# IP Access List Overview

Access control lists (ACLs) perform packet filtering to control which packets move through a network and to where. The packet filtering provides security by helping to limit the network traffic, restrict the access of users and devices to a network, and prevent the traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as bandwidth control, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features. This module provides an overview of IP access lists.



# Information About IP Access Lists

## Benefits of IP Access Lists

Access control lists (ACLs) perform packet filtering to control the flow of packets through a network. Packet filtering can restrict the access of users and devices to a network, providing a measure of security. Access lists can save network resources by reducing traffic. The benefits of using access lists are as follows:

- Authenticate incoming rsh and rcp requests—Access lists can simplify the identification of local users, remote hosts, and remote users in an authentication database that is configured to control access to a device. The authentication database enables Cisco software to receive incoming remote shell (rsh) and remote copy (rcp) protocol requests.
- Block unwanted traffic or users—Access lists can filter incoming or outgoing packets on an interface, thereby controlling access to a network based on source addresses, destination addresses, or user authentication. You can also use access lists to determine the types of traffic that are forwarded or blocked at device interfaces. For example, you can use access lists to permit e-mail traffic to be routed through a network and to block all Telnet traffic from entering the network.
- Control access to vty—Access lists on an inbound vty (Telnet) can control who can access the lines to a device. Access lists on an outbound vty can control the destinations that the lines from a device can reach.
- Identify or classify traffic for QoS features—Access lists provide congestion avoidance by setting the IP precedence for Weighted Random Early Detection (WRED) and committed access rate (CAR). Access

lists also provide congestion management for class-based weighted fair queueing (CBWFQ), priority queueing, and custom queueing.

- Limit debug command output—Access lists can limit debug output based on an IP address or a protocol.
- Provide bandwidth control—Access lists on a slow link can prevent excess traffic on a network.
- Provide NAT control—Access lists can control which addresses are translated by Network Address Translation (NAT).
- Reduce the chance of DoS attacks—Access lists reduce the chance of denial-of-service (DoS) attacks. Specify IP source addresses to control traffic from hosts, networks, or users from accessing your network. Configure the TCP Intercept feature to can prevent servers from being flooded with requests for connection.
- Restrict the content of routing updates—Access lists can control routing updates that are sent, received, or redistributed in networks.
- Trigger dial-on-demand calls—Access lists can enforce dial and disconnect criteria.

# Restrictions for Access Control Lists

- The **deny ip any any** command does not deny the packets that are not fragmented first when it is preceded by the permit tcp any any *port-number* or permit udp any any *port-number*command.

  Example:

  ```
  permit tcp any any eq 3000
  deny ip any any fragment
  ```

  TCP stream with port number 4000 is not denied for the packets that are not fragmented first. ACE works fine for the first fragment as it has TCP port information.
- Defining multiple access control parameters in a single ACL statement is not allowed. Each access control parameter should be defined with a unique ACL statement.

# Border Routers and Firewall Routers Should Use Access Lists

There are many reasons to configure access lists; for example, you can use access lists to restrict contents of routing updates or to provide traffic flow control. One of the most important reasons to configure access lists is to provide a basic level of security for your network by controlling access to it. If you do not configure access lists on your router, all packets passing through the router could be allowed onto all parts of your network.

An access list can allow one host to access a part of your network and prevent another host from accessing the same area. In the figure below, by applying an appropriate access list to the interfaces of the router, Host A is allowed to access the Human Resources network and Host B is prevented from accessing the Human Resources network.

Access lists should be used in firewall routers, which are often positioned between your internal network and an external network such as the Internet. You can also use access lists on a router positioned between two parts of your network, to control traffic entering or exiting a specific part of your internal network.

To provide some security benefits of access lists, you should at least configure access lists on border routers--routers located at the edges of your networks. Such an access list provides a basic buffer from the outside network or from a less controlled area of your own network into a more sensitive area of your network.

On these border routers, you should configure access lists for each network protocol configured on the router interfaces. You can configure access lists so that inbound traffic or outbound traffic or both are filtered on an interface.

Access lists are defined on a per-protocol basis. In other words, you should define access lists for every protocol enabled on an interface if you want to control traffic flow for that protocol.

# Definition of an Access List

Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting the access of traffic into a network, restricting user and device access to a network, and preventing traffic from leaving a network. IP access lists reduce the chance of spoofing and denial-of-service attacks, and allow dynamic, temporary user-access through a firewall.

IP access lists can also be used for purposes other than security, such as to control bandwidth, restrict the content of routing updates, redistribute routes, trigger dial-on-demand (DDR) calls, limit debug output, and identify or classify traffic for quality of service (QoS) features.

An access list is a sequential list that consists of at least one **permit** statement and possibly one or more **deny** statements. In the case of IP access lists, these statements can apply to IP addresses, upper-layer IP protocols, or other fields in IP packets.

Access lists are identified and referenced by a name or a number. Access lists act as packet filters, filtering packets based on the criteria defined in each access list.

After you configure an access list, for the access list to take effect, you must either apply the access list to an interface (by using the **ip access-group** command), a vty (by using the **access-class** command), or reference the access list by any command that accepts an access list. Multiple commands can reference the same access list.

In the following configuration, an IP access list named branchoffices is configured on Fast Ethernet interface 0/1/0 and applied to incoming packets. Networks other than the ones specified by the source address and mask pair cannot access Fast Ethernet interface 0/1/0. The destinations for packets coming from sources on network 172.16.7.0 are unrestricted. The destination for packets coming from sources on network 172.16.2.0 must be 172.31.5.4.

```
ip access-list extended branchoffices
 10 permit 172.16.7.0 0.0.0.3 any
 20 permit 172.16.2.0 0.0.0.255 host 172.31.5.4
!
interface fastethernet 0/1/0
 ip access-group branchoffices in
```

# Access List Rules

The following rules apply to access lists:

- Only one access list per interface, per protocol, and per direction is allowed.
- An access list must contain at least one **permit** statement or all packets are denied entry into the network.
- The order in which access list conditions or match criteria are configured is important. While deciding whether to forward or block a packet, Cisco software tests the packet against each criteria statement in the order in which these statements are created. After a match is found, no more criteria statements are

checked. The same **permit** or **deny** statements specified in a different order can result in a packet being passed under one circumstance and denied in another circumstance.

- If an access list is referenced by a name, but the access list does not exist, all packets pass. An interface or command with an empty access list applied to it permits all traffic into the network.
- Standard access lists and extended access lists cannot have the same name.
- Inbound access lists process packets before the packets are routed to an outbound interface. Inbound access lists that have filtering criteria that deny packet access to a network saves the overhead of routing lookup. Packets that are permitted access to a network based on the configured filtering criteria are processed for routing. For inbound access lists, when you configure a **permit** statement, packets are processed after they are received, and when you configure a **deny** statement, packets are discarded.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed by the outbound access list. For outbound access lists, when you configure a **permit** statement, packets are sent to the output buffer, and when you configure a **deny** statement, packets are discarded.

**Note** Outbound access list is not supported in Cisco ASR 900 RSP3 Module.

- An access list can control traffic arriving at a device or leaving a device, but not traffic originating at a device.

# Helpful Hints for Creating IP Access Lists

The following tips will help you avoid unintended consequences and help you create more efficient, useful access lists.

- Create the access list before applying it to an interface (or elsewhere), because if you apply a nonexistent access list to an interface and then proceed to configure the access list, the first statement is put into effect, and the implicit **deny** statement that follows could cause you immediate access problems.
- Another reason to configure an access list before applying it is because an interface with an empty access list applied to it permits all traffic.
- All access lists need at least one **permit** statement; otherwise, all packets are denied and no traffic passes.
- Use the statement **permit any any** if you want to allow all other packets not already denied. Using the statement **permit any any** in effect avoids denying all other packets with the implicit deny statement at the end of an access list. Do not make your first access list entry **permit any any** because all traffic will get through; no packets will reach the subsequent testing. In fact, once you specify **permit any any**, all traffic not already denied will get through.
- Although all access lists end with an implicit **deny** statement, we recommend use of an explicit **deny** statement (for example, **deny ip any any**). On most platforms, you can display the count of packets denied by issuing the **show access-list** command, thus finding out more information about who your access list is disallowing. Only packets denied by explicit **deny** statements are counted, which is why the explicit **deny** statement will yield more complete data for you.
- While you are creating an access list or after it is created, you might want to delete an entry.

  - You cannot delete an entry from a numbered access list; trying to do so will delete the entire access list. If you need to delete an entry, you need to delete the entire access list and start over.
  - You can delete an entry from a named access list. Use the **no permit** or **no deny** command to delete the appropriate entry.
- In order to make the purpose of individual statements more scannable and easily understood at a glance, you can write a helpful remark before or after any statement by using the **remark** command.
- If you want to deny access to a particular host or network and find out if someone from that network or host is attempting to gain access, include the **log** keyword with the corresponding **deny** statement so that the packets denied from that source are logged for you.
- This hint applies to the placement of your access list. When trying to save resources, remember that an inbound access list applies the filter conditions before the routing table lookup. An outbound access list applies the filter conditions after the routing table lookup.
- Outbound access list is not supported in Cisco ASR 900 RSP3 Module.

# Named or Numbered Access Lists

All access lists must be identified by a name or a number. Named access lists are more convenient than numbered access lists because you can specify a meaningful name that is easier to remember and associate with a task. You can reorder statements in or add statements to a named access list.

Named access lists support the following features that are not supported by numbered access lists:

- IP options filtering
- Noncontiguous ports
- TCP flag filtering
- Deleting of entries with the **no permit** or **no deny** command

**Note** Not all commands that accept a numbered access list will accept a named access list. For example, vty uses only numbered access lists.

# Standard or Extended Access Lists

All access lists are either standard or extended access lists. If you only intend to filter on a source address, the simpler standard access list is sufficient. For filtering on anything other than a source address, an extended access list is necessary.

- Named access lists are specified as standard or extended based on the keyword **standard** or **extended** in the **ip access-list** command syntax.
- Numbered access lists are specified as standard or extended based on their number in the **access-list** command syntax. Standard IP access lists are numbered 1 to 99 or 1300 to 1999; extended IP access lists are numbered 100 to 199 or 2000 to 2699. The range of standard IP access lists was initially only 1 to 99, and was subsequently expanded with the range 1300 to 1999 (the intervening numbers were assigned to other protocols). The extended access list range was similarly expanded.

**Note** Starting from Cisco IOS XE 16.9.4, use the **ip access-list** command to configure object-group based numbered ACL.

### Standard Access Lists

Standard IP access lists test only source addresses of packets (except for two exceptions). Because standard access lists test source addresses, they are very efficient at blocking traffic close to a destination. There are two exceptions when the address in a standard access list is not a source address:

- On outbound VTY access lists, when someone is trying to telnet, the address in the access list entry is used as a destination address rather than a source address.
- When filtering routes, you are filtering the network being advertised to you rather than a source address.

### Extended Access Lists

Extended access lists are good for blocking traffic anywhere. Extended access lists test source and destination addresses and other IP packet data, such as protocols, TCP or UDP port numbers, type of service (ToS), precedence, TCP flags, and IP options. Extended access lists can also provide capabilities that standard access lists cannot, such as the following:

- Filtering IP Options
- Filtering TCP flags
- Filtering noninitial fragments of packets (see the module "Refining an IP Access List")

**Note** Packets that are subject to an extended access list will not be autonomous switched.

# IP Packet Fields You Can Filter to Control Access

You can use an extended access list to filter on any of the following fields in an IP packet. Source address and destination address are the two most frequently specified fields on which to base an access list:

- Source address--Specifies a source address to control packets coming from certain networking devices or hosts.
- Destination address--Specifies a destination address to control packets being sent to certain networking devices or hosts.
- Protocol--Specifies an IP protocol indicated by the keyword **eigrp**, **gre**, **icmp**, **igmp**, **ip**, **ipinip**, **nos**, **ospf**, **tcp**, or **udp**, or indicated by an integer in the range from 0 to 255 (representing an Internet protocol). If you specify a transport layer protocol (**icmp**, **igmp**, **tcp**, or **udp**), the command has a specific syntax.
  - Ports and non-contiguous ports--Specifies TCP or UDP ports by a port name or port number. The port numbers can be noncontiguous port numbers. Port numbers can be useful to filter Telnet traffic or HTTP traffic, for example.
  - TCP flags--Specifies that packets match any flag or all flags set in TCP packets. Filtering on specific TCP flags can help prevent false synchronization packets.

- IP options--Specifies IP options; one reason to filter on IP options is to prevent routers from being saturated with spurious packets containing them.

# Wildcard Mask for Addresses in an Access List

Address filtering uses wildcard masking to indicate to the software whether to check or ignore corresponding IP address bits when comparing the address bits in an access list entry to a packet being submitted to the access list. By carefully setting wildcard masks, you can specify one or more IP addresses for permit or deny tests.

Wildcard masking for IP address bits uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits. A wildcard mask is sometimes referred to as an inverted mask because a 1 and 0 mean the opposite of what they mean in a subnet (network) mask.

- A wildcard mask bit 0 means check the corresponding bit value; they must match.
- A wildcard mask bit 1 means ignore that corresponding bit value; they need not match.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes an implicit wildcard mask of 0.0.0.0, meaning all values must match.

Unlike subnet masks, which require contiguous bits indicating network and subnet to be ones, wildcard masks allow noncontiguous bits in the mask.

The table below shows examples of IP addresses and masks from an access list, along with the corresponding addresses that are considered a match.

***Table 1: Sample IP Addresses, Wildcard Masks, and Match Results***

| Address | Wildcard Mask | Match Results |
|---|---|---|
| 0.0.0.0 | 255.255.255.255 | All addresses will match the access list conditions. |
| 172.18.0.0/16 | 0.0.255.255 | Network 172.18.0.0 |
| 172.18.5.2/16 | 0.0.0.0 | Only host 172.18.5.2 matches |
| 172.18.8.0 | 0.0.0.7 | Only subnet 172.18.8.0/29 matches |
| 172.18.8.8 | 0.0.0.7 | Only subnet 172.18.8.8/29 matches |
| 172.18.8.15 | 0.0.0.3 | Only subnet 172.18.8.15/30 matches |
| 10.1.2.0 | 0.0.252.255 (noncontiguous bits in mask) | Matches any even-numbered network in the range of 10.1.2.0 to 10.1.254.0 |

# Access List Sequence Numbers

The ability to apply sequence numbers to IP access list entries simplifies access list changes. Prior to the IP Access List Entry Sequence Numbering feature, there was no way to specify the position of an entry within an access list. If you wanted to insert an entry in the middle of an existing list, all of the entries after the desired position had to be removed, then the new entry was added, and then all the removed entries had to be reentered. This method was cumbersome and error prone.

This feature allows users to add sequence numbers to access list entries and resequence them. When you add a new entry, you specify the sequence number so that it is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced to create room to insert the new entry.

# Access List Logging

The Cisco IOS software can provide logging messages about packets permitted or denied by a single standard or extended IP access list entry. That is, any packet that matches the entry will cause an informational logging message about the packet to be sent to the console. The level of messages logged to the console is controlled by the **logging console** global configuration command.

The first packet that triggers the access list entry causes an immediate logging message, and subsequent packets are collected over 5-minute intervals before they are displayed or logged. The logging message includes the access list number, whether the packet was permitted or denied, the source IP address of the packet, and the number of packets from that source permitted or denied in the prior 5-minute interval.

However, you can use the **ip access-list log-update** command to set the number of packets that, when match an access list (and are permitted or denied), cause the system to generate a log message. You might want to do this to receive log messages more frequently than at 5-minute intervals.

**Caution** If you set the *number-of-matches* argument to 1, a log message is sent right away, rather than caching it; every packet that matches an access list causes a log message. A setting of 1 is not recommended because the volume of log messages could overwhelm the system.

Even if you use the **ip access-list log-update** command, the 5-minute timer remains in effect, so each cache is emptied at the end of 5 minutes, regardless of the count of messages in each cache. Regardless of when the log message is sent, the cache is flushed and the count reset to 0 for that message the same way it is when a threshold is not specified.

**Note** The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.

## Alternative to Access List Logging

Packets matching an entry in an ACL with a log option are process switched. It is not recommended to use the log option on ACLs, but rather use NetFlow export and match on a destination interface of Null0. This is done in the CEF path. The destination interface of Null0 is set for any packet that is dropped by the ACL.

# Additional IP Access List Features

Beyond the basic steps to create a standard or extended access list, you can enhance your access lists as mentioned below. Each of these methods is described completely in the module entitled “Refining an Access List.”

- You can impose dates and times when **permit** or **deny** statements in an extended access list are in effect, making your access list more granular and specific to an absolute or periodic time period.

- After you create a named access list, you might want to add entries or change the order of the entries, known as resequencing an access list.
- You can achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

# Where to Apply an Access List

You can apply access lists to the inbound or outbound interfaces of a device. Applying an access list to an inbound interface controls the traffic that enters the interface and applying an access list to an outbound interface controls the traffic that exits the interface.

**Note** Outbound access list is not supported in Cisco ASR 900 RSP3 Module.

When software receives a packet at the inbound interface, the software checks the packet against the statements that are configured for the access list. If the access list permits packets, the software processes the packet. Applying access lists to filter incoming packets can save device resources because filtered packets are discarded before entering the device.

Access lists on outbound interfaces filter packets that are transmitted (sent) out of the interface. You can use the TCP Access Control List (ACL) Splitting feature of the Rate-Based Satellite Control Protocol (RBSCP) on the outbound interface to control the type of packets that are subject to TCP acknowledgment (ACK) splitting on an outbound interface.

You can reference an access list by using a **debug** command to limit the amount of debug logs. For example, based on the filtering or matching criteria of the access list, debug logs can be limited to source or destination addresses or protocols.

You can use access lists to control routing updates, dial-on-demand (DDR), and quality of service (QoS) features.

# Additional References

### Related Documents

| Related Topic | Document Title |
|---|---|
| IP access list commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples | Cisco IOS IP Addressing Services Command Reference |
| Filtering on source address, destination address, or protocol | Creating an IP Access List and Applying It to an Interface" module |
| Filtering on IP Options, TCP flags, noncontiguous ports, or TTL | Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports module |

### Standards

| Standards & RFCs | Title |
|---|---|
| None | — |

### MIBs

| MIB | MIBs Link |
|---|---|
| None | To locate and download MIBs for selected platforms, Cisco IOS XE software releases, and feature sets, use Cisco MIB Locator found at the following URL:<br>http://www.cisco.com/go/mibs |

### Technical Assistance

| Description | Link |
|---|---|
| The Cisco Support and Documentation website provides online resources to download documentation, software, and tools. Use these resources to install and configure the software and to troubleshoot and resolve technical issues with Cisco products and technologies. Access to most tools on the Cisco Support and Documentation website requires a Cisco.com user ID and password. | http://www.cisco.com/cisco/web/support/index.html |