IP address and mask are configured, the port number must be specified to match, either by number or by a well-known port name, otherwise all traffic to that destination will be dropped.

Standard and Extended access lists can be applied base on the use of ip access-list command.

Access lists use the deny or permit statement to define which packet is allowed or denied entry into a server or network.

Masks

Masks are used with IP addresses in IP ACLs to specify what should be permitted and denied. Masks in order to configure IP addresses on interfaces start with 255 and have the large values on the left side, for example, IP address 172.16.2.14 with a 255.255.255.0 mask. Masks for IP ACLs are the reverse, for example, mask 0.0.0.255.

This is sometimes called an inverse mask or a wildcard mask. When the value of the mask is broken down into binary (0s and 1s), the results determine which address bits are to be considered in processing the traffic. A 0 indicates that the address bits must be considered (exact match); a 1 in the mask is a “no”.

Note these ACL equivalents.

· **The source/source-wildcard of 0.0.0.0/255.255.255.255 means “any”.**

· **The source/wildcard of 10.1.1.2/0.0.0.0 is the same as “host 10.1.1.2”.**

If you subtract **255.255.255.0** (normal mask) from **255.255.255.255**, it yields 0.0.0.255.

Read about Wildcards

The command below defines an ACL that permits this network 192.168.1.0 0.0.0.255.

access-list acl_permit permit ip 192.168.1.0 0.0.0.255

Inbound traffic to the router is compared to access lists entries based on the order that the entries occur in the router. The router looks through the entries until it has a match. If the router found no match when it reaches the end of the list, the traffic is denied. For this reason, you should have the frequently hit entries at the top of the list.

There is an implied deny for traffic that is not permitted. Single-entry access lists with only one deny entry has the effect of denying all traffic. You must have at least one permit statement in an ACL or all traffic is blocked.

Access lists implicitly deny all access that is not expressly permitted. The following line is auto-appended to all access-lists:

deny ip any any

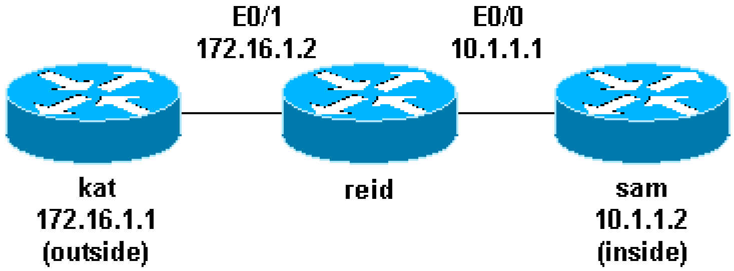
If it is desirable to over-ride this implicit denial statement, enter a permit ip any any statement as the last entry in the access-list.

4. Named Access Control Lists

Named ACLs are the extended version of existing ACLs. Named standard ACL is the extended version of standard ACL. Named extended ACL is the enhanced version of extended ACL. Existing ACLs (Standard and Extended) assign a unique number among all the ACLs. While Named ACLs assign a unique name among all the ACLs.

5. IPv4 ACLs Examples (Standard, Extended and Named)

Network Diagram



Standard ACLs

In all software releases, the *access-list-number* can be anything from 1 to 99. In Cisco IOS Software Release 12.0.1, standard ACLs begin to use additional numbers (1300 to 1999). These additional numbers are referred to as expanded IP ACLs. Cisco IOS Software Release 11.2 added the ability to use list *name* in standard ACLs.

A *source/source-wildcard* setting of 0.0.0.0/255.255.255.255 can be specified as **any**. The wildcard can be omitted if it is all zeros. Therefore, host 10.1.1.2 0.0.0.0 is the same as host 10.1.1.2.

After the ACL is defined, it must be applied to the interface (inbound or outbound). In early software releases, out was the default when a keyword out or in was not specified. The direction must be specified in later software releases.

```
interface <interface>
ip access-group number {in|out}
```

This is an example of the use of a standard ACL in order to block all traffic except that from source 10.1.1.x.

```
interface Ethernet0/0
ip address 10.1.1.1 255.255.255.0
ip access-group 1 in
access-list 1 permit 10.1.1.0 0.0.0.255
```

Extended ACLs

Extended ACLs were introduced in Cisco IOS Software Release 8.3. Extended ACLs control traffic by the comparison of the source and destination addresses of the IP packets to the addresses configured in the ACL.

This is the command syntax format of extended ACLs. Lines are wrapped here for spacing considerations.

IP

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

ICMP

```
access-list access-list-number
     [dynamic dynamic-name [timeout minutes]]
     {deny|permit} icmp source source-wildcard destination destination-wildcard
     [icmp-type [icmp-code] |icmp-message]
     [precedence precedence] [tos tos] [log|log-input]
     [time-range time-range-name]
```

TCP

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

UDP

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

In all software releases, the *access-list-number* can be 100 to 199. In Cisco IOS Software Release 12.0.1, extended ACLs begin to use additional numbers (2000 to 2699). These additional numbers are referred to as expanded IP ACLs. Cisco IOS Software Release 11.2 added the ability to use list *name* in extended ACLs.

The value of 0.0.0.0/255.255.255.255 can be specified as **any**. After the ACL is defined, it must be applied to the interface (inbound or outbound). In early software releases, out was the default when a keyword out or in was not specified. The direction must be specified in later software releases.

```
interface <interface>
ip access-group {number|name} {in|out}
```

This extended ACL is used to permit traffic on the 10.1.1.x network (inside) and to receive ping responses from the outside while it prevents unsolicited pings from people outside, permitting all other traffic.

```
interface Ethernet0/1
ip address 172.16.1.2 255.255.255.0
ip access-group 101 in
access-list 101 deny icmp any 10.1.1.0 0.0.0.255 echo
access-list 101 permit ip any 10.1.1.0 0.0.0.255
```

Named ACLs

IP named ACLs were introduced in Cisco IOS Software Release 11.2. This allows standard and extended ACLs to be given names instead of numbers.

This is the command syntax format for IP named ACLs.

```
ip access-list {extended|standard} name
```

This is a TCP example:

```
{permit|deny} tcp source source-wildcard [operator [port]]
destination destination-wildcard [operator [port]] [established]
[precedence precedence] [tos tos] [log] [time-range time-range-name]
```

This is an example of the use of a named ACL in order to block all traffic except the Telnet connection from host 10.1.1.2 to host 172.16.1.1.

```
interface Ethernet0/0
ip address 10.1.1.1 255.255.255.0
ip access-group in_to_out in

ip access-list extended in_to_out
permit tcp host 10.1.1.2 host 172.16.1.1 eq telnet
```

6. IPv6 ACLs Explained with Examples

There are similarities in operation and configuration of IPv6 ACLs and its predecessor IPv4 ACLs. If you are familiar with the basic operation and configuration of IPv4 access controls list, you will find IPv6 ACLs easy to understand and configure too, the only difference is just the IPv6 addressing.

IPv6 has only one type of ACL, which is comparable to anIPv4 extended named ACL.

There are no numbered ACLs in IPv6, only named ACL.

IPv6 uses the ipv6 traffic-filter command to perform the ACLs function, unlike IPv4 that uses the command ip access-group to apply ACL to an interface.

IPv6 ACLs do not use wildcard masks. Instead, the prefix-length is used to indicate how much of an IPv6 source or destination address should be matched.

How to Configure IPv6 ACLs example topology.

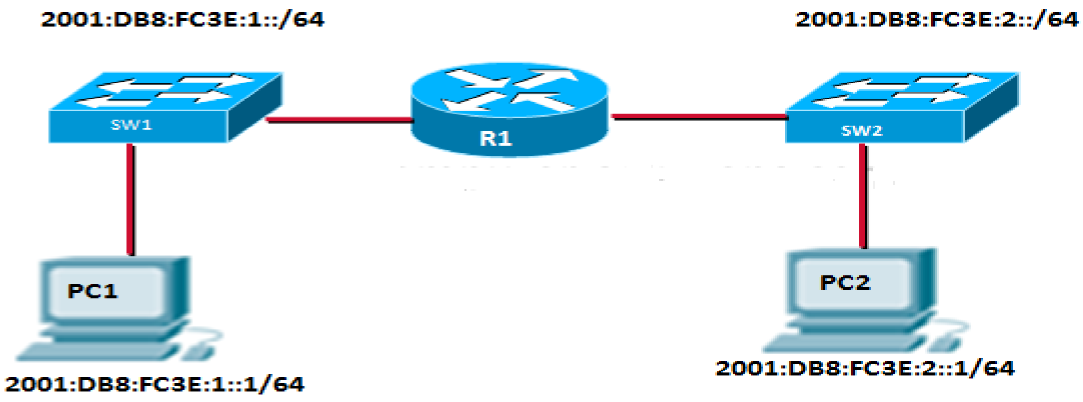
In the example below, we will configure IPv6 ACL on the router to restrict access to its VTY Lines. We will allow only the PC 1 to telnet into R1 while other traffics will be denied.

To do this; You must use the ipv6 access-list command to create a named IPv6 ACL.

IPv6 uses name ACLs as in IPv4, but IPv6 name ACLs are alphanumeric, case sensitive and must be unique.

To determine if a packet is forwarded or dropped, you must use the permit or deny statements to specify this action.

You use the ipv6 access-class command to apply the ACL to the VTY lines.



IPv6 ACL Configuration Example.

From the configuration example below, the permit statement only allows the PC1 to telnet into R1.

Apply the ACL to the VTY lines, using the ipv6 access-class command and with in as the direction.

```
R1(config)#ipv6 access-list NO_TELNET

R1(config-ipv6-acl)#permit tcp host 2001:db8:FC31e:1::1 any eq 23

R1(config-ipv6-acl)#exit

R1(config)#line vty 0 15

R1(config-line)#ipv6 access-class NO_TELNET in

R1(config-line)#exit

R1(config)#
```

VERIFY IPV6 ACLS

To verify all ACLs configured on the router, use the show access-lists command, this will display both IPv4 and IPv6 ACLs configured on the router.

To verify all IPv6 ACLs configured on the router,use the show ipv6 access-list command, this will display all configured IPv6 access lists and their name.

```
R1#show ipv6 access-list

IPv6 access list NO_TELNET

    permit tcp host 2001:DB8:FC31E:1::1 any eq telnet
```

7. General Guidelines for Access Control Lists

- ✓ ACLs are always processed from top to down in sequential order.
- ✓ A packet is compared with ACL conditions until it finds a match.
- ✓ Once a match is found for packet, no further comparison will be done for that packet.
- ✓ Interface will take action based on match condition. There are two possible actions; permit and deny.
- ✓ If permit condition match, packet will be allowed to pass from interface.
- ✓ If deny condition match, packet will be destroyed immediately.
- ✓ Every ACL has a default deny statement at end of it.
- ✓ If a packet does not meet with any condition, it will be destroyed (by the last deny condition).
- ✓ Empty ACL will permit all traffic by default. Implicit deny condition will not work with empty ACL.
- ✓ Implicit (default last deny) condition would work only if ACL has at least one user defined condition.
- ✓ ACL can filter only the traffic passing from interface. It cannot filter the traffic originated from router on which it has been applied.
- ✓ Standard ACL can filter only the source IP address.
- ✓ Standard ACL should be placed near the destination devices.
- ✓ Extended ACL should be placed near the source devices.

- ✓ Each ACL needs a unique number or name.
- ✓ We can have only one ACL applied to an interface in each direction; inbound and outbound.

8. Summary of ACLs Operation

- ACLs can be used for IP packet filtering or to identify traffic to assign it special handling.
- ACLs perform top-down processing and can be configured for incoming or outgoing traffic.
- You can create an ACL using a named or numbered ACL. Named or numbered ACLs can be configured as standard or extended ACLs, which determines what they can filter.
- Reflexive, dynamic, and time-based ACLs add more functionality to standard and extended ACLs.
- Standard IPv4 ACLs allow filtering based on source address.
- Extended IPv4 ACLs allow filtering based on source and destination addresses, as well as protocol and port number.
- IP access list entry sequence numbering allows you to delete individual statements from an ACL to add statements anywhere in the ACL.
- The show access-lists and show ip interface commands are useful for troubleshooting common ACL configuration errors. Show access-list displays all access lists and their parameters configured on the router including the ipx access-lists. Show ip access-list shows only the IP access lists configured on the router. This command doesn't show which interface the list is configured on.
- In a wildcard bit mask, a 0 bit means to match the corresponding address bit, and a 1 bit means to ignore the corresponding address bit.

Reference:

- http://www.ciscopress.com/articles/article.asp?p=1697887

Routing and Switching Essentials