



## **Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S**

### **Americas Headquarters**

Cisco Systems, Inc.  
170 West Tasman Drive  
San Jose, CA 95134-1706  
USA  
<http://www.cisco.com>  
Tel: 408 526-4000  
800 553-NETS (6387)  
Fax: 408 527-0883

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED "AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: <http://www.cisco.com/go/trademarks>. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1110R)

© 2014 Cisco Systems, Inc. All rights reserved.



## CONTENTS

---

### CHAPTER 1

#### **IP Access List Overview 1**

- Finding Feature Information 1
- Information About IP Access Lists 1
  - Benefits of IP Access Lists 1
  - Border Routers and Firewall Routers Should Use Access Lists 2
  - Definition of an Access List 3
  - Access List Rules 3
  - Helpful Hints for Creating IP Access Lists 4
  - Named or Numbered Access Lists 5
  - crStandard or Extended Access Lists 5
  - IP Packet Fields You Can Filter to Control Access 6
  - Wildcard Mask for Addresses in an Access List 7
  - Access List Sequence Numbers 7
  - Access List Logging 8
    - Alternative to Access List Logging 8
  - Additional IP Access List Features 9
  - Where to Apply an Access List 9
- Additional References 9
- Feature Information for IP Access Lists 10

---

### CHAPTER 2

#### **Creating an IP Access List and Applying It to an Interface 13**

- Finding Feature Information 13
- Restrictions for Creating an IP Access List and Applying It to an Interface on Cisco ASR 903 Routers 14
- Information About Creating an IP Access List and Applying It to an Interface 14
  - Helpful Hints for Creating IP Access Lists 14
  - Access List Remarks 15
  - Additional IP Access List Features 15

How to Create an IP Access List and Apply It to an Interface	15
Creating a Standard Access List to Filter on Source Address	16
Creating a Named Access List to Filter on Source Address	16
Creating a Numbered Access List to Filter on Source Address	18
Creating an Extended Access List	20
Creating a Named Extended Access List	20
Creating a Numbered Extended Access List	22
Applying an Access List to an Interface	25
Configuration Examples for Creating an IP Access List and Applying It to an Interface	26
Example: Filtering on Host Source Address	26
Example: Filtering on Subnet Source Address	26
Example: Filtering on Source and Destination Addresses and IP Protocols	26
Example: Filtering on Source Addresses Using a Numbered Access List	27
Example: Preventing Telnet Access to a Subnet	27
Example: Filtering on TCP and ICMP Using Port Numbers	27
Example: Allowing SMTP E-mail and Established TCP Connections	27
Example: Preventing Access to the Web by Filtering on Port Name	28
Example: Filtering on Source Address and Logging the Packets	28
Example: Limiting Debug Output	29
Additional References Creating an IP Access List and Applying It to an Interface	29
Feature Information for Creating an IP Access List and Applying It to an Interface	30

---

## CHAPTER 3

### Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports 33

Finding Feature Information	33
Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports	34
Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports	34
IP Options	34
Benefits of Filtering IP Options	34
Benefits of Filtering on TCP Flags	35
TCP Flags	35
Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature	36
How Filtering on TTL Value Works	36

Benefits of Filtering on TTL Value	37
How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports	37
Filtering Packets That Contain IP Options	37
What to Do Next	39
Filtering Packets That Contain TCP Flags	39
What to Do Next	42
Configuring an Access Control Entry with Noncontiguous Ports	42
Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry	44
What To Do Next	46
Filtering Packets Based on TTL Value	46
Enabling Control Plane Policing to Filter on TTL Values 0 and 1	48
Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports	51
Example: Filtering Packets That Contain IP Options	51
Example: Filtering Packets That Contain TCP Flags	51
Example: Creating an Access List Entry with Noncontiguous Ports	51
Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports	52
Example: Filtering on TTL Value	52
Example: Control Plane Policing to Filter on TTL Values 0 and 1	53
Additional References	53
Feature Information for Creating an IP Access List to Filter	54

---

## CHAPTER 4

### Refining an IP Access List 57

Finding Feature Information	57
Information About Refining an IP Access List	57
Access List Sequence Numbers	57
Benefits of Access List Sequence Numbers	58
Sequence Numbering Behavior	58
Benefits of Time Ranges	59
Benefits Filtering Noninitial Fragments of Packets	59
Access List Processing of Fragments	60
How to Refine an IP Access List	61
Revising an Access List Using Sequence Numbers	61
Restricting an Access List Entry to a Time of Day or Week	64
What to Do Next	66

Configuration Examples for Refining an IP Access List	66
Example Resequencing Entries in an Access List	66
Example Adding an Entry with a Sequence Number	67
Example Adding an Entry with No Sequence Number	67
Example Time Ranges Applied to IP Access List Entries	68
Example Filtering IP Packet Fragments	68
Additional References	68
Feature Information for Refining an IP Access List	69

---

## CHAPTER 5

### IP Named Access Control Lists 71

Finding Feature Information	71
Information About IP Named Access Control Lists	72
Definition of an Access List	72
Named or Numbered Access Lists	72
Benefits of IP Access Lists	73
Access List Rules	73
Helpful Hints for Creating IP Access Lists	74
Where to Apply an Access List	75
How to Configure IP Named Access Control Lists	76
Creating an IP Named Access List	76
Applying an Access List to an Interface	78
Configuration Examples for IP Named Access Control Lists	79
Example: Creating an IP Named Access Control List	79
Example: Applying the Access List to an Interface	79
Additional References for IP Named Access Control Lists	79
Feature Information for IP Named Access Control Lists	80

---

## CHAPTER 6

### Commented IP Access List Entries 81

Finding Feature Information	81
Information About Commented IP Access List Entries	81
Benefits of IP Access Lists	81
Access List Remarks	82
How to Configure Commented IP Access List Entries	83
Writing Remarks in a Named or Numbered Access List	83
Configuration Examples for Commented IP Access List Entries	84

Example: Writing Remarks in an IP Access List	84
Additional References for Commented IP Access List Entries	84
Feature Information for Commented IP Access List Entries	85

---

## CHAPTER 7

### Standard IP Access List Logging 87

Finding Feature Information	87
Restrictions for Standard IP Access List Logging	87
Information About Standard IP Access List Logging	88
Standard IP Access List Logging	88
How to Configure Standard IP Access List Logging	88
Creating a Standard IP Access List Using Numbers	88
Creating a Standard IP Access List Using Names	89
Configuration Examples for Standard IP Access List Logging	91
Example: Creating a Standard IP Access List Using Numbers	91
Example: Creating a Standard IP Access List Using Names	91
Example: Limiting Debug Output	91
Additional References for Standard IP Access List Logging	91
Feature Information for Standard IP Access List Logging	92

---

## CHAPTER 8

### IP Access List Entry Sequence Numbering 93

Finding Feature Information	93
Restrictions for IP Access List Entry Sequence Numbering	93
Information About IP Access List Entry Sequence Numbering	94
Purpose of IP Access Lists	94
How an IP Access List Works	94
IP Access List Process and Rules	94
Helpful Hints for Creating IP Access Lists	95
Source and Destination Addresses	96
Wildcard Mask and Implicit Wildcard Mask	96
Transport Layer Information	96
Benefits IP Access List Entry Sequence Numbering	97
Sequence Numbering Behavior	97
How to Use Sequence Numbers in an IP Access List	98
Sequencing Access-List Entries and Revising the Access List	98
Configuration Examples for IP Access List Entry Sequence Numbering	100

Example: Resequencing Entries in an Access List	100
Example: Adding Entries with Sequence Numbers	101
Example: Entry Without Sequence Number	101
Additional References	102
Feature Information for IP Access List Entry Sequence Numbering	103

---

## CHAPTER 9

### Configuring Lock-and-Key Security (Dynamic Access Lists) 105

Prerequisites for Configuring Lock-and-Key	105
Information About Configuring Lock-and-Key Security (Dynamic Access Lists)	106
About Lock-and-Key	106
Benefits of Lock-and-Key	106
When to Use Lock-and-Key	106
How Lock-and-Key Works	107
Compatibility with Releases Before Cisco IOS Release 11.1	107
Risk of Spoofing with Lock-and-Key	108
Router Performance Impacts with Lock-and-Key	108
Maintaining Lock-and-Key	108
Dynamic Access Lists	109
Lock-and-Key Authentication	109
The autocommand Command	110
How to Configure Lock-and-Key Security (Dynamic Access Lists)	111
Configuring Lock-and-Key	111
Verifying Lock-and-Key Configuration	113
Displaying Dynamic Access List Entries	113
Manually Deleting Dynamic Access List Entries	114
Configuration Examples for Lock-and-Key	114
Example Lock-and-Key with Local Authentication	114
Example Lock-and-Key with TACACS+ Authentication	115

---

## CHAPTER 10

### ACL IP Options Selective Drop 117

Finding Feature Information	117
Restrictions for ACL IP Options Selective Drop	117
Information About ACL IP Options Selective Drop	118
Using ACL IP Options Selective Drop	118
Benefits of Using ACL IP Options Selective Drop	118



How to Configure ACL IP Options Selective Drop	118
Configuring ACL IP Options Selective Drop	118
Configuration Examples for ACL IP Options Selective Drop	119
Example Configuring ACL IP Options Selective Drop	119
Example Verifying ACL IP Options Selective Drop	120
Additional References for IP Access List Entry Sequence Numbering	120
Feature Information for ACL IP Options Selective Drop	121

---

## CHAPTER 11

<b>Displaying and Clearing IP Access List Data Using ACL Manageability</b>	<b>123</b>
Finding Feature Information	123
Information About Displaying and Clearing IP Access List Data Using ACL Manageability	124
Benefits of ACL Manageability	124
Support for Interface-Level ACL Statistics	124
How to Display and Clear IP Access List Data	124
Displaying Global IP ACL Statistics	124
Displaying Interface-Level IP ACL Statistics	125
Clearing the Access List Counters	126
Configuration Examples for Displaying and Clearing IP Access List Data Using ACL Manageability	127
Example Displaying Global IP ACL Statistics	127
Example Displaying Input Statistics	127
Example Displaying Output Statistics	127
Example Displaying Input and Output Statistics	128
Example Clearing Global and Interface Statistics for an IP Access List	128
Example Clearing Global and Interface Statistics for All IP Access Lists	128
Additional References	128
Feature Information for Displaying IP Access List Information and Clearing Counters	129

---

## CHAPTER 12

<b>ACL Syslog Correlation</b>	<b>131</b>
Finding Feature Information	131
Prerequisites for ACL Syslog Correlation	131
Information About ACL Syslog Correlation	132
ACL Syslog Correlation Tags	132
ACE Syslog Messages	132
How to Configure ACL Syslog Correlation	132

Enabling Hash Value Generation on a Device	132
Disabling Hash Value Generation on a Device	134
Configuring ACL Syslog Correlation Using a User-Defined Cookie	135
Configuring ACL Syslog Correlation Using a Hash Value	137
Changing the ACL Syslog Correlation Tag Value	138
Troubleshooting Tips	140
Configuration Examples for ACL Syslog Correlation	140
Example: Configuring ACL Syslog Correlation Using a User-Defined Cookie	140
Example: Configuring ACL Syslog Correlation using a Hash Value	141
Example: Changing the ACL Syslog Correlation Tag Value	141
Additional References for IPv6 IOS Firewall	141
Feature Information for ACL Syslog Correlation	142

---

## CHAPTER 13

### IPv6 Access Control Lists 145

Finding Feature Information	145
Information About IPv6 Access Control Lists	146
Access Control Lists for IPv6 Traffic Filtering	146
IPv6 Packet Inspection	146
Access Class Filtering in IPv6	146
How to Configure IPv6 Access Control Lists	147
Configuring IPv6 Traffic Filtering	147
Creating and Configuring an IPv6 ACL for Traffic Filtering	147
Applying the IPv6 ACL to an Interface	149
Controlling Access to a vty	150
Creating an IPv6 ACL to Provide Access Class Filtering	150
Applying an IPv6 ACL to the Virtual Terminal Line	151
Configuration Examples for IPv6 Access Control Lists	152
Example: Verifying IPv6 ACL Configuration	152
Example: Creating and Applying an IPv6 ACL	153
Example: Controlling Access to a vty	153
Additional References	153
Feature Information for IPv6 Access Control Lists	154

---

## CHAPTER 14

### Configuring Template ACLs 155

Finding Feature Information	155
-----------------------------	-----

Prerequisites for Template ACLs	156
Restrictions for Template ACLs	156
Information About Configuring Template ACLs	156
Template ACL Feature Design	156
Multiple ACLs	157
VSA Cisco-AVPairs	158
RADIUS Attribute 242	158
How to Configure Template ACLs	160
Configuring the Maximum Size of Template ACLs	160
Troubleshooting Tips	161
Configuration Examples for Template ACLs	161
Example Maximum Size of Template ACLs	161
Example Showing ACL Template Summary Information	161
Example Showing ACL Template Tree Information	162
Additional References	162
Feature Information for ACL Template	163

---

## CHAPTER 15

<b>IPv6 Template ACL</b>	<b>165</b>
Finding Feature Information	165
Information About IPv6 ACL—Template ACL	166
IPv6 Template ACL	166
How to Enable IPv6 ACL—Template ACL	166
Enabling IPv6 Template Processing	166
Configuration Examples for IPv6 ACL—Template ACL	167
Example: IPv6 Template ACL Processing	167
Additional References	168
Feature Information for IPv6 ACL—Template ACL	169

---

## CHAPTER 16

<b>IPv4 ACL Chaining Support</b>	<b>171</b>
Finding Feature Information	171
Restrictions for IPv4 ACL Chaining Support	171
Information About IPv4 ACL Chaining Support	172
ACL Chaining Overview	172
IPv4 ACL Chaining Support	172
How to Configure IPv4 ACL Chaining Support	172

Configuring an Interface to Accept Common ACL	173
Configuration Examples for IPv4 ACL Chaining Support	174
Example: Configuring an Interface to Accept a Common ACL	174
Additional References for IPv4 ACL Chaining Support	174
Feature Information for IPv4 ACL Chaining Support	175

---

## CHAPTER 17

### IPv6 ACL Chaining with a Common ACL 177

Finding Feature Information	177
Information About IPv6 ACL Chaining with a Common ACL	177
ACL Chaining Overview	177
IPv6 ACL Chaining with a Common ACL	178
How to Configure IPv6 ACL Chaining with a Common ACL	178
Configuring the IPv6 ACL to an Interface	179
Configuration Examples for IPv6 ACL Chaining with a Common ACL	180
Example: Configuring an Interface to Accept a Common ACL	180
Additional References for IPv6 ACL Chaining with a Common ACL	181
Feature Information for IPv6 ACL Chaining with a Common ACL	182

---

## CHAPTER 18

### IPv6 ACL Extensions for Hop by Hop Filtering 183

Finding Feature Information	183
Information About IPv6 ACL Extensions for Hop by Hop Filtering	183
ACLs and Traffic Forwarding	183
How to Configure IPv6 ACL Extensions for Hop by Hop Filtering	184
Configuring IPv6 ACL Extensions for Hop by Hop Filtering	184
Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering	185
Example: IPv6 ACL Extensions for Hop by Hop Filtering	185
Additional References	186
Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering	187



## CHAPTER

# 1

## 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.

- [Finding Feature Information, page 1](#)
- [Information About IP Access Lists, page 1](#)
- [Additional References, page 9](#)
- [Feature Information for IP Access Lists, page 10](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## 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.

## 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.
- 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.

## 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.

### 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](#)”)
- Time-based entries (see “[Time-Based Access Lists](#)” and 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.

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
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
IP access list commands: complete command syntax, command mode, command history, defaults, usage guidelines, and examples	<a href="#">Cisco IOS IP Addressing Services Command Reference</a>

Related Topic	Document Title
Filtering on source address, destination address, or protocol	<a href="#">“Creating an IP Access List and Applying It to an Interface”</a> module
Filtering on IP Options, TCP flags, noncontiguous ports, or TTL	<a href="#">“Creating an IP Access List to Filter IP Options, TCP Flags, or Noncontiguous Ports”</a> 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:  <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for IP Access Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 2: Feature Information for IP Access Lists**

Feature Name	Releases	Feature Configuration Information
ACL—IP Protocol	Cisco IOS XE Release 3.5S	In Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.







## Creating an IP Access List and Applying It to an Interface

---

IP access lists provide many benefits for securing a network and achieving nonsecurity goals, such as determining quality of service (QoS) factors or limiting **debug** command output. This module describes how to create standard, extended, named, and numbered IP access lists. An access list can be referenced by a name or a number. Standard access lists filter on only the source address in IP packets. Extended access lists can filter on source address, destination address, and other fields in an IP packet.

After you create an access list, you must apply it to something in order for it to have any effect. This module describes how to apply an access list to an interface. However, there are many other uses for access lists, which are mentioned in this module and described in other modules and in other configuration guides for various technologies.

- [Finding Feature Information, page 13](#)
- [Restrictions for Creating an IP Access List and Applying It to an Interface on Cisco ASR 903 Routers, page 14](#)
- [Information About Creating an IP Access List and Applying It to an Interface, page 14](#)
- [How to Create an IP Access List and Apply It to an Interface, page 15](#)
- [Configuration Examples for Creating an IP Access List and Applying It to an Interface, page 26](#)
- [Additional References Creating an IP Access List and Applying It to an Interface, page 29](#)
- [Feature Information for Creating an IP Access List and Applying It to an Interface, page 30](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

# Restrictions for Creating an IP Access List and Applying It to an Interface on Cisco ASR 903 Routers

The following restrictions apply when configuring IPv4 and IPv6 access control lists (ACLs) on Cisco ASR 903 Series Aggregation Services Routers:

- Application control engine (ACE)-specific counters are not supported.
- Layer 3 IPv4 and IPv6 ACLs are not supported on the same interface.
- MAC ACLs are not supported on Ethernet flow points (EFPs) or trunk EFP interfaces to which Layer 3 IPv4 or IPv6 ACLs are applied.
- A maximum of 500 ACEs per ACL are supported.
- IPv4 and IPv6 ACLs are not currently supported on EFP interfaces. IPv4 and IPv6 ACLs are supported on physical interfaces, bridge-domain interfaces, and port-channel interfaces.

## Information About Creating an IP Access List and Applying It to an Interface

### Helpful Hints for Creating IP 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 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.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- A packet will match the first ACE in the ACL. Thus, a **permit ip any any** will match all packets, ignoring all subsequent ACEs.
- 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 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.

## Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
remark Do not allow host1 subnet to telnet out
deny tcp host 172.16.2.88 any eq telnet
```

## 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 *Refining an IP Access List module*.

- 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 or numbered 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.

## How to Create an IP Access List and Apply It to an Interface

This section describes the general ways to create a standard or extended access list using either a name or a number. Access lists are very flexible; the tasks simply illustrate one **permit** command and one **deny** command to provide you the command syntax of each. Only you can determine how many **permit** and **deny** commands you need and their order.

**Note**

The first two tasks in this module create an access list; you must apply the access list in order for it to function. If you want to apply the access list to an interface, perform the task “Applying the Access List to an Interface”.

## Creating a Standard Access List to Filter on Source Address

If you want to filter on source address only, a standard access list is simple and sufficient. There are two alternative types of standard access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features than numbered access lists.

### Creating a Named Access List to Filter on Source Address

Use a standard, named access list if you need to filter on source address only. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list standard *name***
4. **remark *remark***
5. **deny {*source* [*source-wildcard*] | any} [log]**
6. **remark *remark***
7. **permit {*source* [*source-wildcard*] | any} [log]**
8. Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.
9. **end**
10. **show ip access-list**

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>	Enables privileged EXEC mode.
	<b>Example:</b>  Device> enable	<ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list standard <i>name</i></b>  <b>Example:</b> Device(config)# ip access-list standard R&D	Defines a standard IP access list using a name and enters standard named access list configuration mode.
<b>Step 4</b>	<b>remark <i>remark</i></b>  <b>Example:</b> Device(config-std-nacl)# remark deny Sales network	(Optional) Adds a user-friendly comment about an access list entry. <ul style="list-style-type: none"> <li>• A remark can precede or follow an access list entry.</li> <li>• In this example, the remark reminds the network administrator that the subsequent entry denies the Sales network access to the interface (assuming this access list is later applied to an interface).</li> </ul>
<b>Step 5</b>	<b>deny {<i>source</i> [<i>source-wildcard</i>]   <b>any</b>} [log]</b>  <b>Example:</b> Device(config-std-nacl)# deny 172.16.0.0 0.0.255.255 log	(Optional) Denies the specified source based on a source address and wildcard mask. <ul style="list-style-type: none"> <li>• If the <i>source-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.</li> <li>• Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255.</li> <li>• In this example, all hosts on network 172.16.0.0 are denied passing the access list.</li> <li>• Because this example explicitly denies a source address and the <b>log</b> keyword is specified, any packets from that source are logged when they are denied. This is a way to be notified that someone on a network or host is trying to gain access.</li> </ul>
<b>Step 6</b>	<b>remark <i>remark</i></b>  <b>Example:</b> Device(config-std-nacl)# remark Give access to Tester's host	(Optional) Adds a user-friendly comment about an access list entry. <ul style="list-style-type: none"> <li>• A remark can precede or follow an access list entry.</li> <li>• This remark reminds the network administrator that the subsequent entry allows the Tester's host access to the interface.</li> </ul>
<b>Step 7</b>	<b>permit {<i>source</i> [<i>source-wildcard</i>]   <b>any</b>} [log]</b>  <b>Example:</b> Device(config-std-nacl)# permit 172.18.5.22 0.0.0.0	Permits the specified source based on a source address and wildcard mask. <ul style="list-style-type: none"> <li>• Every access list needs at least one <b>permit</b> statement; it need not be the first entry.</li> <li>• If the <i>source-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.</li> </ul>

	Command or Action	Purpose
		<ul style="list-style-type: none"> <li>Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255.</li> <li>In this example, host 172.18.5.22 is allowed to pass the access list.</li> </ul>
<b>Step 8</b>	Repeat some combination of Steps 4 through 7 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit <b>deny</b> statement at the end of the access list.
<b>Step 9</b>	<b>end</b>  <b>Example:</b> Device(config-std-nacl)# end	Exits standard named access list configuration mode and enters privileged EXEC mode.
<b>Step 10</b>	<b>show ip access-list</b>  <b>Example:</b> Device# show ip access-list	(Optional) Displays the contents of all current IP access lists.

## Creating a Numbered Access List to Filter on Source Address

Configure a standard, numbered access list if you need to filter on source address only and you prefer not to use a named access list.

IP standard access lists are numbered 1 to 99 or 1300 to 1999. This task illustrates one **permit** statement and one **deny** statement, but the actual statements you use and their order depend on what you want to filter or allow. Define your **permit** and **deny** statements in the order that achieves your filtering goals.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **permit** {*source* [*source-wildcard*] | **any**} [**log**]
4. **access-list** *access-list-number* **deny** {*source* [*source-wildcard*] | **any**} [**log**]
5. Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.
6. **end**
7. **show ip access-list**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>access-list</b> <i>access-list-number</i> <b>permit</b> { <i>source</i> [ <i>source-wildcard</i> ]   <b>any</b> } [ <b>log</b> ]  <b>Example:</b> Device(config)# access-list 1 permit 172.16.5.22 0.0.0.0	Permits the specified source based on a source address and wildcard mask.  <ul style="list-style-type: none"> <li>Every access list needs at least one permit statement; it need not be the first entry.</li> <li>Standard IP access lists are numbered 1 to 99 or 1300 to 1999.</li> <li>If the source-wildcard is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.</li> <li>Optionally use the keyword <b>any</b> as a substitute for the source source-wildcard to specify the source and source wildcard of 0.0.0.0 255.255.255.255.</li> <li>In this example, host 172.16.5.22 is allowed to pass the access list.</li> </ul>
<b>Step 4</b>	<b>access-list</b> <i>access-list-number</i> <b>deny</b> { <i>source</i> [ <i>source-wildcard</i> ]   <b>any</b> } [ <b>log</b> ]  <b>Example:</b> Device(config)# access-list 1 deny 172.16.7.34 0.0.0.0	Denies the specified source based on a source address and wildcard mask.  <ul style="list-style-type: none"> <li>If the <i>source-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source address.</li> <li>Optionally use the abbreviation <b>any</b> as a substitute for the <i>source source-wildcard</i> to specify the source and source wildcard of 0.0.0.0 255.255.255.255.</li> <li>In this example, host 172.16.7.34 is denied passing the access list.</li> </ul>
<b>Step 5</b>	Repeat some combination of Steps 3 through 6 until you have specified the sources on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit <b>deny</b> statement at the end of the access list.
<b>Step 6</b>	<b>end</b>  <b>Example:</b> Device(config)# end	Exits global configuration mode and enters privileged EXEC mode.

	Command or Action	Purpose
<b>Step 7</b>	<b>show ip access-list</b>  <b>Example:</b> Device# show ip access-list	(Optional) Displays the contents of all current IP access lists.

## Creating an Extended Access List

If you want to filter on anything other than source address, you need to create an extended access list. There are two alternative types of extended access list: named and numbered. Named access lists allow you to identify your access lists with a more intuitive name rather than a number, and they also support more features.

For details on how to filter something other than source or destination address, see the syntax descriptions in the command reference documentation.

### Creating a Named Extended Access List

Create a named extended access list if you want to filter the source and destination address or filter a combination of addresses and other IP fields.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. **deny** *protocol source [source-wildcard] destination [destination-wildcard] [option option-name] [precedence precedence] [tos tos] [established] [log | log-input] [time-range time-range-name] [fragments]*
5. **permit** *protocol source [source-wildcard] destination [destination-wildcard] [option option-name] [precedence precedence] [tos tos] [established] [log | log-input] [time-range time-range-name] [fragments]*
6. Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.
7. **end**
8. **show ip access-list**

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>	Enables privileged EXEC mode.



	Command or Action	Purpose
	<b>Example:</b> <pre>Device&gt; enable</pre>	<ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> <pre>Device# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended <i>name</i></b>  <b>Example:</b> <pre>Device(config)# ip access-list extended acl1</pre>	Defines an extended IP access list using a name and enters extended named access list configuration mode.
<b>Step 4</b>	<b>deny <i>protocol source [source-wildcard] destination [destination-wildcard] [option option-name] [precedence precedence] [tos tos] [established] [log   log-input] [time-range time-range-name] [fragments]</i></b>  <b>Example:</b> <pre>Device(config-ext-nacl)# deny ip 172.18.0.0 0.0.255.255 host 172.16.40.10 log</pre>	<p>(Optional) Denies any packet that matches all of the conditions specified in the statement.</p> <ul style="list-style-type: none"> <li>If the <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.</li> <li>Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> or <i>destination destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255.</li> <li>Optionally use the keyword <b>host</b> <i>source</i> to indicate a source and source wildcard of <i>source</i> 0.0.0.0 or the abbreviation <b>host</b> <i>destination</i> to indicate a destination and destination wildcard of <i>destination</i> 0.0.0.0.</li> <li>In this example, packets from all sources are denied access to the destination network 172.18.0.0. Logging messages about packets permitted or denied by the access list are sent to the facility configured by the <b>logging facility</b> command (for example, console, terminal, or syslog). That is, any packet that matches the access list will cause an informational logging message about the packet to be sent to the configured facility. The level of messages logged to the console is controlled by the <b>logging console</b> command.</li> </ul>
<b>Step 5</b>	<b>permit <i>protocol source [source-wildcard] destination [destination-wildcard] [option option-name] [precedence precedence] [tos tos] [established] [log   log-input] [time-range time-range-name] [fragments]</i></b>	<p>Permits any packet that matches all of the conditions specified in the statement.</p> <ul style="list-style-type: none"> <li>Every access list needs at least one permit statement.</li> <li>If the <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.</li> </ul>

	Command or Action	Purpose
	<b>Example:</b> <pre>Device(config-ext-nacl)# permit tcp any any</pre>	<ul style="list-style-type: none"> <li>Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> or <i>destination destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255.</li> <li>In this example, TCP packets are allowed from any source to any destination.</li> <li>Use the <b>log-input</b> keyword to include input interface, source MAC address, or virtual circuit in the logging output.</li> </ul>
<b>Step 6</b>	Repeat some combination of Steps 4 through 7 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit <b>deny</b> statement at the end of the access list.
<b>Step 7</b>	<b>end</b>  <b>Example:</b> <pre>Device(config-ext-nacl)# end</pre>	Exits standard named access list configuration mode and enters privileged EXEC mode.
<b>Step 8</b>	<b>show ip access-list</b>  <b>Example:</b> <pre>Device# show ip access-list</pre>	(Optional) Displays the contents of all current IP access lists.

## Creating a Numbered Extended Access List

Create a numbered extended access list if you want to filter on source and destination address, or a combination of addresses and other IP fields, and you prefer not to use a name. Extended IP access lists are numbered 100 to 199 or 2000 to 2699.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **remark** *remark*
4. **access-list** *access-list-number* **permit** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
5. **access-list** *access-list-number* **remark** *remark*
6. **access-list** *access-list-number* **deny** *protocol* {*source* [*source-wildcard*] | **any**} {*destination* [*destination-wildcard*] | **any**} [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.
8. **end**
9. **show ip access-list**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>access-list</b> <i>access-list-number</i> <b>remark</b> <i>remark</i>  <b>Example:</b> Device(config)# access-list 107 remark allow Telnet packets from any source to network 172.69.0.0 (headquarters)	(Optional) Adds a user-friendly comment about an access list entry.  <ul style="list-style-type: none"> <li>• A remark of up to 100 characters can precede or follow an access list entry.</li> </ul>
<b>Step 4</b>	<b>access-list</b> <i>access-list-number</i> <b>permit</b> <i>protocol</i> { <i>source</i> [ <i>source-wildcard</i> ]   <b>any</b> } { <i>destination</i> [ <i>destination-wildcard</i> ]   <b>any</b> } [ <b>precedence</b> <i>precedence</i> ] [ <b>tos</b> <i>tos</i> ] [ <b>established</b> ] [ <b>log</b>   <b>log-input</b> ] [ <b>time-range</b> <i>time-range-name</i> ] [ <b>fragments</b> ]	Permits any packet that matches all of the conditions specified in the statement.  <ul style="list-style-type: none"> <li>• Every access list needs at least one <b>permit</b> statement; it need not be the first entry.</li> </ul>

	Command or Action	Purpose
	<p><b>Example:</b></p> <pre>Device(config)# access-list 107 permit tcp any 172.69.0.0 0.0.255.255 eq telnet</pre>	<ul style="list-style-type: none"> <li>Extended IP access lists are numbered 100 to 199 or 2000 to 2699.</li> <li>If the <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.</li> <li>Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> or <i>destination destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255.</li> <li>TCP and other protocols have additional syntax available. See the <b>access-list</b> command in the command reference for complete syntax.</li> </ul>
<b>Step 5</b>	<p><b>access-list access-list-number remark remark</b></p> <p><b>Example:</b></p> <pre>Device(config)# access-list 107 remark deny all other TCP packets</pre>	<p>(Optional) Adds a user-friendly comment about an access list entry.</p> <ul style="list-style-type: none"> <li>A remark of up to 100 characters can precede or follow an access list entry.</li> </ul>
<b>Step 6</b>	<p><b>access-list access-list-number deny protocol {source [source-wildcard]   any} {destination [destination-wildcard]   any} [precedence precedence] [tos tos] [established] [log   log-input] [time-range time-range-name] [fragments]</b></p> <p><b>Example:</b></p> <pre>Device(config)# access-list 107 deny tcp any any</pre>	<p>Denies any packet that matches all of the conditions specified in the statement.</p> <ul style="list-style-type: none"> <li>If the <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.</li> <li>Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> or <i>destination destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255.</li> </ul>
<b>Step 7</b>	Repeat some combination of Steps 3 through 6 until you have specified the fields and values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit <b>deny</b> statement at the end of the access list.
<b>Step 8</b>	<p><b>end</b></p> <p><b>Example:</b></p> <pre>Device(config)# end</pre>	Exits global configuration mode and enters privileged EXEC mode.
<b>Step 9</b>	<p><b>show ip access-list</b></p> <p><b>Example:</b></p> <pre>Device# show ip access-list</pre>	(Optional) Displays the contents of all current IP access lists.

## Applying an Access List to an Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
5. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>type number</i>  <b>Example:</b> Device(config)# interface fastethernet 0/0/0	Specifies an interface and enters interface configuration mode.
<b>Step 4</b>	<b>ip access-group</b> { <i>access-list-number</i>   <i>access-list-name</i> } { <b>in</b>   <b>out</b> }  <b>Example:</b> Device(config-if)# ip access-group acl1 in	Applies the specified access list to the inbound interface.  <ul style="list-style-type: none"> <li>• To filter source addresses, apply the access list to the inbound interface.</li> </ul>
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

# Configuration Examples for Creating an IP Access List and Applying It to an Interface

## Example: Filtering on Host Source Address

In the following example, the workstation belonging to user1 is allowed access to gigabitethernet 0/0/0, and the workstation belonging to user2 is not allowed access:

```
interface gigabitethernet 0/0/0
 ip access-group workstations in
!
ip access-list standard workstations
 remark Permit only user1 workstation through
 permit 172.16.2.88
 remark Do not allow user2 workstation through
 deny 172.16.3.13
```

## Example: Filtering on Subnet Source Address

In the following example, the user1 subnet is not allowed access to gigabitethernet interface 0/0/0, but the Main subnet is allowed access:

```
interface gigabitethernet 0/0/0
 ip access-group prevention in
!
ip access-list standard prevention
 remark Do not allow user1 subnet through
 deny 172.22.0.0 0.0.255.255
 remark Allow Main subnet
 permit 172.25.0.0 0.0.255.255
```

## Example: Filtering on Source and Destination Addresses and IP Protocols

The following configuration example shows an interface with two access lists, one applied to outgoing packets and one applied to incoming packets. The standard access list named Internet-filter filters outgoing packets on source address. The only packets allowed out the interface must be from source 172.16.3.4.

The extended access list named marketing-group filters incoming packets. The access list permits Telnet packets from any source to network 172.26.0.0 and denies all other TCP packets. It permits any ICMP packets. It denies UDP packets from any source to network 172.26.0.0 on port numbers less than 1024. Finally, the access list denies all other IP packets and performs logging of packets passed or denied by that entry.

```
interface gigabitethernet 0/0/0
 ip address 172.20.5.1 255.255.255.0
 ip access-group Internet-filter out
 ip access-group marketing-group in
!
ip access-list standard Internet-filter
 permit 172.16.3.4
ip access-list extended marketing-group
 permit tcp any 172.26.0.0 0.0.255.255 eq telnet
 deny tcp any any
 permit icmp any any
```

```
deny udp any 172.26.0.0 0.0.255.255 lt 1024
deny ip any any
```

## Example: Filtering on Source Addresses Using a Numbered Access List

In the following example, network 10.0.0.0 is a Class A network whose second octet specifies a subnet; that is, its subnet mask is 255.255.0.0. The third and fourth octets of a network 10.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS XE software would accept one address on subnet 48 and reject all others on that subnet. The last line of the list shows that the software would accept addresses on all other network 10.0.0.0 subnets.

```
interface gigabitethernet 0/0/0
 ip access-group 2 in
!
access-list 2 permit 10.48.0.3
access-list 2 deny 10.48.0.0 0.0.255.255
access-list 2 permit 10.0.0.0 0.255.255.255
```

## Example: Preventing Telnet Access to a Subnet

In the following example, the user1 subnet is not allowed to telnet out of gigabitethernet interface 0/0/0:

```
interface gigabitethernet 0/0/0
 ip access-group telnetting out
!
ip access-list extended telnetting
 remark Do not allow user1 subnet to telnet out
 deny tcp 172.20.0.0 0.0.255.255 any eq telnet
 remark Allow Top subnet to telnet out
 permit tcp 172.33.0.0 0.0.255.255 any eq telnet
```

## Example: Filtering on TCP and ICMP Using Port Numbers

In the following example, the first line of the extended access list named acl1 permits any incoming TCP connections with destination ports greater than 1023. The second line permits incoming TCP connections to the Simple Mail Transfer Protocol (SMTP) port of host 172.28.1.2. The last line permits incoming ICMP messages for error feedback.

```
interface gigabitethernet 0/0/0
 ip access-group acl1 in
!
ip access-list extended acl1
 permit tcp any 172.28.0.0 0.0.255.255 gt 1023
 permit tcp any host 172.28.1.2 eq 25
 permit icmp any 172.28.0.0 255.255.255.255
```

## Example: Allowing SMTP E-mail and Established TCP Connections

Suppose you have a network connected to the Internet, and you want any host on an Ethernet to be able to form TCP connections to any host on the Internet. However, you do not want IP hosts to be able to form TCP connections to hosts on the gigabitethernet except to the mail (SMTP) port of a dedicated mail host.

SMTP uses TCP port 25 on one end of the connection and a random port number on the other end. The same two port numbers are used throughout the life of the connection. Mail packets coming in from the Internet

**Example: Preventing Access to the Web by Filtering on Port Name**

will have a destination port of 25. Outbound packets will have the port numbers reversed. The fact that the secure system behind the router always will accept mail connections on port 25 is what makes possible separate control of incoming and outgoing services. The access list can be configured on either the outbound or inbound interface.

In the following example, the gigabitethernet network is a Class B network with the address 172.18.0.0, and the address of the mail host is 172.18.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicate that the packet belongs to an existing connection.

```
interface gigabitethernet 0/0/0
 ip access-group 102 in
!
access-list 102 permit tcp any 172.18.0.0 0.0.255.255 established
access-list 102 permit tcp any host 172.18.1.2 eq 25
```

**Example: Preventing Access to the Web by Filtering on Port Name**

In the following example, the w1 and w2 workstations are not allowed web access; other hosts on network 172.20.0.0 are allowed web access:

```
interface gigabitethernet 0/0/0
 ip access-group no-web out
!
ip access-list extended no-web
 remark Do not allow w1 to browse the web
 deny host 172.20.3.85 any eq http
 remark Do not allow w2 to browse the web
 deny host 172.20.3.13 any eq http
 remark Allow others on our network to browse the web
 permit 172.20.0.0 0.0.255.255 any eq http
```

**Example: Filtering on Source Address and Logging the Packets**

The following example defines access lists 1 and 2, both of which have logging enabled:

```
interface gigabitethernet 0/0/0
 ip address 172.16.1.1 255.0.0.0
 ip access-group 1 in
 ip access-group 2 out
!
access-list 1 permit 172.25.0.0 0.0.255.255 log
access-list 1 deny 172.30.0.0 0.0.255.255 log
!
access-list 2 permit 172.27.3.4 log
access-list 2 deny 172.17.0.0 0.0.255.255 log
```

If the interface receives 10 packets from 172.25.7.7 and 14 packets from 172.17.23.21, the first log will look like the following:

```
list 1 permit 172.25.7.7 1 packet
list 2 deny 172.17.23.21 1 packet
```

Five minutes later, the console will receive the following log:

```
list 1 permit 172.25.7.7 9 packets
list 2 deny 172.17.23.21 13 packets
```



## Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44
```

```
Device# debug mpls ldp advertisements peer-acl acl1
```

```
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

## Additional References Creating an IP Access List and Applying It to an Interface

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>
<ul style="list-style-type: none"> <li>• Order of access list entries</li> <li>• Access list entries based on time of day or week</li> <li>• Packets with noninitial fragments</li> </ul>	<a href="#">Refining an IP Access List</a>
Filtering on IP options, TCP flags, or noncontiguous ports	<a href="#">Creating an IP Access List for Filtering</a>
Controlling logging-related parameters	<a href="#">Understanding Access Control List Logging</a>

**Standards and RFCs**

Standard/RFC	Title
No new or modified standards or RFCs are supported by this feature, and support for existing standards or RFCs has not been modified by this feature.	—

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Creating an IP Access List and Applying It to an Interface

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 3: Feature Information for Creating IP Access Lists and Applying It to an Interface**

Feature Name	Releases	Feature Configuration Information
ACL—Access Control List Source and Destination Address Matching	Cisco IOS XE Release 3.5S	In the Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.
ACL—ICMP Code	Cisco IOS XE Release 3.5S	In the Cisco IOS XE Release 3.5S, support was added for the Cisco ASR 903 Router.

Feature Name	Releases	Feature Configuration Information
ACL Performance Enhancement	Cisco IOS XE Release 2.1	<p>This feature was introduced on the Cisco ASR 1000 Series Aggregation Services Routers.</p> <p>No commands were introduced or modified for this feature.</p>





## Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

This module describes how to use an IP access list to filter IP packets that contain certain IP Options, TCP flags, noncontiguous ports.

- [Finding Feature Information, page 33](#)
- [Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports , page 34](#)
- [Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports , page 34](#)
- [How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports , page 37](#)
- [Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports , page 51](#)
- [Additional References, page 53](#)
- [Feature Information for Creating an IP Access List to Filter, page 54](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

# Prerequisites for Creating an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

Before you perform any of the tasks in this module, you should be familiar with the information in the following modules:

- “IP Access List Overview”
- “Creating an IP Access List and Applying It to an Interface”

## Information About Creating an IP Access List to Filter IP Options, TCP Flags, Noncontiguous Ports

### IP Options

IP uses four key mechanisms in providing its service: Type of Service, Time to Live, Options, and Header Checksum.

The Options, commonly referred to as IP Options, provide for control functions that are required in some situations but unnecessary for the most common communications. IP Options include provisions for time stamps, security, and special routing.

IP Options may or may not appear in datagrams. They must be implemented by all IP modules (host and gateways). What is optional is their transmission in any particular datagram, not their implementation. In some environments the security option may be required in all datagrams.

The option field is variable in length. There may be zero or more options. IP Options can have one of two formats:

- Format 1: A single octet of option-type.
- Format 2: An option-type octet, an option-length octet, and the actual option-data octets.

The option-length octet counts the option-type octet, the option-length octet, and the option-data octets.

The option-type octet is viewed as having three fields: a 1-bit copied flag, a 2-bit option class, and a 5-bit option number. These fields form an 8-bit value for the option type field. IP Options are commonly referred to by their 8-bit value.

For a complete list and description of IP Options, refer to RFC 791, *Internet Protocol* at the following URL: <http://www.faqs.org/rfcs/rfc791.html>

### Benefits of Filtering IP Options

- Filtering of packets that contain IP Options from the network relieves downstream devices and hosts of the load from options packets.

- This feature also minimizes load to the Route Processor (RP) for packets with IP Options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Filtering the packets prevents them from impacting the RP.

## Benefits of Filtering on TCP Flags

The ACL TCP Flags Filtering feature provides a flexible mechanism for filtering on TCP flags. Previously, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.

Because TCP packets can be sent as false synchronization packets that can be accepted by a listening port, it is recommended that administrators of firewall devices set up some filtering rules to drop false TCP packets.

The ACEs that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have a very specific group of TCP flags set or not set. The ACL TCP Flags Filtering feature provides a greater degree of packet-filtering control in the following ways:

- You can select any desired combination of TCP flags on which to filter TCP packets.
- You can configure ACEs to allow matching on a flag that is set, as well as on a flag that is not set.

## TCP Flags

The table below lists the TCP flags, which are further described in RFC 793, *Transmission Control Protocol*.

**Table 4: TCP Flags**

TCP Flag	Purpose
ACK	Acknowledge flag—Indicates that the acknowledgment field of a segment specifies the next sequence number the sender of this segment is expecting to receive.
FIN	Finish flag—Used to clear connections.
PSH	Push flag—Indicates the data in the call should be immediately pushed through to the receiving user.
RST	Reset flag—Indicates that the receiver should delete the connection without further interaction.
SYN	Synchronize flag—Used to establish connections.
URG	Urgent flag—Indicates that the urgent field is meaningful and must be added to the segment sequence number.

## Benefits of Using the Named ACL Support for Noncontiguous Ports on an Access Control Entry Feature

This feature greatly reduces the number of access control entries (ACEs) required in an access control list to handle multiple entries for the same source address, destination address, and protocol. If you maintain large numbers of ACEs, use this feature to consolidate existing groups of access list entries wherever it is possible and when you create new access list entries. When you configure access list entries with noncontiguous ports, you will have fewer access list entries to maintain.

## How Filtering on TTL Value Works

IP extended named and numbered access lists may filter on the TTL value of packets arriving at or leaving an interface. Packets with any possible TTL values 0 through 255 may be permitted or denied (filtered). Like filtering on other fields, such as source or destination address, the **ip access-group** command specifies **in** or **out**, which makes the access list ingress or egress and applies it to incoming or outgoing packets, respectively. The TTL value is checked in conjunction with the specified protocol, application, and any other settings in the access list entry, and all conditions must be met.

### Special Handling for Packets with TTL Value of 0 or 1 Arriving at an Ingress Interface

The software switching paths—distributed Cisco Express Forwarding (dCEF), CEF, fast switching, and process switching—will usually permit or discard the packets based on the access list statements. However, when the TTL value of packets arriving at an ingress interface have a TTL of 0 or 1, special handling is required. The packets with a TTL value of 0 or 1 get sent to the process level before the ingress access list is checked in CEF, dCEF, or the fast switching paths. The ingress access list is applied to packets with TTL values 2 through 255 and a permit or deny decision is made.

Packets with a TTL value of 0 or 1 are sent to the process level because they will never be forwarded out of the device; the process level must check whether each packet is destined for the device and whether an Internet Control Message Protocol (ICMP) TTL Expire message needs to be sent back. This means that even if an ACL with TTL value 0 or 1 filtering is configured on the ingress interface with the intention to drop packets with a TTL of 0 or 1, the dropping of the packets will not happen in the faster paths. It will instead happen in the process level when the process applies the ACL. This is also true for hardware switching platforms. Packets with TTL value of 0 or 1 are sent to the process level of the route processor (RP) or Multilayer Switch Feature Card (MSFC).

On egress interfaces, access list filtering on TTL value works just like other access list features. The check will happen in the fastest switching path enabled in the device. This is because the faster switching paths handle all the TTL values (0 through 255) equally on the egress interface.

### Control Plane Policing for Filtering TTL Values 0 and 1

The special behavior for packets with a TTL value of 0 or 1 results in higher CPU usage for the device. If you are filtering on TTL value of 0 or 1, you should use control plane policing (CPP) to protect the CPU from being overwhelmed. In order to leverage CPP, you must configure an access list especially for filtering TTL values 0 and 1 and apply the access list through CPP. This access list will be a separate access list from any other interface access lists. Because CPP works for the entire system, not just on individual interfaces, you would need to configure only one such special access list for the entire device. This task is described in the section "Enabling Control Plane Policing to Filter on TTL Values 0 and 1".



## Benefits of Filtering on TTL Value

- Filtering on time-to-live (TTL) value provides a way to control which packets are allowed to reach the device or are prevented from reaching the device. By looking at your network layout, you can choose whether to accept or deny packets from a certain device based on how many hops away it is. For example, in a small network, you can deny packets from a location more than three hops away. Filtering on TTL value allows you to validate if the traffic originated from a neighboring device. You can accept only packets that reach you in one hop, for example, by accepting only packets with a TTL value of one less than the initial TTL value of a particular protocol.
- Many control plane protocols communicate only with their neighbors, but receive packets from everyone. By applying an access list that filters on TTL to receiving routers, you can block unwanted packets.
- The Cisco software sends all packets with a TTL value of 0 or 1 to the process level. The device must then send an Internet Control Message Protocol (ICMP) TTL value expire message to the source. By filtering packets that have a TTL value of 0 through 2, you can reduce the load on the process level.

## How to Create an IP Access List to Filter IP Options TCP Flags Noncontiguous Ports

### Filtering Packets That Contain IP Options

Complete these steps to configure an access list to filter packets that contain IP options and to verify that the access list has been configured correctly.

**Note**

- The ACL Support for Filtering IP Options feature can be used only with named, extended ACLs.
- Resource Reservation Protocol (RSVP) Multiprotocol Label Switching Traffic Engineering (MPLS TE), Internet Group Management Protocol Version 2 (IGMPV2), and other protocols that use IP options packets may not function in drop or ignore mode if this feature is configured.
- On most Cisco devices, a packet with IP options is not switched in hardware, but requires control plane software processing (primarily because there is a need to process the options and rewrite the IP header), so all IP packets with IP options will be filtered and switched in software.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **deny** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-value*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Repeat Step 4 or Step 5 as necessary.
7. **end**
8. **show ip access-lists** *access-list-name*

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
Step 2	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
Step 3	<b>ip access-list extended</b> <i>access-list-name</i>  <b>Example:</b> Device(config)# ip access-list extended mylist1	Specifies the IP access list by name and enters named access list configuration mode.
Step 4	[ <i>sequence-number</i> ] <b>deny</b> <i>protocol source source-wildcard destination destination-wildcard</i> [ <b>option</b> <i>option-value</i> ] [ <b>precedence</b> <i>precedence</i> ] [ <b>tos</b> <i>tos</i> ] [ <b>log</b> ] [ <b>time-range</b> <i>time-range-name</i> ] [ <b>fragments</b> ]  <b>Example:</b> Device(config-ext-nacl)# deny ip any any option traceroute	(Optional) Specifies a <b>deny</b> statement in named IP access list mode.  • This access list happens to use a <b>deny</b> statement first, but a <b>permit</b> statement could appear first, depending on the order of statements you need.  • Use the <b>option</b> keyword and <i>option-value</i> argument to filter packets that contain a particular IP Option.  • In this example, any packet that contains the traceroute IP option will be filtered out.  • Use the <b>no</b> <i>sequence-number</i> form of this command to delete an entry.
Step 5	[ <i>sequence-number</i> ] <b>permit</b> <i>protocol source source-wildcard destination destination-wildcard</i>	Specifies a <b>permit</b> statement in named IP access list mode.

	Command or Action	Purpose
	<p>[<b>option</b> <i>option-value</i>] [<b>precedence</b> <i>precedence</i>] [<b>tos</b> <i>tos</i>] [<b>log</b>] [<b>time-range</b> <i>time-range-name</i>] [<b>fragments</b>]</p> <p><b>Example:</b>  Device(config-ext-nacl)# permit ip any any option security</p>	<ul style="list-style-type: none"> <li>• In this example, any packet (not already filtered) that contains the security IP option will be permitted.</li> <li>• Use the <b>no</b> <i>sequence-number</i> form of this command to delete an entry.</li> </ul>
<b>Step 6</b>	Repeat Step 4 or Step 5 as necessary.	Allows you to revise the access list.
<b>Step 7</b>	<p><b>end</b></p> <p><b>Example:</b>  Device(config-ext-nacl)# end</p>	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
<b>Step 8</b>	<p><b>show ip access-lists</b> <i>access-list-name</i></p> <p><b>Example:</b>  Device# show ip access-lists mylist1</p>	(Optional) Displays the contents of the IP access list.

## What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.



### Note

To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

## Filtering Packets That Contain TCP Flags

This task configures an access list to filter packets that contain TCP flags and verifies that the access list has been configured correctly.

**Note**

- TCP flag filtering can be used only with named, extended ACLs.
- The ACL TCP Flags Filtering feature is supported only for Cisco ACLs.
- Previously, the following command-line interface (CLI) format could be used to configure a TCP flag-checking mechanism:

**permit tcp any any rst** The following format that represents the same ACE can now be used: **permit tcp any any match-any +rst** Both the CLI formats are accepted; however, if the new keywords **match-all** or **match-any** are chosen, they must be followed by the new flags that are prefixed with “+” or “-”. It is advisable to use only the old format or the new format in a single ACL. You cannot mix and match the old and new CLI formats.

**Caution**

If a device having ACEs with the new syntax format is reloaded with a previous version of the Cisco software that does not support the ACL TCP Flags Filtering feature, the ACEs will not be applied, leading to possible security loopholes.

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **ip access-list extended *access-list-name***
4. *[sequence-number]* **permit tcp** *source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established] {match-any | match-all} {+ | -} flag-name [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
5. *[sequence-number]* **deny tcp** *source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established] {match-any | match-all} {+ | -} flag-name [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
6. Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no sequence-number** command to delete an entry.
7. **end**
8. **show ip access-lists *access-list-name***

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended access-list-name</b>  <b>Example:</b> Device(config)# ip access-list extended kmd1	Specifies the IP access list by name and enters named access list configuration mode.
<b>Step 4</b>	<code>[sequence-number] permit tcp source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established] {match-any   match-all} {+   -} flag-name [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</code>  <b>Example:</b> Device(config-ext-nacl)# permit tcp any any match-any +rst	Specifies a <b>permit</b> statement in named IP access list mode. <ul style="list-style-type: none"> <li>• This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>• Use the TCP command syntax of the <b>permit</b> command.</li> <li>• Any packet with the RST TCP header flag set will be matched and allowed to pass the named access list kmd1 in Step 3.</li> </ul>
<b>Step 5</b>	<code>[sequence-number] deny tcp source source-wildcard [operator [port]] destination destination-wildcard [operator [port]] [established] {match-any   match-all} {+   -} flag-name [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</code>  <b>Example:</b> Device(config-ext-nacl)# deny tcp any any match-all -ack -fin	(Optional) Specifies a <b>deny</b> statement in named IP access list mode. <ul style="list-style-type: none"> <li>• This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>• Use the TCP command syntax of the <b>deny</b> command.</li> <li>• Any packet that does not have the ACK flag set, and also does not have the FIN flag set, will not be allowed to pass the named access list kmd1 in Step 3.</li> <li>• See the <b>deny(IP)</b> command for additional command syntax to permit upper-layer protocols (ICMP, IGMP, TCP, and UDP).</li> </ul>
<b>Step 6</b>	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the <b>no sequence-number</b> command to delete an entry.	Allows you to revise the access list.
<b>Step 7</b>	<b>end</b>  <b>Example:</b> Device(config-ext-nacl)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
<b>Step 8</b>	<b>show ip access-lists access-list-name</b>	(Optional) Displays the contents of the IP access list.

	Command or Action	Purpose
	<b>Example:</b>  Device# show ip access-lists kmd1	<ul style="list-style-type: none"> <li>Review the output to confirm that the access list includes the new entry.</li> </ul>

## What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

## Configuring an Access Control Entry with Noncontiguous Ports

Perform this task to create access list entries that use noncontiguous TCP or UDP port numbers. Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.



### Note

The ACL—Named ACL Support for Noncontiguous Ports on an Access Control Entry feature can be used only with named, extended ACLs.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. *[sequence-number]* **permit tcp** *source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [established {match-any | match-all} {+ | -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
5. *[sequence-number]* **deny tcp** *source source-wildcard [operator port [port]] destination destination-wildcard [operator [port]] [established {match-any | match-all} {+ | -} flag-name] [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
6. Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the **no sequence-number** command to delete an entry.
7. **end**
8. **show ip access-lists** *access-list-name*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended access-list-name</b>  <b>Example:</b> Device(config)# ip access-list extended acl-extd-1	Specifies the IP access list by name and enters named access list configuration mode.
<b>Step 4</b>	<code>[sequence-number] permit tcp source source-wildcard  [operator port [port]] destination destination-wildcard  [operator [port]] [established {match-any    match-all} {+   -} flag-name] [precedence  precedence] [tos tos] [log] [time-range  time-range-name] [fragments]</code>  <b>Example:</b> Device(config-ext-nacl)# permit tcp any eq telnet ftp any eq 450 679	Specifies a <b>permit</b> statement in named IP access list configuration mode.  <ul style="list-style-type: none"> <li>Operators include <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), and <b>range</b> (inclusive range).</li> <li>If the operator is positioned after the source and source-wildcard arguments, it must match the source port. If the operator is positioned after the destination and destination-wildcard arguments, it must match the destination port.</li> <li>The <b>range</b> operator requires two port numbers. You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators. All other operators require one port number.</li> <li>To filter UDP ports, use the UDP syntax of this command.</li> </ul>
<b>Step 5</b>	<code>[sequence-number] deny tcp source source-wildcard  [operator port [port]] destination destination-wildcard  [operator [port]] [established {match-any    match-all} {+   -} flag-name] [precedence  precedence] [tos tos] [log] [time-range  time-range-name] [fragments]</code>  <b>Example:</b> Device(config-ext-nacl)# deny tcp any neq 45 565 632	(Optional) Specifies a <b>deny</b> statement in named access list configuration mode.  <ul style="list-style-type: none"> <li>Operators include <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), and <b>range</b> (inclusive range).</li> <li>If the <i>operator</i> is positioned after the <i>source</i> and <i>source-wildcard</i> arguments, it must match the source port. If the <i>operator</i> is positioned after the <i>destination</i> and <i>destination-wildcard</i> arguments, it must match the destination port.</li> <li>The <b>range</b> operator requires two port numbers. You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators. All other operators require one port number.</li> <li>To filter UDP ports, use the UDP syntax of this command.</li> </ul>

	Command or Action	Purpose
<b>Step 6</b>	Repeat Step 4 or Step 5 as necessary, adding statements by sequence number where you planned. Use the <b>no sequence-number</b> command to delete an entry.	Allows you to revise the access list.
<b>Step 7</b>	<b>end</b>  <b>Example:</b> Device(config-ext-nacl)# end	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
<b>Step 8</b>	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b> Device# show ip access-lists kmdl	(Optional) Displays the contents of the access list.

## Consolidating Access List Entries with Noncontiguous Ports into One Access List Entry

Perform this task to consolidate a group of access list entries with noncontiguous ports into one access list entry.

Although this task uses TCP ports, you could use the UDP syntax of the **permit** and **deny** commands to filter noncontiguous UDP ports.

Although this task uses a **permit** command first, use the **permit** and **deny** commands in the order that achieves your filtering goals.

### SUMMARY STEPS

1. **enable**
2. **show ip access-lists** *access-list-name*
3. **configure terminal**
4. **ip access-list extended** *access-list-name*
5. **no** [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. [*sequence-number*] **permit** *protocol source source-wildcard* [*operator port* [*port*]] *destination destination-wildcard* [*operator port* [*port*]] [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat Steps 5 and 6 as necessary, adding **permit** or **deny** statements to consolidate access list entries where possible. Use the **no sequence-number** command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*



## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b> Device# show ip access-lists mylist1	(Optional) Displays the contents of the IP access list. <ul style="list-style-type: none"> <li>Review the output to see if you can consolidate any access list entries.</li> </ul>
<b>Step 3</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 4</b>	<b>ip access-list extended</b> <i>access-list-name</i>  <b>Example:</b> Device(config)# ip access-list extended mylist1	Specifies the IP access list by name and enters named access list configuration mode.
<b>Step 5</b>	<b>no</b> [ <i>sequence-number</i> ] <b>permit</b> <i>protocol source source-wildcard destination destination-wildcard</i> [ <b>option</b> <i>option-name</i> ] [ <b>precedence</b> <i>precedence</i> ] [ <b>tos</b> <i>tos</i> ] [ <b>log</b> ] [ <b>time-range</b> <i>time-range-name</i> ] [ <b>fragments</b> ]  <b>Example:</b> Device(config-ext-nacl)# no 10	Removes the redundant access list entry that can be consolidated. <ul style="list-style-type: none"> <li>Repeat this step to remove entries to be consolidated because only the port numbers differ.</li> <li>After this step is repeated to remove the access list entries 20, 30, and 40, for example, those entries are removed because they will be consolidated into one <b>permit</b> statement.</li> <li>If a <i>sequence-number</i> is specified, the rest of the command syntax is optional.</li> </ul>
<b>Step 6</b>	[ <i>sequence-number</i> ] <b>permit</b> <i>protocol source source-wildcard</i> [ <b>operator</b> <i>port</i> [ <i>port</i> ]] <i>destination destination-wildcard</i> [ <b>operator</b> <i>port</i> [ <i>port</i> ]] [ <b>option</b> <i>option-name</i> ] [ <b>precedence</b> <i>precedence</i> ] [ <b>tos</b> <i>tos</i> ] [ <b>log</b> ] [ <b>time-range</b> <i>time-range-name</i> ] [ <b>fragments</b> ]  <b>Example:</b> Device(config-ext-nacl)# permit tcp any neq 45 565 632 any eq 23 45 34 43	Specifies a <b>permit</b> statement in named access list configuration mode. <ul style="list-style-type: none"> <li>In this instance, a group of access list entries with noncontiguous ports was consolidated into one <b>permit</b> statement.</li> <li>You can configure up to 10 ports after the <b>eq</b> and <b>neq</b> operators.</li> </ul>
<b>Step 7</b>	Repeat Steps 5 and 6 as necessary, adding <b>permit</b> or <b>deny</b> statements to consolidate access list entries where possible. Use the <b>no</b> <i>sequence-number</i> command to delete an entry.	Allows you to revise the access list.

	Command or Action	Purpose
<b>Step 8</b>	<b>end</b>  <b>Example:</b> Device(config-std-nacl) # end	(Optional) Exits named access list configuration mode and returns to privileged EXEC mode.
<b>Step 9</b>	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b> Device# show ip access-lists mylist1	(Optional) Displays the contents of the access list.

## What To Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.

## Filtering Packets Based on TTL Value

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.



### Note

When the access list specifies the operation EQ or NEQ, depending on the Cisco software release in use on the device, the access lists can specify up to ten TTL values. The number of TTL values can vary by the Cisco software release.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard* [**option** *option-name*] [**precedence** *precedence*] [**tos** *tos*] [**ttl operator value**] [**log**] [**time-range** *time-range-name*] [**fragments**]
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **interface** *type number*
8. **ip access-group** *access-list-name* {**in** | **out**}

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended access-list-name</b>  <b>Example:</b> Device(config)# ip access-list extended ttlfilter	Defines an IP access list by name.  <ul style="list-style-type: none"> <li>• An access list that filters on TTL value must be an extended access list.</li> </ul>
<b>Step 4</b>	[sequence-number] <b>permit</b> protocol source source-wildcard destination destination-wildcard[option option-name] [precedence precedence] [tos tos] [ttl operator value] [log] [time-range time-range-name] [fragments]  <b>Example:</b> Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2	Sets conditions to allow a packet to pass a named IP access list.  <ul style="list-style-type: none"> <li>• Every access list must have at least one <b>permit</b> statement.</li> <li>• This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.</li> </ul>
<b>Step 5</b>	Continue to add <b>permit</b> or <b>deny</b> statements to achieve the filtering you want.	--
<b>Step 6</b>	<b>exit</b>  <b>Example:</b> Device(config-ext-nacl)# exit	Exits any configuration mode to the next highest mode in the command-line interface (CLI) mode hierarchy.
<b>Step 7</b>	<b>interface type number</b>  <b>Example:</b> Device(config)# interface ethernet 0	Configures an interface type and enters interface configuration mode.
<b>Step 8</b>	<b>ip access-group access-list-name {in   out}</b>  <b>Example:</b> Device(config-if)# ip access-group ttlfilter in	Applies the access list to an interface.

## Enabling Control Plane Policing to Filter on TTL Values 0 and 1

Perform this task to filter IP packets based on a TTL value of 0 or 1 and to protect the CPU from being overwhelmed. This task configures an access list for classification on TTL value 0 and 1, configures the Modular QoS Command-Line Interface (CLI) (MQC), and applies a policy map to the control plane. Any packets that pass the access list are dropped. This special access list is separate from any other interface access lists.

Because access lists are very flexible, it is not possible to define only one combination of **permit** and **deny** commands to filter packets based on the TTL value. This task illustrates just one example that achieves TTL filtering. Configure the appropriate **permit** and **deny** statements that will accomplish your filtering plan.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *access-list-name*
4. [*sequence-number*] **permit** *protocol source source-wildcard destination destination-wildcard ttl operator value*
5. Continue to add **permit** or **deny** statements to achieve the filtering you want.
6. **exit**
7. **class-map** *class-map-name* [**match-all** | **match-any**]
8. **match access-group** {*access-group* | **name** *access-group-name*}
9. **exit**
10. **policy-map** *policy-map-name*
11. **class** {*class-name* | **class-default**}
12. **drop**
13. **exit**
14. **exit**
15. **control-plane**
16. **service-policy** {**input** | **output**} *policy-map-name*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>	Enables privileged EXEC mode.
	<b>Example:</b>  Device> enable	<ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
Step 2	<b>configure terminal</b>	Enters global configuration mode.
	<b>Example:</b>  Device# configure terminal	

	Command or Action	Purpose
<b>Step 3</b>	<b>ip access-list extended</b> <i>access-list-name</i>  <b>Example:</b> <pre>Device(config)# ip access-list extended ttlfilter</pre>	Defines an IP access list by name. <ul style="list-style-type: none"> <li>An access list that filters on a TTL value must be an extended access list.</li> </ul>
<b>Step 4</b>	<b>[sequence-number] permit protocol source source-wildcard destination destination-wildcard ttl operator value</b>  <b>Example:</b> <pre>Device(config-ext-nacl)# permit ip host 172.16.1.1 any ttl lt 2</pre>	Sets conditions to allow a packet to pass a named IP access list. <ul style="list-style-type: none"> <li>Every access list must have at least one <b>permit</b> statement.</li> <li>This example permits packets from source 172.16.1.1 to any destination with a TTL value less than 2.</li> </ul>
<b>Step 5</b>	Continue to add <b>permit</b> or <b>deny</b> statements to achieve the filtering you want.	The packets that pass the access list will be dropped.
<b>Step 6</b>	<b>exit</b>  <b>Example:</b> <pre>Device(config-ext-nacl)# exit</pre>	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
<b>Step 7</b>	<b>class-map class-map-name [match-all   match-any]</b>  <b>Example:</b> <pre>Device(config)# class-map acl-filtering</pre>	Creates a class map to be used for matching packets to a specified class.
<b>Step 8</b>	<b>match access-group {access-group   name access-group-name}</b>  <b>Example:</b> <pre>Device(config-cmap)# match access-group name ttlfilter</pre>	Configures the match criteria for a class map on the basis of the specified access control list.
<b>Step 9</b>	<b>exit</b>  <b>Example:</b> <pre>Device(config-cmap)# exit</pre>	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
<b>Step 10</b>	<b>policy-map policy-map-name</b>  <b>Example:</b> <pre>Device(config)# policy-map acl-filter</pre>	Creates or modifies a policy map that can be attached to one or more interface to specify a service policy.

	Command or Action	Purpose
<b>Step 11</b>	<b>class</b> { <i>class-name</i>   <b>class-default</b> }  <b>Example:</b> Device(config-pmap)# class acl-filter-class	Specifies the name of the class whose policy you want to create or change or to specify the default class (commonly known as the class-default class) before you configure its policy.
<b>Step 12</b>	<b>drop</b>  <b>Example:</b> Device(config-pmap-c)# drop	Configures a traffic class to discard packets belonging to a specific class.
<b>Step 13</b>	<b>exit</b>  <b>Example:</b> Device(config-pmap-c)# exit	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
<b>Step 14</b>	<b>exit</b>  <b>Example:</b> Device(config-pmap)# exit	Exits any configuration mode to the next highest mode in the CLI mode hierarchy.
<b>Step 15</b>	<b>control-plane</b>  <b>Example:</b> Device(config)# control-plane	Associates or modifies attributes or parameters that are associated with the control plane of the device.
<b>Step 16</b>	<b>service-policy</b> { <b>input</b>   <b>output</b> } <i>policy-map-name</i>  <b>Example:</b> Device(config-cp)# service-policy input acl-filter	Attaches a policy map to a control plane for aggregate control plane services.

# Configuration Examples for Filtering IP Options, TCP Flags, Noncontiguous Ports

## Example: Filtering Packets That Contain IP Options

The following example shows an extended access list named `mylist2` that contains access list entries (ACEs) that are configured to permit TCP packets only if they contain the IP Options that are specified in the ACEs:

```
ip access-list extended mylist2
 10 permit ip any any option eool
 20 permit ip any any option record-route
 30 permit ip any any option zsu
 40 permit ip any any option mtup
```

The **show access-list** command has been entered to show how many packets were matched and therefore permitted:

```
Device# show ip access-list mylist2
Extended IP access list test
10 permit ip any any option eool (1 match)
20 permit ip any any option record-route (1 match)
30 permit ip any any option zsu (1 match)
40 permit ip any any option mtup (1 match)
```

## Example: Filtering Packets That Contain TCP Flags

The following access list allows TCP packets only if the TCP flags ACK and SYN are set and the FIN flag is not set:

```
ip access-list extended aaa
 permit tcp any any match-all +ack +syn -fin
end
```

The **show access-list** command has been entered to display the ACL:

```
Device# show access-list aaa

Extended IP access list aaa
 10 permit tcp any any match-all +ack +syn -fin
```

## Example: Creating an Access List Entry with Noncontiguous Ports

The following access list entry can be created because up to ten ports can be entered after the **eq** and **neq** operators:

```
ip access-list extended aaa
 permit tcp any eq telnet ftp any eq 23 45 34
end
```

Enter the **show access-lists** command to display the newly created access list entry.

```
Device# show access-lists aaa
```

```
Extended IP access list aaa
10 permit tcp any eq telnet ftp any eq 23 45 34
```

## Example: Consolidating Some Existing Access List Entries into One Access List Entry with Noncontiguous Ports

The **show access-lists** command is used to display a group of access list entries for the access list named abc:

```
Device# show access-lists abc
Extended IP access list abc
10 permit tcp any eq telnet any eq 450
20 permit tcp any eq telnet any eq 679
30 permit tcp any eq ftp any eq 450
40 permit tcp any eq ftp any eq 679
```

Because the entries are all for the same **permit** statement and simply show different ports, they can be consolidated into one new access list entry. The following example shows the removal of the redundant access list entries and the creation of a new access list entry that consolidates the previously displayed group of access list entries:

```
ip access-list extended abc
no 10
no 20
no 30
no 40
permit tcp any eq telnet ftp any eq 450 679
end
```

When the **show access-lists** command is reentered, the consolidated access list entry is displayed:

```
Device# show access-lists abc
Extended IP access list abc
10 permit tcp any eq telnet ftp any eq 450 679
```

## Example: Filtering on TTL Value

The following access list filters IP packets containing type of service (ToS) level 3 with time-to-live (TTL) values 10 and 20. It also filters IP packets with a TTL greater than 154 and applies that rule to noninitial fragments. It permits IP packets with a precedence level of flash and a TTL value not equal to 1, and it sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended incomingfilter
deny ip any any tos 3 ttl eq 10 20
deny ip any any ttl gt 154 fragments
permit ip any any precedence flash ttl neq 1 log
!
interface ethernet 0

ip access-group incomingfilter in
```



## Example: Control Plane Policing to Filter on TTL Values 0 and 1

The following example configures a traffic class called `acl-filter-class` for use in a policy map called `acl-filter`. An access list permits IP packets from any source having a time-to-live (TTL) value of 0 or 1. Any packets matching the access list are dropped. The policy map is attached to the control plane.

```
ip access-list extended ttlfilter
  permit ip any any ttl eq 0 1
class-map acl-filter-class

  match access-group name ttlfilter

policy-map acl-filter
  class acl-filter-class
    drop
control-plane
  service-policy input acl-filter
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<i>Cisco IOS Security Command Reference</i>
Configuring the device to drop or ignore packets containing IP Options by using the <b>no ip options</b> command.	"ACL IP Options Selective Drop"
Overview information about access lists.	"IP Access List Overview"
Information about creating an IP access list and applying it to an interface	"Creating an IP Access List and Applying It to an Interface"
QoS commands	<i>Cisco IOS Quality of Service Solutions Command Reference</i>

**RFCs**

<b>RFC</b>	<b>Title</b>
RFC 791	<i>Internet Protocol</i> <a href="http://www.faqs.org/rfcs/rfc791.html">http://www.faqs.org/rfcs/rfc791.html</a> <a href="http://www.faqs.org/rfcs/rfc791.html">http://www.faqs.org/rfcs/rfc791.html</a>
RFC 793	<i>Transmission Control Protocol</i>
RFC 1393	<i>Traceroute Using an IP Option</i>

**Technical Assistance**

<b>Description</b>	<b>Link</b>
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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Creating an IP Access List to Filter

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 5: Feature Information for Creating an IP Access List to Filter**

<b>Feature Name</b>	<b>Releases</b>	<b>Feature Configuration Information</b>
ACL--Named ACL Support for Noncontiguous Ports on an Access Control Entry	12.3(7)T 12.2(25)S	This feature allows you to specify noncontiguous ports in a single access control entry, which greatly reduces the number of entries required in an access control list when several entries have the same source address, destination address, and protocol, but differ only in the ports.

Feature Name	Releases	Feature Configuration Information
ACL Support for Filtering IP Options	12.3(4)T 12.2(25)S 15.2(2)S 15.4(1)S	<p>This feature allows you to filter packets having IP Options, in order to prevent routers from becoming saturated with spurious packets.</p> <p>In Cisco IOS Release 15.4(1)S, support was added for the Cisco ASR 901S series routers.</p>
ACL TCP Flags Filtering	12.3(4)T 12.2(25)S	<p>This feature provides a flexible mechanism for filtering on TCP flags. Before Cisco IOS Release 12.3(4)T, an incoming packet was matched as long as any TCP flag in the packet matched a flag specified in the access control entry (ACE). This behavior allows for a security loophole, because packets with all flags set could get past the access control list (ACL). The ACL TCP Flags Filtering feature allows you to select any combination of flags on which to filter. The ability to match on a flag set and on a flag not set gives you a greater degree of control for filtering on TCP flags, thus enhancing security.</p>





## Refining an IP Access List

There are several ways to refine an access list while or after you create it. You can change the order of the entries in an access list or add entries to an access list. You can restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering noninitial fragments of packets.

- [Finding Feature Information, page 57](#)
- [Information About Refining an IP Access List, page 57](#)
- [How to Refine an IP Access List, page 61](#)
- [Configuration Examples for Refining an IP Access List, page 66](#)
- [Additional References, page 68](#)
- [Feature Information for Refining an IP Access List, page 69](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Information About Refining an IP Access List

#### 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.

Sequence numbers allow users to add 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.

## Benefits of Access List Sequence Numbers

An access list sequence number is a number at the beginning of a **permit** or **deny** command in an access list. The sequence number determines the order that the entry appears in the access list. The ability to apply sequence numbers to IP access list entries simplifies access list changes.

Prior to having sequence numbers, users could only add access list entries to the end of an access list; therefore, needing to add statements anywhere except the end of the list required reconfiguring the entire access list. There was no way to specify the position of an entry within an access list. If a user wanted to insert an entry (statement) 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 a user adds a new entry, the user chooses 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. Sequence numbers make revising an access list much easier.

## Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

```
Exceeded maximum sequence number.
```

- If the user enters an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If the user enters an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If the user enters a sequence number that is already present, the following error message is generated:

```
Duplicate sequence number.
```

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment. The function is provided for backward compatibility with software releases that do not support sequence numbering.

- This feature works with named and numbered, standard and extended IP access lists.

## Benefits of Time Ranges

Benefits and possible uses of time ranges include the following:

- The network administrator has more control over permitting or denying a user access to resources. These resources could be an application (identified by an IP address/mask pair and a port number), policy routing, or an on-demand link (identified as interesting traffic to the dialer).
- Network administrators can set time-based security policy, including the following:
  - Perimeter security using access lists
  - Data confidentiality with IP Security Protocol (IPsec)
- When provider access rates vary by time of day, it is possible to automatically reroute traffic cost effectively.
- Network administrators can control logging messages. Access list entries can log traffic at certain times of the day, but not constantly. Therefore, administrators can simply deny access without needing to analyze many logs generated during peak hours.

## Benefits Filtering Noninitial Fragments of Packets

Filter noninitial fragments of packets with an extended access list if you want to block more of the traffic you intended to block, not just the initial fragment of such packets. You should first understand the following concepts.

If the **fragments** keyword is used in additional IP access list entries that deny fragments, the fragment control feature provides the following benefits:

### Additional Security

You are able to block more of the traffic you intended to block, not just the initial fragment of such packets. The unwanted fragments no longer linger at the receiver until the reassembly timeout is reached because they are blocked before being sent to the receiver. Blocking a greater portion of unwanted traffic improves security and reduces the risk from potential hackers.

### Reduced Cost

By blocking unwanted noninitial fragments of packets, you are not paying for traffic you intended to block.

### Reduced Storage

By blocking unwanted noninitial fragments of packets from ever reaching the receiver, that destination does not have to store the fragments until the reassembly timeout period is reached.

**Expected Behavior Is Achieved**

The noninitial fragments will be handled in the same way as the initial fragment, which is what you would expect. There are fewer unexpected policy routing results and fewer fragments of packets being routed when they should not be.

## Access List Processing of Fragments

The behavior of access list entries regarding the use or lack of use of the **fragments** keyword can be summarized as follows:

If the Access-List Entry Has...	Then...
...no <b>fragments</b> keyword (the default), and assuming all of the access-list entry information matches,	<p>For an access list entry that contains only Layer 3 information:</p> <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.</li> </ul> <p>For an access list entry that contains Layer 3 and Layer 4 information:</p> <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets and initial fragments. <ul style="list-style-type: none"> <li>• If the entry is a <b>permit</b> statement, then the packet or fragment is permitted.</li> <li>• If the entry is a <b>deny</b> statement, then the packet or fragment is denied.</li> </ul> </li> <li>• The entry is also applied to noninitial fragments in the following manner. Because noninitial fragments contain only Layer 3 information, only the Layer 3 portion of an access list entry can be applied. If the Layer 3 portion of the access list entry matches, and <ul style="list-style-type: none"> <li>• If the entry is a <b>permit</b> statement, then the noninitial fragment is permitted.</li> <li>• If the entry is a <b>deny</b> statement, then the next access list entry is processed.</li> </ul> </li> </ul> <p><b>Note</b> The <b>deny</b> statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.</p>
...the <b>fragments</b> keyword, and assuming all of the access-list entry information matches,	<p>The access list entry is applied only to noninitial fragments.</p> <p>The <b>fragments</b> keyword cannot be configured for an access list entry that contains any Layer 4 information.</p>



Be aware that you should not add the **fragments** keyword to every access list entry because the first fragment of the IP packet is considered a nonfragment and is treated independently of the subsequent fragments. An initial fragment will not match an access list **permit** or **deny** entry that contains the **fragments** keyword. The packet is compared to the next access list entry, and so on, until it is either permitted or denied by an access list entry that does not contain the **fragments** keyword. Therefore, you may need two access list entries for every **deny** entry. The first **deny** entry of the pair will not include the **fragments** keyword and applies to the initial fragment. The second **deny** entry of the pair will include the **fragments** keyword and applies to the subsequent fragments. In the cases in which there are multiple **deny** entries for the same host but with different Layer 4 ports, a single **deny** access list entry with the **fragments** keyword for that host is all that needs to be added. Thus all the fragments of a packet are handled in the same manner by the access list.

Packet fragments of IP datagrams are considered individual packets, and each counts individually as a packet in access list accounting and access list violation counts.

## How to Refine an IP Access List

The tasks in this module provide you with various ways to refine an access list if you did not already do so while you were creating it. You can change the order of the entries in an access list, add entries to an access list, restrict access list entries to a certain time of day or week, or achieve finer granularity when filtering packets by filtering on noninitial fragments of packets.

### Revising an Access List Using Sequence Numbers

Perform this task if you want to add entries to an existing access list, change the order of entries, or simply number the entries in an access list to accommodate future changes.

**Note**

Remember that if you want to delete an entry from an access list, you can simply use the **no deny** or **no permit** form of the command, or the **no sequence-number** command if the statement already has a sequence number.

**Note**

- Access list sequence numbers do not support dynamic, reflexive, or firewall access lists.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list** {**standard**|**extended**} *access-list-name*
5. Do one of the following:
  - *sequence-number* **permit** *source source-wildcard*
  - *sequence-number* **permit** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
6. Do one of the following:
  - *sequence-number* **deny** *source source-wildcard*
  - *sequence-number* **deny** *protocol source source-wildcard destination destination-wildcard* [**precedence** *precedence*][**tos** *tos*] [**log**] [**time-range** *time-range-name*] [**fragments**]
7. Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the **no sequence-number** command to delete an entry.
8. **end**
9. **show ip access-lists** *access-list-name*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list resequence</b> <i>access-list-name starting-sequence-number increment</i>  <b>Example:</b> Router(config)# ip access-list resequence kmd1 100 15	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.  • This example resequences an access list named kmd1. The starting sequence number is 100 and the increment is 15.
<b>Step 4</b>	<b>ip access-list</b> { <b>standard</b>   <b>extended</b> } <i>access-list-name</i>	Specifies the IP access list by name and enters named access list configuration mode.

	Command or Action	Purpose
	<p><b>Example:</b></p> <pre>Router(config)# ip access-list standard xyz123</pre>	<ul style="list-style-type: none"> <li>If you specify <b>standard</b>, make sure you specify subsequent <b>permit</b> and <b>deny</b> statements using the standard access list syntax.</li> <li>If you specify <b>extended</b>, make sure you specify subsequent <b>permit</b> and <b>deny</b> statements using the extended access list syntax.</li> </ul>
<b>Step 5</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li><i>sequence-number permit source source-wildcard</i></li> <li><i>sequence-number permit protocol source source-wildcard destination destination-wildcard [precedence precedence][tos tos] [log] [time-range time-range-name] [fragments]</i></li> </ul> <p><b>Example:</b></p> <pre>Router(config-std-nacl)# 105 permit 10.5.5.5 0.0.0.255</pre>	<p>Specifies a permit statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>See the <b>permit</b> (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP).</li> <li>Use the <b>no sequence-number</b> command to delete an entry.</li> <li>As the prompt indicates, this access list was a standard access list. If you had specified <b>extended</b> in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended <b>permit</b> command syntax.</li> </ul>
<b>Step 6</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li><i>sequence-number deny source source-wildcard</i></li> <li><i>sequence-number deny protocol source source-wildcard destination destination-wildcard [precedence precedence][tos tos] [log] [time-range time-range-name] [fragments]</i></li> </ul> <p><b>Example:</b></p> <pre>Router(config-std-nacl)# 110 deny 10.6.6.7 0.0.0.255</pre>	<p>(Optional) Specifies a deny statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>See the <b>deny</b> (IP) command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP).</li> <li>Use the <b>no sequence-number</b> command to delete an entry.</li> <li>As the prompt indicates, this access list was a standard access list. If you had specified <b>extended</b> in Step 4, the prompt for this step would be Router(config-ext-nacl)# and you would use the extended <b>deny</b> command syntax.</li> </ul>
<b>Step 7</b>	<p>Repeat Step 5 and Step 6 as necessary, adding statements by sequence number where you planned. Use the <b>no sequence-number</b> command to delete an entry.</p>	<p>Allows you to revise the access list.</p>
<b>Step 8</b>	<p><b>end</b></p> <p><b>Example:</b></p> <pre>Router(config-std-nacl)# end</pre>	<p>(Optional) Exits the configuration mode and returns to privileged EXEC mode.</p>

	Command or Action	Purpose
<b>Step 9</b>	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b> Router# show ip access-lists xyz123	(Optional) Displays the contents of the IP access list.  <ul style="list-style-type: none"> <li>Review the output to see that the access list includes the new entry.</li> </ul>

### Examples

The following is sample output from the **show ip access-lists** command when the **xyz123** access list is specified.

```
Router# show ip access-lists xyz123
Standard IP access list xyz123
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.5, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

## Restricting an Access List Entry to a Time of Day or Week

By default, access list statements are always in effect once they are applied. However, you can define the times of the day or week that **permit** or **deny** statements are in effect by defining a time range, and then referencing the time range by name in an individual access list statement. IP and Internetwork Packet Exchange (IPX) named or numbered extended access lists can use time ranges.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended** *name*
4. *[sequence-number]* **deny** *protocol* *source[source-wildcard]* *[operator port[port]]* *destination[destination-wildcard]* *[operator port[port]]*
5. *[sequence-number]* **deny** *protocol* *source[source-wildcard]* *[operator port[port]]* *destination[destination-wildcard]* *[operator port[port]]* **fragments**
6. *[sequence-number]* **permit** *protocol* *source[source-wildcard]* *[operator port[port]]* *destination[destination-wildcard]* *[operator port[port]]*
7. Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.
8. **end**
9. **show ip access-list**

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended</b> <i>name</i>  <b>Example:</b> Router(config)# ip access-list extended rstrct4	Defines an extended IP access list using a name and enters extended named access list configuration mode.
<b>Step 4</b>	[ <i>sequence-number</i> ] <b>deny</b> <i>protocol</i> <i>source</i> [ <i>source-wildcard</i> ] [ <i>operator port</i> [ <i>port</i> ]] <i>destination</i> [ <i>destination-wildcard</i> ] [ <i>operator port</i> [ <i>port</i> ]]  <b>Example:</b> Router(config-ext-nacl)# deny ip any 172.20.1.1	(Optional) Denies any packet that matches all of the conditions specified in the statement.  <ul style="list-style-type: none"> <li>• This statement will apply to nonfragmented packets and initial fragments.</li> </ul>
<b>Step 5</b>	[ <i>sequence-number</i> ] <b>deny</b> <i>protocol</i> <i>source</i> [ <i>source-wildcard</i> ][ <i>operator port</i> [ <i>port</i> ]] <i>destination</i> [ <i>destination-wildcard</i> ] [ <i>operator port</i> [ <i>port</i> ]] <b>fragments</b>  <b>Example:</b> Router(config-ext-nacl)# deny ip any 172.20.1.1 fragments	(Optional) Denies any packet that matches all of the conditions specified in the statement  <ul style="list-style-type: none"> <li>• This statement will apply to noninitial fragments.</li> </ul>
<b>Step 6</b>	[ <i>sequence-number</i> ] <b>permit</b> <i>protocol</i> <i>source</i> [ <i>source-wildcard</i> ] [ <i>operator port</i> [ <i>port</i> ]] <i>destination</i> [ <i>destination-wildcard</i> ] [ <i>operator port</i> [ <i>port</i> ]]  <b>Example:</b> Router(config-ext-nacl)# permit tcp any any	Permits any packet that matches all of the conditions specified in the statement.  <ul style="list-style-type: none"> <li>• Every access list needs at least one <b>permit</b> statement.</li> <li>• If the <i>source-wildcard</i> or <i>destination-wildcard</i> is omitted, a wildcard mask of 0.0.0.0 is assumed, meaning match on all bits of the source or destination address, respectively.</li> <li>• Optionally use the keyword <b>any</b> as a substitute for the <i>source source-wildcard</i> or <i>destination destination-wildcard</i> to specify the address and wildcard of 0.0.0.0 255.255.255.255.</li> </ul>

	Command or Action	Purpose
<b>Step 7</b>	Repeat some combination of Steps 4 through 6 until you have specified the values on which you want to base your access list.	Remember that all sources not specifically permitted are denied by an implicit <b>deny</b> statement at the end of the access list.
<b>Step 8</b>	<b>end</b>  <b>Example:</b> Router(config-ext-nacl)# end	Ends configuration mode and returns the system to privileged EXEC mode.
<b>Step 9</b>	<b>show ip access-list</b>  <b>Example:</b> Router# show ip access-list	(Optional) Displays the contents of all current IP access lists.

## What to Do Next

Apply the access list to an interface or reference it from a command that accepts an access list.



### Note

To effectively eliminate all packets that contain IP Options, we recommend that you configure the global **ip options drop** command.

# Configuration Examples for Refining an IP Access List

## Example Resequencing Entries in an Access List

The following example shows an access list before and after resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values that users provide, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default it has a sequence number of 10 more than the last entry in the access list.

```
Router# show access-list carls
Extended IP access list carls
 10 permit ip host 10.3.3.3 host 172.16.5.34
 20 permit icmp any any
 30 permit tcp any host 10.3.3.3
 40 permit ip host 10.4.4.4 any
 50 Dynamic test permit ip any any
 60 permit ip host 172.16.2.2 host 10.3.3.12
 70 permit ip host 10.3.3.3 any log
 80 permit tcp host 10.3.3.3 host 10.1.2.2
 90 permit ip host 10.3.3.3 any
```

```

100 permit ip any any
Router(config)# ip access-list extended carls
Router(config)# ip access-list resequence carls 1 2
Router(config)# end
Router# show access-list carls
Extended IP access list carls
  1 permit ip host 10.3.3.3 host 172.16.5.34
  3 permit icmp any any
  5 permit tcp any host 10.3.3.3
  7 permit ip host 10.4.4.4 any
  9 Dynamic test permit ip any any
 11 permit ip host 172.16.2.2 host 10.3.3.12
 13 permit ip host 10.3.3.3 any log
 15 permit tcp host 10.3.3.3 host 10.1.2.2
 17 permit ip host 10.3.3.3 any
 19 permit ip any any

```

## Example Adding an Entry with a Sequence Number

In the following example, a new entry (sequence number 15) is added to an access list:

```

Router# show ip access-list
Standard IP access list tryon
 2 permit 10.4.4.2, wildcard bits 0.0.255.255
 5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255
Router(config)# ip access-list standard tryon
Router(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Router# show ip access-list
Standard IP access list tryon
 2 permit 10.4.0.0, wildcard bits 0.0.255.255
 5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255

```

## Example Adding an Entry with No Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```

Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
Router(config)# ip access-list standard resources
Router(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Router(config-std-nacl)# end
Router# show access-list
Standard IP access list resources
10 permit 10.1.1.1, wildcard bits 0.0.0.255
20 permit 10.2.2.2, wildcard bits 0.0.0.255
30 permit 10.3.3.3, wildcard bits 0.0.0.255
40 permit 10.4.4.4, wildcard bits 0.0.0.255

```

## Example Time Ranges Applied to IP Access List Entries

The following example creates a time range called `no-http`, which extends from Monday to Friday from 8:00 a.m. to 6:00 p.m. That time range is applied to the **deny** statement, thereby denying HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.

The time range called `udp-yes` defines weekends from noon to 8:00 p.m. That time range is applied to the **permit** statement, thereby allowing UDP traffic on Saturday and Sunday from noon to 8:00 p.m. only. The access list containing both statements is applied to inbound packets on Fast Ethernet interface 0/0/0.

```
time-range no-http
 periodic weekdays 8:00 to 18:00
!
time-range udp-yes
 periodic weekend 12:00 to 20:00
!
ip access-list extended strict
 deny tcp any any eq http time-range no-http
 permit udp any any time-range udp-yes
!
interface fastethernet 0/0/0
 ip access-group strict in
```

## Example Filtering IP Packet Fragments

In the following access list, the first statement will deny only noninitial fragments destined for host 172.16.1.1. The second statement will permit only the remaining nonfragmented and initial fragments that are destined for host 172.16.1.1 TCP port 80. The third statement will deny all other traffic. In order to block noninitial fragments for any TCP port, we must block noninitial fragments for all TCP ports, including port 80 for host 172.16.1.1. That is, non-initial fragments will not contain Layer 4 port information, so, in order to block such traffic for a given port, we have to block fragments for all ports.

```
access-list 101 deny ip any host 172.16.1.1 fragments
access-list 101 permit tcp any host 172.16.1.1 eq 80
access-list 101 deny ip any any
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
Using the <b>time-range</b> command to establish time ranges	The chapter “Performing Basic System Management” in the <i>Cisco IOS XE Network Management Configuration Guide</i>
Network management command descriptions	<i>Cisco IOS Network Management Command Reference</i>



**Standards**

Standard	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

**MIBs**

MIB	MIBs Link
No new or modified MIBs are supported by this feature, and support for existing MIBs has not been modified by this feature.	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:  <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

**RFCs**

RFC	Title
No new or modified RFCs are supported by this feature, and support for existing RFCs has not been modified by this feature.	--

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Refining an IP Access List

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 6: Feature Information for Refining an IP Access List**

Feature Name	Releases	Feature Configuration Information
Time-Based Access Lists	Cisco IOS XE Release 2.1	This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.  No commands were introduced or modified for this feature.



## IP Named Access Control Lists

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.

The IP Named Access Control Lists feature gives network administrators the option of using names to identify their access lists.

This module describes IP named access lists and how to configure them.

- [Finding Feature Information, page 71](#)
- [Information About IP Named Access Control Lists, page 72](#)
- [How to Configure IP Named Access Control Lists, page 76](#)
- [Configuration Examples for IP Named Access Control Lists, page 79](#)
- [Additional References for IP Named Access Control Lists, page 79](#)
- [Feature Information for IP Named Access Control Lists, page 80](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

# Information About IP Named Access Control Lists

## 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
```

## 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.

## 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.

## 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.
- 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.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- 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.

## 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.

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.

# How to Configure IP Named Access Control Lists

## Creating an IP Named Access List

You can create an IP named access list to filter source addresses and destination addresses or a combination of addresses and other IP fields. Named access lists allow you to identify your access lists with an intuitive name.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list extended *name***
4. **remark *remark***
5. **deny *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} [**log**]**
6. **remark *remark***
7. **permit *protocol* [*source source-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} {*destination* [*destination-wildcard*] {**any** | **host** {*address* | *name*} | **object-group** *object-group-name*} [**log**]**
8. Repeat Steps 4 through 7 to specify more statements for your access list.
9. **end**
10. **show ip access-lists**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list extended <i>name</i></b>  <b>Example:</b> Device(config)# ip access-list extended acl1	Defines an extended IP access list using a name and enters extended named access list configuration mode.



	Command or Action	Purpose
<b>Step 4</b>	<b>remark</b> <i>remark</i>  <b>Example:</b> Device(config-ext-nacl)# remark protect server by denying sales access to the acl1 network	(Optional) Adds a description for an access list statement. <ul style="list-style-type: none"> <li>• A remark can precede or follow an IP access list entry.</li> <li>• In this example, the <b>remark</b> command reminds the network administrator that the <b>deny</b> command configured in Step 5 denies the Sales network access to the interface.</li> </ul>
<b>Step 5</b>	<b>deny</b> <i>protocol</i> [ <i>source source-wildcard</i> ] { <b>any</b>   <b>host</b> { <i>address</i>   <i>name</i> }   <b>object-group</b> <i>object-group-name</i> } { <i>destination</i> [ <i>destination-wildcard</i> ] { <b>any</b>   <b>host</b> { <i>address</i>   <i>name</i> }   <b>object-group</b> <i>object-group-name</i> } [ <b>log</b> ]}  <b>Example:</b> Device(config-ext-nacl)# deny ip 192.0.2.0 0.0.255.255 host 192.0.2.10 log	(Optional) Denies all packets that match all conditions specified by the remark.
<b>Step 6</b>	<b>remark</b> <i>remark</i>  <b>Example:</b> Device(config-ext-nacl)# remark allow TCP from any source to any destination	(Optional) Adds a description for an access list statement. <ul style="list-style-type: none"> <li>• A remark can precede or follow an IP access list entry.</li> </ul>
<b>Step 7</b>	<b>permit</b> <i>protocol</i> [ <i>source source-wildcard</i> ] { <b>any</b>   <b>host</b> { <i>address</i>   <i>name</i> }   <b>object-group</b> <i>object-group-name</i> } { <i>destination</i> [ <i>destination-wildcard</i> ] { <b>any</b>   <b>host</b> { <i>address</i>   <i>name</i> }   <b>object-group</b> <i>object-group-name</i> } [ <b>log</b> ]}  <b>Example:</b> Device(config-ext-nacl)# permit tcp any any	Permits all packets that match all conditions specified by the statement.
<b>Step 8</b>	Repeat Steps 4 through 7 to specify more statements for your access list.	<b>Note</b> All source addresses that are not specifically permitted by a statement are denied by an implicit deny statement at the end of the access list.
<b>Step 9</b>	<b>end</b>  <b>Example:</b> Device(config-ext-nacl)# end	Exits extended named access list configuration mode and returns to privileged EXEC mode.
<b>Step 10</b>	<b>show ip access-lists</b>  <b>Example:</b> Device# show ip access-lists	Displays the contents of all current IP access lists.

**Example:**

The following is sample output from the **show ip access-lists** command:

```
Device# show ip access-lists acl1

Extended IP access list acl1
  permit tcp any 192.0.2.0 255.255.255.255 eq telnet
  deny tcp any any
  deny udp any 192.0.2.0 255.255.255.255 lt 1024
  deny ip any any log
```

## Applying an Access List to an Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip access-group** {*access-list-number* | *access-list-name*} {**in** | **out**}
5. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>type number</i>  <b>Example:</b> Device(config)# interface fastethernet 0/0/0	Specifies an interface and enters interface configuration mode.
<b>Step 4</b>	<b>ip access-group</b> { <i>access-list-number</i>   <i>access-list-name</i> } { <b>in</b>   <b>out</b> }  <b>Example:</b> Device(config-if)# ip access-group acl1 in	Applies the specified access list to the inbound interface.  <ul style="list-style-type: none"> <li>• To filter source addresses, apply the access list to the inbound interface.</li> </ul>
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and returns to privileged EXEC mode.

# Configuration Examples for IP Named Access Control Lists

## Example: Creating an IP Named Access Control List

```
Device# configure terminal
Device(config)# ip access-list extended acl1
Device(config-ext-nacl)# remark protect server by denying sales access to the acl1 network
Device(config-ext-nacl)# deny ip 192.0.2.0 0.0.255.255 host 192.0.2.10 log
Device(config-ext-nacl)# remark allow TCP from any source to any destination
Device(config-ext-nacl)# permit tcp any any
```

## Example: Applying the Access List to an Interface

```
Device# configure terminal
Device(config)# interface fastethernet 0/0/0
Device(config-if)# ip access-group acl1 in
```

## Additional References for IP Named Access Control Lists

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<ul style="list-style-type: none"><li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li></ul>

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for IP Named Access Control Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 7: Feature Information for IP Named Access Control Lists**

Feature Name	Releases	Feature Information
IP Named Access Control Lists	Cisco IOS XE Release 2.1	Access control lists (ACLs) perform packet filtering to control the movement of packets through a network. Packet filtering provides security by limiting 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.



## Commented IP Access List Entries

The Commented IP Access List Entries feature allows you to include comments or remarks about **deny** or **permit** conditions in any IP access list. These remarks make access lists easier for network administrators to understand. Each remark is limited to 100 characters in length.

This module provides information about the Commented IP Access List Entries feature.

- [Finding Feature Information, page 81](#)
- [Information About Commented IP Access List Entries, page 81](#)
- [How to Configure Commented IP Access List Entries, page 83](#)
- [Configuration Examples for Commented IP Access List Entries, page 84](#)
- [Additional References for Commented IP Access List Entries, page 84](#)
- [Feature Information for Commented IP Access List Entries, page 85](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Information About Commented IP Access List Entries

### 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.

## Access List Remarks

You can include comments or remarks about entries in any IP access list. An access list remark is an optional remark before or after an access list entry that describes the entry so that you do not have to interpret the purpose of the entry. Each remark is limited to 100 characters in length.

The remark can go before or after a **permit** or **deny** statement. Be consistent about where you add remarks. Users may be confused if some remarks precede the associated **permit** or **deny** statements and some remarks follow the associated statements.

The following is an example of a remark that describes function of the subsequent **deny** statement:

```
ip access-list extended telnetting
remark Do not allow host1 subnet to telnet out
deny tcp host 172.16.2.88 any eq telnet
```

# How to Configure Commented IP Access List Entries

## Writing Remarks in a Named or Numbered Access List

You can use a named or numbered access list configuration. You must apply the access list to an interface or terminal line after the access list is created for the configuration to work.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list {standard | extended} {*name* | *number*}**
4. **remark *remark***
5. **deny *protocol* *host* *host-address* any eq *port***
6. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list {standard   extended} {<i>name</i>   <i>number</i>}</b>  <b>Example:</b> Device(config)# ip access-list extended telnetting	Identifies the access list by a name or number and enters extended named access list configuration mode.
<b>Step 4</b>	<b>remark <i>remark</i></b>  <b>Example:</b> Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out	Adds a remark for an entry in a named IP access list.  <ul style="list-style-type: none"> <li>• The remark indicates the purpose of the <b>permit</b> or <b>deny</b> statement.</li> </ul>
<b>Step 5</b>	<b>deny <i>protocol</i> <i>host</i> <i>host-address</i> any eq <i>port</i></b>  <b>Example:</b> Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet	Sets conditions in a named IP access list that denies packets.

	Command or Action	Purpose
<b>Step 6</b>	<b>end</b>  <b>Example:</b> Device(config-ext-nacl)# end	Exits extended named access list configuration mode and enters privileged EXEC mode.

## Configuration Examples for Commented IP Access List Entries

### Example: Writing Remarks in an IP Access List

```

Device# configure terminal
Device(config)# ip access-list extended telnetting
Device(config-ext-nacl)# remark Do not allow host1 subnet to telnet out
Device(config-ext-nacl)# deny tcp host 172.16.2.88 any eq telnet
Device(config-ext-nacl)# end

```

## Additional References for Commented IP Access List Entries

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>



### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Commented IP Access List Entries

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 8: Feature Information for Commented IP Access List Entries**

Feature Name	Releases	Feature Information
Commented IP Access List Entries	Cisco IOS XE Release 2.1	The Commented IP Access List Entries feature allows you to include comments or remarks about <b>deny</b> or <b>permit</b> conditions in any IP access list. These remarks make access lists easier for network administrators to understand. Each remark is limited to 100 characters in length.  The following command was introduced or modified: <b>remark</b> .





## Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list logs an information message about the packet at the device console.

This module provides information about standard IP access list logging.

- [Finding Feature Information, page 87](#)
- [Restrictions for Standard IP Access List Logging, page 87](#)
- [Information About Standard IP Access List Logging, page 88](#)
- [How to Configure Standard IP Access List Logging, page 88](#)
- [Configuration Examples for Standard IP Access List Logging, page 91](#)
- [Additional References for Standard IP Access List Logging, page 91](#)
- [Feature Information for Standard IP Access List Logging, page 92](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Restrictions for Standard IP Access List Logging

IP access list logging is supported only for routed interfaces or router access control lists (ACLs).

# Information About Standard IP Access List Logging

## Standard IP Access List Logging

The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list causes an information log message about the packet to be sent to the device console. The log level of messages that are printed to the device console is controlled by the **logging console** command.

The first packet that the access list inspects triggers the access list to log a message at the device console. Subsequent packets are collected over 5-minute intervals before they are displayed or logged. Log messages include information about the access list number, the source IP address of packets, the number of packets from the same source that were permitted or denied in the previous 5-minute interval, and whether a packet was permitted or denied. You can also monitor the number of packets that are permitted or denied by a particular access list, including the source address of each packet.

## How to Configure Standard IP Access List Logging

### Creating a Standard IP Access List Using Numbers

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* {deny | permit} *host address* [log]
4. **access-list** *access-list-number* {deny | permit} **any** [log]
5. **interface** *type number*
6. **ip access-group** *access-list-number* {in | out}
7. **end**

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.

	Command or Action	Purpose
<b>Step 3</b>	<b>access-list</b> <i>access-list-number</i> {deny   permit} <b>host</b> <i>address</i> [log]  <b>Example:</b> Device(config)# access-list 1 permit host 10.1.1.1 log	Defines a standard numbered IP access list using a source address and wildcard, and configures the logging of informational messages about packets that match the access list entry at the device console.
<b>Step 4</b>	<b>access-list</b> <i>access-list-number</i> {deny   permit} <b>any</b> [log]  <b>Example:</b> Device(config)# access-list 1 permit any log	Defines a standard numbered IP access list by using an abbreviation for the source and source mask 0.0.0.0 255.255.255.255.
<b>Step 5</b>	<b>interface</b> <i>type number</i>  <b>Example:</b> Device(config)# interface gigabitethernet 0/0/0	Configures an interface and enters interface configuration mode.
<b>Step 6</b>	<b>ip access-group</b> <i>access-list-number</i> {in   out}  <b>Example:</b> Device(config-if)# ip access-group 1 in	Applies the specified numbered access list to the incoming or outgoing interface. <ul style="list-style-type: none"> <li>When you filter based on source addresses, you typically apply the access list to an incoming interface.</li> </ul>
<b>Step 7</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and enters privileged EXEC mode.

## Creating a Standard IP Access List Using Names

### SUMMARY STEPS

1. enable
2. configure terminal
3. ip access-list standard *name*
4. {deny | permit} {host *address* | any} log
5. exit
6. interface *type number*
7. ip access-group *access-list-name* {in | out}
8. end

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list standard <i>name</i></b>  <b>Example:</b> Device(config)# ip access-list standard acl1	Defines a standard IP access list and enters standard named access list configuration mode.
<b>Step 4</b>	<b>{deny   permit} {host <i>address</i>   any} log</b>  <b>Example:</b> Device(config-std-nacl)# permit host 10.1.1.1 log	Sets conditions in a named IP access list that will deny packets from entering a network or permit packets to enter a network, and configures the logging of informational messages about packets that match the access list entry at the device console.
<b>Step 5</b>	<b>exit</b>  <b>Example:</b> Device(config-std-nacl)# exit	Exits standard named access list configuration mode and enters global configuration mode.
<b>Step 6</b>	<b>interface <i>type number</i></b>  <b>Example:</b> Device(config)# interface gigabitethernet 0/0/0	Configures an interface and enters interface configuration mode.
<b>Step 7</b>	<b>ip access-group <i>access-list-name</i> {in   out}</b>  <b>Example:</b> Device(config-if)# ip access-group acl1 in	Applies the specified access list to the incoming or outgoing interface.  <ul style="list-style-type: none"> <li>• When you filter based on source addresses, you typically apply the access list to an incoming interface.</li> </ul>
<b>Step 8</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	Exits interface configuration mode and enters privileged EXEC mode.

# Configuration Examples for Standard IP Access List Logging

## Example: Creating a Standard IP Access List Using Numbers

```
Device# configure terminal
Device(config)# access-list 1 permit host 10.1.1.1 log
Device(config)# access-list 1 permit any log
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip access-group 1 in
```

## Example: Creating a Standard IP Access List Using Names

```
Device# configure terminal
Device(config)# ip access-list standard acl1
Device(config-std-nacl)# permit host 10.1.1.1 log
Device(config-std-nacl)# exit
Device(config)# interface gigabitethernet 0/0/0
Device(config-if)# ip access-group acl1 in
```

## Example: Limiting Debug Output

The following sample configuration uses an access list to limit the **debug** command output. Limiting the **debug** output restricts the volume of data to what you are interested in, saving you time and resources.

```
Device(config)# ip access-list acl1
Device(config-std-nacl)# remark Displays only advertisements for LDP peer in acl1
Device(config-std-nacl)# permit host 10.0.0.44

Device# debug mpls ldp advertisements peer-acl acl1

tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.17.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.16.0.31
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 172.22.0.33
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.1
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.0.3
tagcon: peer 10.0.0.44:0 (pp 0x60E105BC): advertise 192.168.1.33
```

# Additional References for Standard IP Access List Logging

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>

Related Topic	Document Title
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>

### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Standard IP Access List Logging

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 9: Feature Information for Standard IP Access List Logging**

Feature Name	Releases	Feature Information
Standard IP Access List Logging	Cisco IOS XE Release 2.1	The Standard IP Access List Logging feature provides the ability to log messages about packets that are permitted or denied by a standard IP access list. Any packet that matches the access list logs an information message about the packet at the device console.





## IP Access List Entry Sequence Numbering

The IP Access List Entry Sequence Numbering feature allows you to apply sequence numbers to **permit** or **deny** statements as well as reorder, add, or remove such statements from a named IP access list. The IP Access List Entry Sequence Numbering feature makes revising IP access lists much easier. Prior to this feature, you could add access list entries to the end of an access list only; therefore, needing to add statements anywhere except at the end of a named IP access list required reconfiguring the entire access list.

- [Finding Feature Information, page 93](#)
- [Restrictions for IP Access List Entry Sequence Numbering, page 93](#)
- [Information About IP Access List Entry Sequence Numbering, page 94](#)
- [How to Use Sequence Numbers in an IP Access List, page 98](#)
- [Configuration Examples for IP Access List Entry Sequence Numbering, page 100](#)
- [Additional References, page 102](#)
- [Feature Information for IP Access List Entry Sequence Numbering, page 103](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Restrictions for IP Access List Entry Sequence Numbering

- This feature does not support dynamic, reflexive, or firewall access lists.
- This feature does not support old-style numbered access lists, which existed before named access lists. Keep in mind that you can name an access list with a number, so numbers are allowed when they are entered in the standard or extended named access list (NACL) configuration mode.

# Information About IP Access List Entry Sequence Numbering

## Purpose of IP Access Lists

Access lists perform packet filtering to control which packets move through the network and where. Such control can help limit network traffic and restrict the access of users and devices to the network. Access lists have many uses, and therefore many commands accept a reference to an access list in their command syntax. Access lists can be used to do the following:

- Filter incoming packets on an interface.
- Filter outgoing packets on an interface.
- Restrict the contents of routing updates.
- Limit debug output based on an address or protocol.
- Control virtual terminal line access.
- Identify or classify traffic for advanced features, such as congestion avoidance, congestion management, and priority and custom queuing.
- Trigger dial-on-demand routing (DDR) calls.

## How an IP Access List Works

An access list is a sequential list consisting of a permit statement and a deny statement that apply to IP addresses and possibly upper-layer IP protocols. The access list has a name by which it is referenced. Many software commands accept an access list as part of their syntax.

An access list can be configured and named, but it is not in effect until the access list is referenced by a command that accepts an access list. Multiple commands can reference the same access list. An access list can control traffic arriving at the device or leaving the device, but not traffic originating at the device.

## IP Access List Process and Rules

- The software tests the source or destination address or the protocol of each packet being filtered against the conditions in the access list, one condition (**permit** or **deny** statement) at a time.
- If a packet does not match an access list statement, the packet is then tested against the next statement in the list.
- If a packet and an access list statement match, the rest of the statements in the list are skipped and the packet is permitted or denied as specified in the matched statement. The first entry that the packet matches determines whether the software permits or denies the packet. That is, after the first match, no subsequent entries are considered.
- If the access list denies the address or protocol, the software discards the packet and returns an Internet Control Message Protocol (ICMP) Host Unreachable message.

- If no conditions match, the packet is dropped. This is because each access list ends with an unwritten or implicit **deny** statement. That is, if the packet has not been permitted by the time it was tested against each statement, it is denied.
- Because the software stops testing conditions after the first match, the order of the conditions is critical. The same **permit** or **deny** statements specified in a different order could result in a packet being passed under one circumstance and denied in another circumstance.
- If an access list is referenced by name in a command, but the access list does not exist, all packets pass.
- Only one access list per interface, per protocol, per direction is allowed.
- Inbound access lists process packets arriving at the device. Incoming packets are processed before being routed to an outbound interface. An inbound access list is efficient because it saves the overhead of routing lookups if the packet is to be discarded because it is denied by the filtering tests. If the packet is permitted by the tests, it is then processed for routing. For inbound lists, **permit** means continue to process the packet after receiving it on an inbound interface; **deny** means discard the packet.
- Outbound access lists process packets before they leave the device. Incoming packets are routed to the outbound interface and then processed through the outbound access list. For outbound lists, **permit** means send it to the output buffer; **deny** means discard the packet.

## 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.
- Because the software stops testing conditions after it encounters the first match (to either a **permit** or **deny** statement), you will reduce processing time and resources if you put the statements that packets are most likely to match at the beginning of the access list. Place more frequently occurring conditions before less frequent conditions.
- Organize your access list so that more specific references in a network or subnet appear before more general ones.
- 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.

## Source and Destination Addresses

Source and destination address fields in an IP packet are two typical fields on which to base an access list. Specify source addresses to control the packets being sent from certain networking devices or hosts. Specify destination addresses to control the packets being sent to certain networking devices or hosts.

## Wildcard Mask and Implicit Wildcard Mask

When comparing the address bits in an access list entry to a packet being submitted to the access list, address filtering uses wildcard masking to determine whether to check or ignore the corresponding IP address bits. By carefully setting wildcard masks, an administrator can select 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.
- A wildcard mask bit 1 means ignore that corresponding bit value.

If you do not supply a wildcard mask with a source or destination address in an access list statement, the software assumes a default wildcard mask of 0.0.0.0.

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

## Transport Layer Information

You can filter packets based on transport layer information, such as whether the packet is a TCP, UDP, Internet Control Message Protocol (ICMP) or Internet Group Management Protocol (IGMP) packet.

## Benefits IP Access List Entry Sequence Numbering

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 (statement) in the middle of an existing list, all of the entries *after* the desired position had to be removed. Then, once you added the new entry, you needed to reenter all of the entries you removed earlier. This method was cumbersome and error prone.

The IP Access List Entry Sequence Numbering feature allows you to add sequence numbers to access list entries and resequence them. When you add a new entry, you can choose the sequence number so that the entry is in a desired position in the access list. If necessary, entries currently in the access list can be resequenced (reordered) to create room to insert the new entry.

## Sequence Numbering Behavior

- For backward compatibility with previous releases, if entries with no sequence numbers are applied, the first entry is assigned a sequence number of 10, and successive entries are incremented by 10. The maximum sequence number is 2147483647. If the generated sequence number exceeds this maximum number, the following message is displayed:

`Exceeded maximum sequence number.`

- If you enter an entry without a sequence number, it is assigned a sequence number that is 10 greater than the last sequence number in that access list and is placed at the end of the list.
- If you enter an entry that matches an already existing entry (except for the sequence number), then no changes are made.
- If you enter a sequence number that is already present, the following error message is generated:

`Duplicate sequence number.`

- If a new access list is entered from global configuration mode, then sequence numbers for that access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the Route Processor (RP) and line card (LC) are always synchronized.
- Sequence numbers are not nvgened. That is, the sequence numbers themselves are not saved. In the event that the system is reloaded, the configured sequence numbers revert to the default sequence starting number and increment from that number. The function is provided for backward compatibility with software releases that do not support sequence numbering.
- The IP Access List Entry Sequence Numbering feature works with named standard and extended IP access lists. Because the name of an access list can be designated as a number, numbers are acceptable.

# How to Use Sequence Numbers in an IP Access List

## Sequencing Access-List Entries and Revising the Access List

This task shows how to assign sequence numbers to entries in a named IP access list and how to add or delete an entry to or from an access list. When completing this task, keep the following points in mind:

- Resequencing the access list entries is optional. The resequencing step in this task is shown as required because that is one purpose of this feature and this task demonstrates that functionality.
- In the following procedure, the **permit** command is shown in Step 5 and the **deny** command is shown in Step 6. However, that order can be reversed. Use the order that suits the need of your configuration.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence** *access-list-name starting-sequence-number increment*
4. **ip access-list {standard| extended}** *access-list-name*
5. Do one of the following:
  - *sequence-number permit source source-wildcard*
  - *sequence-number permit protocol source source-wildcard destination destination-wildcard [precedence precedence][tos tos] [log] [time-range time-range-name] [fragments]*
6. Do one of the following:
  - *sequence-number deny source source-wildcard*
  - *sequence-number deny protocol source source-wildcard destination destination-wildcard [precedence precedence][tos tos] [log] [time-range time-range-name] [fragments]*
7. Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.
8. **end**
9. **show ip access-lists** *access-list-name*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. Enter your password if prompted.

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list resequence</b> <i>access-list-name</i> <i>starting-sequence-number increment</i>  <b>Example:</b> Device(config)# ip access-list resequence kmd1 100 15	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.
<b>Step 4</b>	<b>ip access-list {standard  extended}</b> <i>access-list-name</i>  <b>Example:</b> Device(config)# ip access-list standard kmd1	Specifies the IP access list by name and enters named access list configuration mode. <ul style="list-style-type: none"> <li>If you specify <b>standard</b>, make sure you subsequently specify <b>permit</b> and/or <b>deny</b> statements using the standard access list syntax.</li> <li>If you specify <b>extended</b>, make sure you subsequently specify <b>permit</b> and/or <b>deny</b> statements using the extended access list syntax.</li> </ul>
<b>Step 5</b>	Do one of the following: <ul style="list-style-type: none"> <li><i>sequence-number</i> <b>permit</b> <i>source</i> <i>source-wildcard</i></li> <li><i>sequence-number</i> <b>permit</b> <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [<b>precedence</b> <i>precedence</i>][<b>tos</b> <i>tos</i>] [<b>log</b>] [<b>time-range</b> <i>time-range-name</i>] [<b>fragments</b>]</li> </ul> <b>Example:</b> Device(config-std-nacl)# 105 permit 10.5.5.5 0.0.0 255	Specifies a permit statement in named IP access list mode. <ul style="list-style-type: none"> <li>This access list happens to use a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>As the prompt indicates, this access list was a standard access list. If you had specified <b>extended</b> in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended <b>permit</b> command syntax.</li> </ul>
<b>Step 6</b>	Do one of the following: <ul style="list-style-type: none"> <li><i>sequence-number</i> <b>deny</b> <i>source</i> <i>source-wildcard</i></li> <li><i>sequence-number</i> <b>deny</b> <i>protocol</i> <i>source</i> <i>source-wildcard</i> <i>destination</i> <i>destination-wildcard</i> [<b>precedence</b> <i>precedence</i>][<b>tos</b> <i>tos</i>] [<b>log</b>] [<b>time-range</b> <i>time-range-name</i>] [<b>fragments</b>]</li> </ul>	(Optional) Specifies a deny statement in named IP access list mode. <ul style="list-style-type: none"> <li>This access list uses a <b>permit</b> statement first, but a <b>deny</b> statement could appear first, depending on the order of statements you need.</li> <li>As the prompt indicates, this access list was a standard access list. If you had specified <b>extended</b> in Step 4, the prompt for this step would be Device(config-ext-nacl) and you would use the extended <b>deny</b> command syntax.</li> </ul>

	Command or Action	Purpose
	<b>Example:</b>  Device(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255	
<b>Step 7</b>	Repeat Step 5 and/or Step 6 to add sequence number statements, as applicable.	Allows you to revise the access list.
<b>Step 8</b>	<b>end</b>  <b>Example:</b>  Device(config-std-nacl)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.
<b>Step 9</b>	<b>show ip access-lists</b> <i>access-list-name</i>  <b>Example:</b>  Device# show ip access-lists kmd1	(Optional) Displays the contents of the IP access list.

### Examples

Review the output of the **show ip access-lists** command to see that the access list includes the new entries:

```
Device# show ip access-lists kmd1

Standard IP access list kmd1
100 permit 10.4.4.0, wildcard bits 0.0.0.255
105 permit 10.5.5.0, wildcard bits 0.0.0.255
115 permit 10.0.0.0, wildcard bits 0.0.0.255
130 permit 10.5.5.0, wildcard bits 0.0.0.255
145 permit 10.0.0.0, wildcard bits 0.0.0.255
```

## Configuration Examples for IP Access List Entry Sequence Numbering

### Example: Resequencing Entries in an Access List

The following example shows access list resequencing. The starting value is 1, and increment value is 2. The subsequent entries are ordered based on the increment values specified, and the range is from 1 to 2147483647.

When an entry with no sequence number is entered, by default the entry has a sequence number of 10 more than the last entry in the access list.

```
Device# show access-list 150

Extended IP access list 150
```



```

10 permit ip host 10.3.3.3 host 172.16.5.34
20 permit icmp any any
30 permit tcp any host 10.3.3.3
40 permit ip host 10.4.4.4 any
50 Dynamic test permit ip any any
60 permit ip host 172.16.2.2 host 10.3.3.12
70 permit ip host 10.3.3.3 any log
80 permit tcp host 10.3.3.3 host 10.1.2.2
90 permit ip host 10.3.3.3 any
100 permit ip any any

Device(config)# ip access-list extended 150
Device(config)# ip access-list resequence 150 1 2
Device(config)# exit

Device# show access-list 150

Extended IP access list 150
 1 permit ip host 10.3.3.3 host 172.16.5.34
 3 permit icmp any any
 5 permit tcp any host 10.3.3.3
 7 permit ip host 10.4.4.4 any
 9 Dynamic test permit ip any any
11 permit ip host 172.16.2.2 host 10.3.3.12
13 permit ip host 10.3.3.3 any log
15 permit tcp host 10.3.3.3 host 10.1.2.2
17 permit ip host 10.3.3.3 any
19 permit ip any any

```

## Example: Adding Entries with Sequence Numbers

In the following example, a new entry is added to a specified access list:

```

Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.4.2, wildcard bits 0.0.255.255
5 permit 10.0.0.44, wildcard bits 0.0.0.255
10 permit 10.0.0.1, wildcard bits 0.0.0.255
20 permit 10.0.0.2, wildcard bits 0.0.0.255

Device(config)# ip access-list standard tryon
Device(config-std-nacl)# 15 permit 10.5.5.5 0.0.0.255
Device(config-std-nacl)# exit
Device(config)# exit
Device# show ip access-list

Standard IP access list tryon
2 permit 10.4.0.0, wildcard bits 0.0.255.255
5 permit 10.0.0.0, wildcard bits 0.0.0.255
10 permit 10.0.0.0, wildcard bits 0.0.0.255
15 permit 10.5.5.0, wildcard bits 0.0.0.255
20 permit 10.0.0.0, wildcard bits 0.0.0.255

```

## Example: Entry Without Sequence Number

The following example shows how an entry with no specified sequence number is added to the end of an access list. When an entry is added without a sequence number, it is automatically given a sequence number that puts it at the end of the access list. Because the default increment is 10, the entry will have a sequence number 10 higher than the last entry in the existing access list.

```

Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.1.1.1 0.0.0.255

```

```

Device(config-std-nacl)# permit 10.2.2.2 0.0.0.255
Device(config-std-nacl)# permit 10.3.3.3 0.0.0.255
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255

Device(config)# ip access-list standard 1
Device(config-std-nacl)# permit 10.4.4.4 0.0.0.255
Device(config-std-nacl)# end
Device(config-std-nacl)## exit
Device# show access-list

Standard IP access list 1
10 permit 0.0.0.0, wildcard bits 0.0.0.255
20 permit 0.0.0.0, wildcard bits 0.0.0.255
30 permit 0.0.0.0, wildcard bits 0.0.0.255
40 permit 0.0.0.0, wildcard bits 0.0.0.255

```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
IP access list commands	<i>Cisco IOS Security Command Reference</i>
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

# Feature Information for IP Access List Entry Sequence Numbering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 10: Feature Information for IP Access List Entry Sequence Numbering**

Feature Name	Releases	Feature Information
IP Access List Entry Sequence Numbering	12.2(14)S 12.2(15)T 12.3(2)T	Users can apply sequence numbers to <b>permit</b> or <b>deny</b> statements and also reorder, add, or remove such statements from a named IP access list. This feature makes revising IP access lists much easier. Prior to this feature, users could add access list entries to the end of an access list only; therefore needing to add statements anywhere except the end required reconfiguring the access list entirely.  The following commands were introduced or modified: <b>deny (IP)</b> , <b>ip access-list resequence deny (IP)</b> , <b>permit (IP)</b> .
IP Access List Entry—Persistent Sequence Numbering Across Reloads	12.2(33)SXJ6 15.0(1)SY5 15.4(2)S	The following command was introduced: <b>ip access-list persistent</b> .
IP Access List Entry—Persistent Sequence Numbering Across Reloads	15.0(1)SY5	The following command was introduced: <b>ip access-list persistent</b> .





## Configuring Lock-and-Key Security (Dynamic Access Lists)

### Feature History

Release	Modification
Cisco IOS	For information about feature support in Cisco IOS software, use Cisco Feature Navigator.

This chapter describes how to configure lock-and-key security at your router. Lock-and-key is a traffic filtering security feature available for the IP protocol.

For a complete description of lock-and-key commands, refer to the *Cisco IOS Security Command Reference*. To locate documentation of other commands that appear in this chapter, use the command reference master index or search online.

To identify the hardware platform or software image information associated with a feature, use the Feature Navigator on Cisco.com to search for information about the feature or refer to the software release notes for a specific release.

- [Prerequisites for Configuring Lock-and-Key, page 105](#)
- [Information About Configuring Lock-and-Key Security \(Dynamic Access Lists\), page 106](#)
- [How to Configure Lock-and-Key Security \(Dynamic Access Lists\), page 111](#)
- [Configuration Examples for Lock-and-Key, page 114](#)

## Prerequisites for Configuring Lock-and-Key

Lock-and-key uses IP extended access lists. You must have a solid understanding of how access lists are used to filter traffic, before you attempt to configure lock-and-key. Access lists are described in the chapter “Access Control Lists: Overview and Guidelines.”

Lock-and-key employs user authentication and authorization as implemented in Cisco’s authentication, authorization, and accounting (AAA) paradigm. You must understand how to configure AAA user authentication

and authorization before you configure lock-and-key. User authentication and authorization is explained in the “Authentication, Authorization, and Accounting (AAA)” part of this document.

Lock-and-key uses the **autocommand** command, which you should understand. This command is described in the *Cisco IOS Terminal Services Command Reference*.

## Information About Configuring Lock-and-Key Security (Dynamic Access Lists)

### About Lock-and-Key

Lock-and-key is a traffic filtering security feature that dynamically filters IP protocol traffic. Lock-and-key is configured using IP dynamic extended access lists. Lock-and-key can be used in conjunction with other standard access lists and static extended access lists.

When lock-and-key is configured, designated users whose IP traffic is normally blocked at a router can gain temporary access through the router. When triggered, lock-and-key reconfigures the interface's existing IP access list to permit designated users to reach their designated host(s). Afterwards, lock-and-key reconfigures the interface back to its original state.

For a user to gain access to a host through a router with lock-and-key configured, the user must first open a Telnet session to the router. When a user initiates a standard Telnet session to the router, lock-and-key automatically attempts to authenticate the user. If the user is authenticated, they will then gain temporary access through the router and be able to reach their destination host.

### Benefits of Lock-and-Key

Lock-and-key provides the same benefits as standard and static extended access lists (these benefits are discussed in the chapter “Access Control Lists: Overview and Guidelines”). However, lock-and-key also has the following security benefits over standard and static extended access lists:

- Lock-and-key uses a challenge mechanism to authenticate individual users.
- Lock-and-key provides simpler management in large internetworks.
- In many cases, lock-and-key reduces the amount of router processing required for access lists.
- Lock-and-key reduces the opportunity for network break-ins by network hackers.

With lock-and-key, you can specify which users are permitted access to which source and destination hosts. These users must pass a user authentication process before they are permitted access to their designated hosts. Lock-and-key creates dynamic user access through a firewall, without compromising other configured security restrictions.

### When to Use Lock-and-Key

Two examples of when you might use lock-and-key follow:

- When you want a specific remote user (or group of remote users) to be able to access a host within your network, connecting from their remote hosts via the Internet. Lock-and-key authenticates the user, then permits limited access through your firewall router for the individual's host or subnet, for a finite period of time.
- When you want a subset of hosts on a local network to access a host on a remote network protected by a firewall. With lock-and-key, you can enable access to the remote host only for the desired set of local user's hosts. Lock-and-key require the users to authenticate through a TACACS+ server, or other security server, before allowing their hosts to access the remote hosts.

## How Lock-and-Key Works

The following process describes the lock-and-key access operation:

- 1 A user opens a Telnet session to a border (firewall) router configured for lock-and-key. The user connects via the virtual terminal port on the router.
- 2 The Cisco IOS software receives the Telnet packet, opens a Telnet session, prompts for a password, and performs a user authentication process. The user must pass authentication before access through the router is allowed. The authentication process can be done by the router or by a central access security server such as a TACACS+ or RADIUS server.
- 3 When the user passes authentication, they are logged out of the Telnet session, and the software creates a temporary entry in the dynamic access list. (Per your configuration, this temporary entry can limit the range of networks to which the user is given temporary access.)
- 4 The user exchanges data through the firewall.
- 5 The software deletes the temporary access list entry when a configured timeout is reached, or when the system administrator manually clears it. The configured timeout can either be an idle timeout or an absolute timeout.

**Note**

The temporary access list entry is not automatically deleted when the user terminates a session. The temporary access list entry remains until a configured timeout is reached or until it is cleared by the system administrator.

## Compatibility with Releases Before Cisco IOS Release 11.1

Enhancements to the **access-list** command are used for lock-and-key. These enhancements are backward compatible--if you migrate from a release before Cisco IOS Release 11.1 to a newer release, your access lists will be automatically converted to reflect the enhancements. However, if you try to use lock-and-key with a release before Cisco IOS Release 11.1, you might encounter problems as described in the following caution paragraph:

**Caution**

Cisco IOS releases before Release 11.1 are not upwardly compatible with the lock-and-key access list enhancements. Therefore, if you save an access list with software older than Release 11.1, and then use this software, the resulting access list will not be interpreted correctly. This could cause you severe security problems. You must save your old configuration files with Cisco IOS Release 11.1 or later software before booting an image with these files.

## Risk of Spoofing with Lock-and-Key

**Caution**

Lock-and-key access allows an external event (a Telnet session) to place an opening in the firewall. While this opening exists, the router is susceptible to source address spoofing.

When lock-and-key is triggered, it creates a dynamic opening in the firewall by temporarily reconfiguring an interface to allow user access. While this opening exists, another host might spoof the authenticated user's address to gain access behind the firewall. Lock-and-key does not cause the address spoofing problem; the problem is only identified here as a concern to the user. Spoofing is a problem inherent to all access lists, and lock-and-key does not specifically address this problem.

To prevent spoofing, configure encryption so that traffic from the remote host is encrypted at a secured remote router, and decrypted locally at the router interface providing lock-and-key. You want to ensure that all traffic using lock-and-key will be encrypted when entering the router; this way no hackers can spoof the source address, because they will be unable to duplicate the encryption or to be authenticated as is a required part of the encryption setup process.

## Router Performance Impacts with Lock-and-Key

When lock-and-key is configured, router performance can be affected in the following ways:

- When lock-and-key is triggered, the dynamic access list forces an access list rebuild on the silicon switching engine (SSE). This causes the SSE switching path to slow down momentarily.
- Dynamic access lists require the idle timeout facility (even if the timeout is left to default) and therefore cannot be SSE switched. These entries must be handled in the protocol fast-switching path.
- When remote users trigger lock-and-key at a border router, additional access list entries are created on the border router interface. The interface's access list will grow and shrink dynamically. Entries are dynamically removed from the list after either the idle-timeout or max-timeout period expires. Large access lists can degrade packet switching performance, so if you notice performance problems, you should look at the border router configuration to see if you should remove temporary access list entries generated by lock-and-key.

## Maintaining Lock-and-Key

When lock-and-key is in use, dynamic access lists will dynamically grow and shrink as entries are added and deleted. You need to make sure that entries are being deleted in a timely way, because while entries exist, the



risk of a spoofing attack is present. Also, the more entries there are, the bigger the router performance impact will be.

If you do not have an idle or absolute timeout configured, entries will remain in the dynamic access list until you manually remove them. If this is the case, make sure that you are extremely vigilant about removing entries.

## Dynamic Access Lists

Use the following guidelines for configuring dynamic access lists:

- Do not create more than one dynamic access list for any one access list. The software only refers to the first dynamic access list defined.
- Do not assign the same *dynamic-name* to another access list. Doing so instructs the software to reuse the existing list. All named entries must be globally unique within the configuration.
- Assign attributes to the dynamic access list in the same way you assign attributes for a static access list. The temporary access list entries inherit the attributes assigned to this list.
- Configure Telnet as the protocol so that users must open a Telnet session into the router to be authenticated before they can gain access through the router.
- Either define an idle timeout now with the **timeout** keyword in the **access-enable** command in the **autocommand** command, or define an absolute timeout value later with the **access-list** command. You must define either an idle timeout or an absolute timeout--otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated their session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)
- If you configure an idle timeout, the idle timeout value should be equal to the WAN idle timeout value.
- If you configure both idle and absolute timeouts, the idle timeout value must be less than the absolute timeout value.
- If you realize that a job will run past the ACL's absolute timer, use the **access-list dynamic-extend** command to extend the absolute timer of the dynamic ACL by six minutes. This command allows you to open a new Telnet session into the router to re-authentication yourself using lock-and-key.
- The only values replaced in the temporary entry are the source or destination address, depending whether the access list was in the input access list or output access list. All other attributes, such as port, are inherited from the main dynamic access list.
- Each addition to the dynamic list is always put at the beginning of the dynamic list. You cannot specify the order of temporary access list entries.
- Temporary access list entries are never written to NVRAM.
- To manually clear or to display dynamic access lists, refer to the section "Maintaining Lock-and-Key" later in this chapter.

## Lock-and-Key Authentication

There are three possible methods to configure an authentication query process. These three methods are described in this section.

**Note**

Cisco recommends that you use the TACACS+ server for your authentication query process. TACACS+ provides authentication, authorization, and accounting services. It also provides protocol support, protocol specification, and a centralized security database. Using a TACACS+ server is described in the next section, “Method 1--Configuring a Security Server.”

Use a network access security server such as TACACS+ server. This method requires additional configuration steps on the TACACS+ server but allows for stricter authentication queries and more sophisticated tracking capabilities.

```
Router(config-line) # login tacacs
```

Use the **username** command. This method is more effective because authentication is determined on a user basis.

```
Router(config) # username
```

```
name
{ nopassword
|
password
{
mutual-password
|
encryption-type
encryption-password
}}
}
```

Use the **password** and **login** commands. This method is less effective because the password is configured for the port, not for the user. Therefore, any user who knows the password can authenticate successfully.

```
R
outer(config-line) # password

password
Router(config-line) # login local
```

## The autocommand Command

The **autocommand** command configures the system to automatically execute a specified privileged EXEC command when a user connects to a particular line. Use the following guidelines for configuring the **autocommand** command:

- If you use a TACACS+ server to authenticate the user, you should configure the **autocommand** command on the TACACS+ server as a per-user autocommand. If you use local authentication, use the **autocommand** command on the line.
- Configure all virtual terminal (VTY) ports with the same **autocommand** command. Omitting an **autocommand** command on a VTY port allows a random host to gain privileged EXEC mode access to the router and does not create a temporary access list entry in the dynamic access list.
- If you do not define an idle timeout with the **autocommand access-enable** command, you must define an absolute timeout with the **access-list** command. You must define either an idle timeout or an absolute timeout--otherwise, the temporary access list entry will remain configured indefinitely on the interface (even after the user has terminated the session) until the entry is removed manually by an administrator. (You could configure both idle and absolute timeouts if you wish.)

- If you configure both idle and absolute timeouts, the absolute timeout value must be greater than the idle timeout value.

# How to Configure Lock-and-Key Security (Dynamic Access Lists)

## Configuring Lock-and-Key

To configure lock-and-key, use the following commands beginning in global configuration mode. While completing these steps, be sure to follow the guidelines listed in the “Lock-and-Key Configuration Guidelines” section of this chapter.

### SUMMARY STEPS

1. Router(config)# **access-list** *access-list-number* [**dynamic** *dynamic-name* [**timeout** *minutes*]] {**deny** | **permit**} **telnet** *source source-wildcard destination destination-wildcard* [**precedence** *precedence*] [**tos** *tos*] [**established**] [**log**]
2. Router(config)# **access-list dynamic-extend**
3. Router(config)# **interface** *type number*
4. Router(config-if)# **ip access-group** *access-list-number*
5. Router(config-if)# **exit**
6. Router(config)# **line vty** *line-number* [*ending-line-number*]
7. Do one of the following:
  - Router(config-line)# **login tacacs**
  - Router(config-line)# **password** *password*
8. Do one of the following:
  - Router(config-line)# **autocommand access-enable** [**host**] [**timeout** *minutes*]
  - Router# **access-enable** [**host**] [**timeout** *minutes*]

### DETAILED STEPS

	Command or Action	Purpose
Step 1	Router(config)# <b>access-list</b> <i>access-list-number</i> [ <b>dynamic</b> <i>dynamic-name</i> [ <b>timeout</b> <i>minutes</i> ]] { <b>deny</b>   <b>permit</b> } <b>telnet</b> <i>source source-wildcard</i>	Configures a dynamic access list, which serves as a template and placeholder for temporary access list entries.

	Command or Action	Purpose
	<i>destination destination-wildcard</i> [precedence precedence] [tos tos] [established] [log]	
<b>Step 2</b>	Router(config)# <b>access-list dynamic-extend</b>	(Optional) Extends the absolute timer of the dynamic ACL by six minutes when you open another Telnet session into the router to re-authenticate yourself using lock-and-key. Use this command if your job will run past the ACL's absolute timer.
<b>Step 3</b>	Router(config)# <b>interface</b> type number	Configures an interface and enters interface configuration mode.
<b>Step 4</b>	Router(config-if)# <b>ip access-group</b> access-list-number	Applies the access list to the interface.
<b>Step 5</b>	Router(config-if)# <b>exit</b>	Exits interface configuration mode and enters global configuration mode.
<b>Step 6</b>	Router(config)# <b>line vty</b> line-number [ending-line-number]	Defines one or more virtual terminal (VTY) ports and enters line configuration mode. If you specify multiple VTY ports, they must all be configured identically because the software hunts for available VTY ports on a round-robin basis. If you do not want to configure all your VTY ports for lock-and-key access, you can specify a group of VTY ports for lock-and-key support only.
<b>Step 7</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>Router(config-line)# <b>login tacacs</b></li> <li>.</li> <li>Router(config-line)# <b>password</b> password</li> </ul> <p><b>Example:</b></p> <pre>Router (config-line) # login local</pre> <p><b>Example:</b></p> <pre>Router (config-line) # exit</pre> <p><b>Example:</b></p> <pre>then</pre> <p><b>Example:</b></p> <pre>Router (config) # username name password secret</pre>	Configures user authentication in line or global configuration mode.

	Command or Action	Purpose
<b>Step 8</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>Router(config-line)# <b>autocommand access-enable</b> [host] [timeout <i>minutes</i>]</li> <li>•</li> <li>Router# <b>access-enable</b> [host] [timeout <i>minutes</i>]</li> </ul>	<p>Enables the creation of temporary access list entries in line configuration or privilege EXEC mode.</p> <p>Using the <b>autocommand</b> with the <b>access-enable</b> command in line configuration mode configures the system to automatically create a temporary access list entry in the dynamic access list when the host connects to the line (or lines).</p> <p>If the optional <b>host</b> keyword is not specified, all hosts on the entire network are allowed to set up a temporary access list entry. The dynamic access list contains the network mask to enable the new network connection.</p> <p>If the optional <b>timeout</b> keyword is specified, it defines the idle timeout for the temporary access list.</p> <p>Valid values, in minutes, range from 1 to 9999.</p>

## Verifying Lock-and-Key Configuration

You can verify that lock-and-key is successfully configured on the router by asking a user to test the connection. The user should be at a host that is permitted in the dynamic access list, and the user should have AAA authentication and authorization configured.

To test the connection, the user should Telnet to the router, allow the Telnet session to close, and then attempt to access a host on the other side of the router. This host must be one that is permitted by the dynamic access list. The user should access the host with an application that uses the IP protocol.

The following sample display illustrates what end-users might see if they are successfully authenticated. Notice that the Telnet connection is closed immediately after the password is entered and authenticated. The temporary access list entry is then created, and the host that initiated the Telnet session now has access inside the firewall.

```
Router% telnet corporate
Trying 172.21.52.1 ...
Connected to corporate.example.com.
Escape character is '^]'.
User Access Verification
Password:Connection closed by foreign host.
```

You can then use the **show access-lists** command at the router to view the dynamic access lists, which should include an additional entry permitting the user access through the router.

## Displaying Dynamic Access List Entries

You can display temporary access list entries when they are in use. After a temporary access list entry is cleared by you or by the absolute or idle timeout parameter, it can no longer be displayed. The number of matches displayed indicates the number of times the access list entry was hit.

To view dynamic access lists and any temporary access list entries that are currently established, use the following command in privileged EXEC mode:

Command	Purpose
Router# <b>show access-lists</b> [ <i>access-list-number</i> ]	Displays dynamic access lists and temporary access list entries.

## Manually Deleting Dynamic Access List Entries

To manually delete a temporary access list entry, use the following command in privileged EXEC mode:

Command	Purpose
Router# <b>clear access-template</b> [ <i>access-list-number</i>   <i>name</i> ] [ <i>dynamic-name</i> ] [ <i>source</i> ] [ <i>destination</i> ]	Deletes a dynamic access list.

## Configuration Examples for Lock-and-Key

### Example Lock-and-Key with Local Authentication

This example shows how to configure lock-and-key access, with authentication occurring locally at the router. Lock-and-key is configured on the Ethernet 0 interface.

```
interface ethernet0
 ip address 172.18.23.9 255.255.255.0
 ip access-group 101 in
 access-list 101 permit tcp any host 172.18.21.2 eq telnet
 access-list 101 dynamic mytestlist timeout 120 permit ip any any
 line vty 0
 login local
 autocmd access-enable timeout 5
```

The first access-list entry allows only Telnet into the router. The second access-list entry is always ignored until lock-and-key is triggered.

In the **access-list** command, the timeout is the absolute timeout. In this example, the lifetime of the mytestlist ACL is 120 minutes; that is, when a user logs in and enable the **access-enable** command, a dynamic ACL is created for 120 minutes (the maximum absolute time). The session is closed after 120 minutes, whether or not anyone is using it.

In the **access-enable** command, the timeout is the idle timeout. In this example, each time the user logs in or authenticates there is a 5-minute session. If there is no activity, the session closes in 5 minutes and the user has to reauthenticate. If the user uses the connection, the absolute time takes affect and the session closes in 120 minutes.

After a user opens a Telnet session into the router, the router will attempt to authenticate the user. If authentication is successful, the **autocmd** executes and the Telnet session terminates. The **autocmd**

creates a temporary inbound access list entry at the Ethernet 0 interface, based on the second access-list entry (mytestlist). If there is no activity, this temporary entry will expire after 5 minutes, as specified by the timeout.

## Example Lock-and-Key with TACACS+ Authentication

Cisco recommends that you use a TACACS+ server for authentication, as shown in the example.

The following example shows how to configure lock-and-key access, with authentication on a TACACS+ server. Lock-and-key access is configured on the BRI0 interface. Four VTY ports are defined with the password "password1".

```
aaa authentication login default group tacacs+ enable
aaa accounting exec stop-only group tacacs+
aaa accounting network stop-only group tacacs+
enable password ciscotac
!
isdn switch-type basic-dms100
!
interface ethernet0
ip address 172.18.23.9 255.255.255.0
!
interface BRI0
ip address 172.18.21.1 255.255.255.0
encapsulation ppp
dialer idle-timeout 3600
dialer wait-for-carrier-time 100
dialer map ip 172.18.21.2 name dialermapname
dialer-group 1
isdn spid1 2036333715291
isdn spid2 2036339371566
ppp authentication chap
ip access-group 102 in
!
access-list 102 permit tcp any host 172.18.21.2 eq telnet
access-list 102 dynamic testlist timeout 5 permit ip any any
!
!
ip route 172.18.250.0 255.255.255.0 172.18.21.2
priority-list 1 interface BRI0 high
tacacs-server host 172.18.23.21
tacacs-server host 172.18.23.14
tacacs-server key test1
tftp-server rom alias all
!
dialer-list 1 protocol ip permit
!
line con 0
password password1
line aux 0
line VTY 0 4
autocommand access-enable timeout 5
password password1
!
```







## ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options.

- [Finding Feature Information, page 117](#)
- [Restrictions for ACL IP Options Selective Drop, page 117](#)
- [Information About ACL IP Options Selective Drop, page 118](#)
- [How to Configure ACL IP Options Selective Drop, page 118](#)
- [Configuration Examples for ACL IP Options Selective Drop, page 119](#)
- [Additional References for IP Access List Entry Sequence Numbering, page 120](#)
- [Feature Information for ACL IP Options Selective Drop, page 121](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Restrictions for ACL IP Options Selective Drop

Resource Reservation Protocol (RSVP) (Multiprotocol Label Switching traffic engineering [MPLS TE]), Internet Group Management Protocol Version 2 (IGMPv2), and other protocols that use IP options packets may not function in drop or ignore modes.

# Information About ACL IP Options Selective Drop

## Using ACL IP Options Selective Drop

The ACL IP Options Selective Drop feature allows a router to filter IP options packets, thereby mitigating the effects of these packets on a router and downstream routers, and perform the following actions:

- Drop all IP options packets that it receives and prevent options from going deeper into the network.
- Ignore IP options packets destined for the router and treat them as if they had no IP options.

For many users, dropping the packets is the best solution. However, in environments in which some IP options may be legitimate, reducing the load that the packets present on the routers is sufficient. Therefore, users may prefer to skip options processing on the router and forward the packet as though it were pure IP.

## Benefits of Using ACL IP Options Selective Drop

- Drop mode filters packets from the network and relieves downstream routers and hosts of the load from options packets.
- Drop mode minimizes loads to the Route Processor (RP) for options that require RP processing on distributed systems. Previously, the packets were always routed to or processed by the RP CPU. Now, the ignore and drop forms prevent the packets from impacting the RP performance.

# How to Configure ACL IP Options Selective Drop

## Configuring ACL IP Options Selective Drop

This section describes how to configure the ACL IP Options Selective Drop feature.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip options {drop | ignore}**
4. **exit**
5. **show ip traffic**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b> Router> enable	<ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip options {drop   ignore}</b>  <b>Example:</b> Router(config)# ip options drop	Drops or ignores IP options packets that are sent to the router.
<b>Step 4</b>	<b>exit</b>  <b>Example:</b> Router(config)# exit	Returns to privileged EXEC mode.
<b>Step 5</b>	<b>show ip traffic</b>  <b>Example:</b> Router# show ip traffic	(Optional) Displays statistics about IP traffic.

## Configuration Examples for ACL IP Options Selective Drop

### Example Configuring ACL IP Options Selective Drop

The following example shows how to configure the router (and downstream routers) to drop all options packets that enter the network:

```
Router(config)# ip options drop
% Warning:RSVP and other protocols that use IP Options packets may not function in drop or
ignore modes.
end
```

## Example Verifying ACL IP Options Selective Drop

The following sample output is displayed after using the **ip options drop** command:

```
Router# show ip traffic
IP statistics:
  Rcvd:  428 total, 323 local destination
         0 format errors, 0 checksum errors, 0 bad hop count
         0 unknown protocol, 0 not a gateway
         0 security failures, 0 bad options, 0 with options
  Opts:  0 end, 0 nop, 0 basic security, 0 loose source route
         0 timestamp, 0 extended security, 0 record route
         0 stream ID, 0 strict source route, 0 alert, 0 cipso, 0 ump
         0 other, 30 ignored
  Frags: 0 reassembled, 0 timeouts, 0 couldn't reassemble
         0 fragmented, 0 fragments, 0 couldn't fragment
  Bcast: 0 received, 0 sent
  Mcast: 323 received, 809 sent
  Sent:  809 generated, 591 forwarded
  Drop:  0 encapsulation failed, 0 unresolved, 0 no adjacency
         0 no route, 0 unicast RPF, 0 forced drop, 0 unsupported-addr
         0 options denied, 0 source IP address zero
```

## Additional References for IP Access List Entry Sequence Numbering

The following sections provide references related to IP access lists.

### Related Documents

Related Topic	Document Title
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
IP access list commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>

**Technical Assistance**

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a>

## Feature Information for ACL IP Options Selective Drop

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 11: Feature Information for ACL IP Options Selective Drop**

Feature Name	Releases	Feature Information
ACL IP Options Selective Drop	Cisco IOS XE Release 2.1	<p>The ACL IP Options Selective Drop feature allows Cisco routers to filter packets containing IP options or to mitigate the effects of IP options on a router or downstream routers by dropping these packets or ignoring the processing of the IP options.</p> <p>This feature was introduced on Cisco ASR 1000 Series Aggregation Services Routers.</p> <p>The following command was introduced: <b>ip options</b>.</p>





## Displaying and Clearing IP Access List Data Using ACL Manageability

This module describes how to display the entries in an IP access list and the number of packets that have matched each entry. Users can get these statistics globally, or per interface and per incoming or outgoing traffic direction, by using the ACL Manageability feature. Viewing details of incoming and outgoing traffic patterns on various interfaces of a network device can help secure devices against attacks coming in on a particular interface. This module also describes how to clear counters so that the count of packets matching an access list entry will restart from zero.

- [Finding Feature Information, page 123](#)
- [Information About Displaying and Clearing IP Access List Data Using ACL Manageability, page 124](#)
- [How to Display and Clear IP Access List Data, page 124](#)
- [Configuration Examples for Displaying and Clearing IP Access List Data Using ACL Manageability, page 127](#)
- [Additional References, page 128](#)
- [Feature Information for Displaying IP Access List Information and Clearing Counters, page 129](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

# Information About Displaying and Clearing IP Access List Data Using ACL Manageability

## Benefits of ACL Manageability

Prior to Cisco IOS Release 12.4(6)T, the ACL infrastructure in Cisco IOS software maintained only global statistics for each ACE in an ACL. With this method, if an ACL is applied to multiple interfaces, the maintained ACE statistics are the sum of incoming and outgoing packet matches (hits) on all the interfaces on which that ACL is applied.

However, if ACE statistics are maintained per interface and per incoming or outgoing traffic direction, users can view specific details of incoming and outgoing traffic patterns and the effectiveness of ACEs on the various interfaces of a network device. This type of information is useful for securing devices against attacks coming in on a particular interface.

## Support for Interface-Level ACL Statistics

With Cisco IOS Release 12.4(6)T, the ACL infrastructure in Cisco IOS software is now extended to support the maintenance, display, and clearing of ACE statistics per interface and per incoming or outgoing traffic direction for ACLs. This support is often referred to as “support for interface-level statistics.”

**Note**

If the same access-group ACL is also used by other features, the maintained interface statistics are not updated when a packet match is detected by the other features. In this case, the sum of all the interface level statistics that are maintained for an ACL may not add up to the global statistics for that ACL.

## How to Display and Clear IP Access List Data

This section contains the following procedures for displaying IP access lists and the counts of packets that match (hit) each list, and for clearing IP access list counters.

**Note**

Alternatively, 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. For more information, see the “IP Access List Logging” section of the “IP Access List Overview.”

## Displaying Global IP ACL Statistics

Perform this task to display all IP access lists on the router and counts of packets that have matched.



## SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-number* | *access-list-name*]

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>show ip access-list</b> [ <i>access-list-number</i>   <i>access-list-name</i> ]  <b>Example:</b> Router# show ip access-list limited	Displays IP access list information.  <ul style="list-style-type: none"> <li>• This example displays statistics for all interfaces that use the access list named "limited."</li> </ul>

## Displaying Interface-Level IP ACL Statistics

This section describes how to display IP ACE statistics per interface and per incoming or outgoing traffic direction for ACLs. This feature is known as ACL Manageability.

**Note**

- ACL Manageability supports:
  - Only nondistributed software switched platforms.
  - Standard and extended statically configured ACLs, and Threat Mitigation Service (TMS) dynamic ACEs.
- ACL Manageability does not support:
  - Reflexive and user-configured dynamic ACLs and dynamic ACE blocks, such as Firewall and Authentication Proxy.
  - Virtual-template and virtual-access interfaces.

&gt;

## SUMMARY STEPS

1. `enable`
2. `show ip access-list interface interface-name [in|out]`

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>show ip access-list interface interface-name [in out]</b>  <b>Example:</b> Router# show ip access-list interface FastEthernet 0/0 in	Displays IP access list information. <ul style="list-style-type: none"> <li>• This example displays statistics about traffic coming into the FastEthernet interface.</li> <li>• To display debugging information about ACL interface-level statistics, use the <b>debug ip access-list intstats</b> command.</li> </ul>

## Clearing the Access List Counters

The system counts how many packets match (hit) each line of an access list; the counters are displayed by the **show access-lists** EXEC command. Perform this task to clear the counters of an access list. You might do this if you are trying to determine a more recent count of packets that match an access list, starting from zero.

## SUMMARY STEPS

1. `enable`
2. `clear ip access-list counters {access-list-number | access-list-name}`

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>clear ip access-list counters {access-list-number   access-list-name}</b>	Clears IP access list counters.

	Command or Action	Purpose
	<b>Example:</b> Router# clear access-list counters corpmark	

## Configuration Examples for Displaying and Clearing IP Access List Data Using ACL Manageability

### Example Displaying Global IP ACL Statistics

The following example displays global statistics for ACL 150:

```
Router# show ip access-list 150
Extended IP access list 150
 10 permit ip host 10.1.1.1 any (3 matches)
 30 permit ip host 10.2.2.2 any (27 matches)
```

### Example Displaying Input Statistics

The following example displays statistics on incoming packets gathered from the FastEthernet interface 0/1, associated with access list 150 (ACL number):

```
Router#
show ip access-list interface FastEthernet 0/1 in
Extended IP access list 150 in
 10 permit ip host 10.1.1.1 any (3 matches)
 30 permit ip host 10.2.2.2 any (12 matches)
```

### Example Displaying Output Statistics

The following example displays statistics on outgoing packets gathered from the FastEthernet interface 0/0:

```
Router#
show ip access-list interface FastEthernet 0/0 out
Extended IP access list myacl out
 5 deny ip any 10.1.0.0 0.0.255.255
 10 permit udp any any eq snmp (6 matches)
```

## Example Displaying Input and Output Statistics



### Note

If no direction is specified, any input and output ACLs applied to that interface are displayed.

The following example displays input and output statistics gathered from the FastEthernet interface 0/0:

```
Router#
show ip access-list interface FastEthernet 0/0
Extended IP access list 150 in
  10 permit ip host 10.1.1.1 any
  30 permit ip host 10.2.2.2 any (15 matches)
Extended IP access list myacl out
  5 deny ip any 10.1.0.0 0.0.255.255
  10 permit udp any any eq snmp (6 matches)
```

## Example Clearing Global and Interface Statistics for an IP Access List

The following example clears global and interface statistics for IP ACL 150:

```
Router#
clear ip access-list counters 150
```

## Example Clearing Global and Interface Statistics for All IP Access Lists

The following example clears global and interface statistics for all IP ACLs:

```
Router#
clear ip access-list counters
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
Security commands	<i>Cisco IOS Security Command Reference</i>

### Standards

Standard	Title
No new or modified standards are supported by this feature.	--

**MIBs**

MIB	MIBs Link
No new or modified MIBs are supported by this feature.	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

**RFCs**

RFC	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	--

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for Displaying IP Access List Information and Clearing Counters

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 12: Feature Information for Displaying and Clearing IP Access List Data Using ACL Manageability**

Feature Name	Releases	Feature Information
ACL Manageability	Cisco IOS XE Release 3.9S	The ACL Manageability feature enables users to display and clear Access Control Entry (ACE) statistics per interface and per incoming or outgoing traffic direction for access control lists (ACLs).



## ACL Syslog Correlation

The Access Control List (ACL) Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies the ACE, within the ACL, that generated the syslog entry.

- [Finding Feature Information, page 131](#)
- [Prerequisites for ACL Syslog Correlation, page 131](#)
- [Information About ACL Syslog Correlation, page 132](#)
- [How to Configure ACL Syslog Correlation, page 132](#)
- [Configuration Examples for ACL Syslog Correlation, page 140](#)
- [Additional References for IPv6 IOS Firewall, page 141](#)
- [Feature Information for ACL Syslog Correlation, page 142](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

### Prerequisites for ACL Syslog Correlation

Before you configure the ACL Syslog Correlation feature, you must understand the concepts in the "IP Access List Overview" module.

The ACL Syslog Correlation feature appends a user-defined cookie or a device-generated hash value to ACE messages in the syslog. These values are only appended to ACE messages when the log option is enabled for the ACE.

# Information About ACL Syslog Correlation

## ACL Syslog Correlation Tags

The ACL Syslog Correlation feature appends a tag (either a user-defined cookie or a device-generated MD5 hash value) to access control entry (ACE) syslog entries. This tag uniquely identifies an ACE that generated the syslog entry.

Network management software can use the tag to identify which ACE generated a specific syslog event. For example, network administrators can select an ACE rule in the network management application and can then view the corresponding syslog events for that ACE rule.

To append a tag to the syslog message, the ACE that generates the syslog event must have the log option enabled. The system appends only one type of tag (either a user-defined cookie or a device-generated MD5 hash value) to each message.

To specify a user-defined cookie tag, the user must enter the cookie value when configuring the ACE log option. The cookie must be in alpha-numeric form, it cannot be greater than 64 characters, and it cannot start with hex-decimal notation (such as 0x).

To specify a device-generated MD5 hash value tag, the hash-generation mechanism must be enabled on the device and the user must not enter a cookie value while configuring the ACE log option.

## ACE Syslog Messages

When a packet is matched against an access control entry (ACE) in an ACL, the system checks whether the log option is enabled for that event. If the log option is enabled and the ACL Syslog Correlation feature is configured on the device, the system attaches the tag to the syslog message. The tag is displayed at the end of the syslog message, in addition to the standard information.

The following is a sample syslog message showing a user-defined cookie tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) -> 192.168.16.2(23), 1 packet [User_permitted_ACE]
```

The following is a sample syslog message showing a hash value tag:

```
Jun 5 12:55:44.359: %SEC-6-IPACCESSLOGP: list logacl permitted tcp 192.168.16.1(38402) -> 192.168.16.2(23), 1 packet [0x723E6E12]
```

# How to Configure ACL Syslog Correlation

## Enabling Hash Value Generation on a Device

Perform this task to configure the device to generate an MD5 hash value for each log-enabled access control entry (ACE) in the system that is not configured with a user-defined cookie.



When the hash value generation setting is enabled, the system checks all existing ACEs and generates a hash value for each ACE that requires one. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

## SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list logging hash-generation**
4. **end**
5. Do one of the following:
  - **show ip access-list** *access-list-number*
  - **show ip access-list** *access-list-name*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list logging hash-generation</b>  <b>Example:</b> Device(config)# ip access-list logging hash-generation	Enables hash value generation on the device.  • If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.
<b>Step 4</b>	<b>end</b>  <b>Example:</b> Device(config)# end	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
<b>Step 5</b>	Do one of the following: <ul style="list-style-type: none"> <li>• <b>show ip access-list</b> <i>access-list-number</i></li> <li>• <b>show ip access-list</b> <i>access-list-name</i></li> </ul>	(Optional) Displays the contents of the numbered or named IP access list.  • Review the output to confirm that the access list for a log-enabled ACE includes the generated hash value.

	Command or Action	Purpose
	<b>Example:</b> <pre>Device# show ip access-list 101</pre> <b>Example:</b> <pre>Device# show ip access-list acl</pre>	

## Disabling Hash Value Generation on a Device

Perform this task to disable hash value generation on the device. When the hash value generation setting is disabled, all previously generated hash values are removed from the system.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **no ip access-list logging hash-generation**
4. **end**
5. Do one of the following:
  - **show ip access-list** *access-list-number*
  - **show ip access-list** *access-list-name*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> <pre>Device&gt; enable</pre>	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> <pre>Device# configure terminal</pre>	Enters global configuration mode.
<b>Step 3</b>	<b>no ip access-list logging hash-generation</b>	Disables hash value generation on the device.

	Command or Action	Purpose
	<b>Example:</b>  Device(config)# no ip access-list logging hash-generation	<ul style="list-style-type: none"> <li>The system removes any previously created hash values from the system.</li> </ul>
<b>Step 4</b>	<b>end</b>  <b>Example:</b>  Device(config)# end	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
<b>Step 5</b>	Do one of the following: <ul style="list-style-type: none"> <li><b>show ip access-list</b> <i>access-list-number</i></li> <li><b>show ip access-list</b> <i>access-list-name</i></li> </ul> <b>Example:</b>  Device# show ip access-list 101  <b>Example:</b>  Device# show ip access-list acl	(Optional) Displays the contents of the IP access list. <ul style="list-style-type: none"> <li>Review the output to confirm that the access list for a log-enabled ACE does not have a generated hash value.</li> </ul>

## Configuring ACL Syslog Correlation Using a User-Defined Cookie

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a user-defined cookie as the syslog message tag.

The example in this section shows how to configure the ACL Syslog Correlation feature using a user-defined cookie for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a user-defined cookie for both numbered and named access lists, and for both standard and extended access lists.

**Note**

The following restrictions apply when choosing the user-defined cookie value:

- The maximum number of characters is 64.
- The cookie cannot start with hexadecimal notation (such as 0x).
- The cookie cannot be the same as, or a subset of, the following keywords: **reflect**, **fragment**, **time-range**. For example, reflect and ref are not valid values. However, the cookie can start with the keywords. For example, reflectedACE and fragment\_33 are valid values
- The cookie must contain only alphanumeric characters.

&gt;

**SUMMARY STEPS**

1. **enable**
2. **configure terminal**
3. **access-list** *access-list-number* **permit** *protocol source destination* **log** *word*
4. **end**
5. **show ip access-list** *access-list-number*

**DETAILED STEPS**

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>access-list</b> <i>access-list-number</i> <b>permit</b> <i>protocol source destination</i> <b>log</b> <i>word</i>  <b>Example:</b> Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log UserDefinedValue	Defines an extended IP access list and a user-defined cookie value.  <ul style="list-style-type: none"> <li>• Enter the cookie value as the <i>word</i> argument.</li> </ul>
<b>Step 4</b>	<b>end</b>  <b>Example:</b> Device(config)# end	(Optional) Exits global configuration mode and returns to privileged EXEC mode.

	Command or Action	Purpose
<b>Step 5</b>	<b>show ip access-list</b> <i>access-list-number</i>  <b>Example:</b> Device# show ip access-list 101	(Optional) Displays the contents of the IP access list.  <ul style="list-style-type: none"> <li>Review the output to confirm that the access list includes the user-defined cookie value.</li> </ul>

### Examples

The following is sample output from the **show ip access-list** command for an access list with a user-defined cookie value.

```
Device# show ip access-list
101
Extended IP access list 101
30 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = UserDefinedValue)
```

## Configuring ACL Syslog Correlation Using a Hash Value

Perform this task to configure the ACL Syslog Correlation feature on a device for a specific access list, using a device-generated hash value as the syslog message tag.

The steps in this section shows how to configure the ACL Syslog Correlation feature using a device-generated hash value for a numbered access list. However, you can configure the ACL Syslog Correlation feature using a device-generated hash value for both numbered and named access lists, and for both standard and extended access lists.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list logging hash-generation**
4. **access-list** *access-list-number* **permit protocol source destination log**
5. **end**
6. **show ip access-list** *access-list-number*

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ip access-list logging hash-generation</b>  <b>Example:</b> Device(config)# ip access-list logging hash-generation	Enables hash value generation on the device. <ul style="list-style-type: none"> <li>• If an ACE exists that is log enabled, and requires a hash value, the device automatically generates the value and displays the value on the console.</li> </ul>
<b>Step 4</b>	<b>access-list <i>access-list-number</i> permit protocol source destination log</b>  <b>Example:</b> Device(config)# access-list 102 permit tcp host 10.1.1.1 host 10.1.1.2 log	Defines an extended IP access list. <ul style="list-style-type: none"> <li>• Enable the log option for the access list, but do not specify a cookie value.</li> <li>• The device automatically generates a hash value for the newly defined access list.</li> </ul>
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config)# end	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
<b>Step 6</b>	<b>show ip access-list <i>access-list-number</i></b>  <b>Example:</b> Device# show ip access-list 102	(Optional) Displays the contents of the IP access list. <ul style="list-style-type: none"> <li>• Review the output to confirm that the access list includes the router-generated hash value.</li> </ul>

### Examples

The following is sample output from the **show ip access-list** command for an access list with a device-generated hash value.

```
Device# show ip access-list
102
Extended IP access list 102
10 permit tcp host 10.1.1.1 host 10.1.1.2 log (hash = 0x7F9CF6B9)
```

## Changing the ACL Syslog Correlation Tag Value

Perform this task to change the value of the user-defined cookie or replace a device-generated hash value with a user-defined cookie.

The steps in this section shows how to change the ACL Syslog Correlation tag value on a numbered access list. However, you can change the ACL Syslog Correlation tag value for both numbered and named access lists, and for both standard and extended access lists.

## SUMMARY STEPS

1. **enable**
2. **show access-list**
3. **configure terminal**
4. **access-list** *access-list-number* **permit** *protocol source destination* **log** *word*
5. **end**
6. **show ip access-list** *access-list-number*

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>show access-list</b>  <b>Example:</b> Device(config)# show access-list	(Optional) Displays the contents of the access list.
<b>Step 3</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 4</b>	<b>access-list</b> <i>access-list-number</i> <b>permit</b> <i>protocol source destination</i> <b>log</b> <i>word</i>  <b>Example:</b> Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV  <b>Example:</b> OR  <b>Example:</b>	Modifies the cookie or changes the hash value to a cookie. <ul style="list-style-type: none"> <li>• You must enter the entire access list configuration command, replacing the previous tag value with the new tag value.</li> </ul>

	Command or Action	Purpose
	<b>Example:</b> <pre>Device(config)# access-list 101 permit tcp any any log replacehash</pre>	
<b>Step 5</b>	<b>end</b> <b>Example:</b> <pre>Device(config)# end</pre>	(Optional) Exits global configuration mode and returns to privileged EXEC mode.
<b>Step 6</b>	<b>show ip access-list <i>access-list-number</i></b> <b>Example:</b> <pre>Device# show ip access-list 101</pre>	(Optional) Displays the contents of the IP access list. <ul style="list-style-type: none"> <li>• Review the output to confirm the changes.</li> </ul>

## Troubleshooting Tips

Use the **debug ip access-list hash-generation** command to display access list debug information. The following is an example of the **debug** command output:

```
Device# debug ip access-list hash-generation
Syslog hash code generation debugging is on
Device# show debug
IP ACL:
Syslog hash code generation debugging is on
Device# no debug ip access-list hash-generation

Syslog hash code generation debugging is off
Device# show debug
Device#
```

# Configuration Examples for ACL Syslog Correlation

## Example: Configuring ACL Syslog Correlation Using a User-Defined Cookie

The following example shows how to configure the ACL Syslog Correlation feature on a device using a user-defined cookie.

```
Device#
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.6 log cook_33_std
Device(config)# do show ip access 33
Standard IP access list 33
```



```
10 permit 10.10.10.6 log (tag = cook_33_std)
Device(config)# end
```

## Example: Configuring ACL Syslog Correlation using a Hash Value

The following examples shows how to configure the ACL Syslog Correlation feature on a device using a device-generated hash value.

```
Device# debug ip access-list hash-generation
Syslog MD5 hash code generation debugging is on
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# access-list 33 permit 10.10.10.7 log
Device(config)#
*Nov 7 13:51:23.615: %IPACL-HASHGEN: Hash Input: 33 standard permit 10.10.10.7
Hash Output: 0xCE87F535
Device(config)#
do show ip access 33

Standard IP access list 33
 10 permit 10.10.10.6 log (tag = cook_33_std)
 20 permit 10.10.10.7 log (hash = 0xCE87F535)
```

## Example: Changing the ACL Syslog Correlation Tag Value

The following example shows how to replace an existing access list user-defined cookie with a new cookie value, and how to replace a device-generated hash value with a user-defined cookie value.

```
Device# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Device(config)# do show ip access-list 101
Extended IP access list 101
 10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = MyCookie)
 20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp host 10.1.1.1 host 10.1.1.2 log NewUDV
Device(config)# do show access-list
Extended IP access list 101
 10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
 20 permit tcp any any log (hash = 0x75F078B9)
Device(config)# access-list 101 permit tcp any any log replacehash
Device(config)# do show access-list
Extended IP access list 101
 10 permit tcp host 10.1.1.1 host 10.1.1.2 log (tag = NewUDV)
 20 permit tcp any any log (tag = replacehash)
```

## Additional References for IPv6 IOS Firewall

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>

Related Topic	Document Title
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>
IPv6 commands	<a href="#">Cisco IOS IPv6 Command Reference</a>
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS IPv6 features	<a href="#">Cisco IOS IPv6 Feature Mapping</a>

### Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

### 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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for ACL Syslog Correlation

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 13: Feature Information for ACL Syslog Correlation**

Feature Name	Releases	Feature Information
ACL Syslog Correlation	Cisco IOS XE Release 3.6S	The ACL Syslog Correlation feature appends a tag (either a user-defined cookie or a router-generated MD5 hash value) to ACE syslog entries. This tag uniquely identifies the ACE , within the ACL, that generated the syslog entry.





## IPv6 Access Control Lists

Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering of traffic based on source and destination addresses, and inbound and outbound traffic to a specific interface. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control. Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.

This module describes how to configure IPv6 traffic filtering and to control access to virtual terminal lines.

- [Finding Feature Information, page 145](#)
- [Information About IPv6 Access Control Lists, page 146](#)
- [How to Configure IPv6 Access Control Lists, page 147](#)
- [Configuration Examples for IPv6 Access Control Lists, page 152](#)
- [Additional References, page 153](#)
- [Feature Information for IPv6 Access Control Lists, page 154](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

# Information About IPv6 Access Control Lists

## Access Control Lists for IPv6 Traffic Filtering

The standard ACL functionality in IPv6 is similar to standard ACLs in IPv4. Access lists determine what traffic is blocked and what traffic is forwarded at device interfaces and allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Each access list has an implicit deny statement at the end. IPv6 ACLs are defined and their deny and permit conditions are set using the **ipv6 access-list** command with the **deny** and **permit** keywords in global configuration mode.

IPv6 extended ACLs augments standard IPv6 ACL functionality to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control (functionality similar to extended ACLs in IPv4).

## IPv6 Packet Inspection

The following header fields are used for IPv6 inspection: traffic class, flow label, payload length, next header, hop limit, and source or destination IP address. For further information on and descriptions of the IPv6 header fields, see RFC 2474.

## Access Class Filtering in IPv6

Filtering incoming and outgoing connections to and from the device based on an IPv6 ACL is performed using the **ipv6 access-class** command in line configuration mode. The **ipv6 access-class** command is similar to the **access-class** command, except the IPv6 ACLs are defined by a name. If the IPv6 ACL is applied to inbound traffic, the source address in the ACL is matched against the incoming connection source address and the destination address in the ACL is matched against the local device address on the interface. If the IPv6 ACL is applied to outbound traffic, the source address in the ACL is matched against the local device address on the interface and the destination address in the ACL is matched against the outgoing connection source address. We recommend that identical restrictions are set on all the virtual terminal lines because a user can attempt to connect to any of them.

# How to Configure IPv6 Access Control Lists

## Configuring IPv6 Traffic Filtering

### Creating and Configuring an IPv6 ACL for Traffic Filtering


**Note**

IPv6 ACLs on the Cisco ASR 1000 platform do not contain implicit permit rules. The IPv6 neighbor discovery process uses the IPv6 network-layer service; therefore, to enable IPv6 neighbor discovery, you must add IPv6 ACLs to allow IPv6 neighbor discovery packets to be sent and received on an interface. In IPv4, the Address Resolution Protocol (ARP), which is equivalent to the IPv6 neighbor discovery process, uses a separate data-link-layer protocol; therefore, by default IPv4 ACLs implicitly allow ARP packets to be sent and received on an interface.

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list *access-list-name***
4. Do one of the following:
  - **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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
  - **deny 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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]

#### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.

	Command or Action	Purpose
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ipv6 access-list <i>access-list-name</i></b>  <b>Example:</b> Device(config)# ipv6 access-list outbound	Defines an IPv6 ACL, and enters IPv6 access list configuration mode. <ul style="list-style-type: none"> <li>The <i>access-list name</i> argument specifies the name of the IPv6 ACL. IPv6 ACL names cannot contain a space or quotation mark, or begin with a numeral.</li> </ul>
<b>Step 4</b>	Do one of the following: <ul style="list-style-type: none"> <li><b>permit protocol</b> {<i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix</i> / <i>prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>} [<b>operator</b> [<i>port-number</i>]] [<b>dest-option-type</b> [<i>doh-number</i>   <i>doh-type</i>]] [<b>dscp</b> <i>value</i>] [<b>flow-label</b> <i>value</i>] [<b>fragments</b>] [<b>log</b>] [<b>log-input</b>] [<b>mobility</b>] [<b>mobility-type</b> [<i>mh-number</i>   <i>mh-type</i>]] [<b>routing</b>] [<b>routing-type</b> <i>routing-number</i>] [<b>sequence</b> <i>value</i>] [<b>time-range</b> <i>name</i>]</li> <li><b>deny protocol</b> {<i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>} [<i>operator</i> <i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>} [<i>operator</i> [<i>port-number</i>]] [<b>dest-option-type</b> [<i>doh-number</i>   <i>doh-type</i>]] [<b>dscp</b> <i>value</i>] [<b>flow-label</b> <i>value</i>] [<b>fragments</b>] [<b>log</b>] [<b>log-input</b>] [<b>mobility</b>] [<b>mobility-type</b> [<i>mh-number</i>   <i>mh-type</i>]] [<b>routing</b>] [<b>routing-type</b> <i>routing-number</i>] [<b>sequence</b> <i>value</i>] [<b>time-range</b> <i>name</i>] [<b>undetermined-transport</b>]</li> </ul> <b>Example:</b> Device(config-ipv6-acl)# permit tcp 2001:DB8:0300:0201::/32 eq telnet any <b>Example:</b> Device(config-ipv6-acl)# deny tcp host 2001:DB8:1::1 any log-input	Specifies permit or deny conditions for an IPv6 ACL.



## Applying the IPv6 ACL to an Interface

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ipv6 traffic-filter** *access-list-name* {**in**|**out**}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>type number</i>  <b>Example:</b> Device(config)# interface gigabitethernet 0/0/0	Specifies the interface type and number, and enters interface configuration mode.
<b>Step 4</b>	<b>ipv6 traffic-filter</b> <i>access-list-name</i> { <b>in</b>   <b>out</b> }  <b>Example:</b> Device(config-if)# ipv6 traffic-filter outbound out	Applies the specified IPv6 access list to the interface specified in the previous step.

# Controlling Access to a vty

## Creating an IPv6 ACL to Provide Access Class Filtering

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list *access-list-name***
4. Do one of the following:
  - **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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
  - **deny 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*]] [**dest-option-type** [*doh-number* | *doh-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  • Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ipv6 access-list <i>access-list-name</i></b>  <b>Example:</b> Device(config)# ipv6 access-list cisco	Defines an IPv6 ACL, and enters IPv6 access list configuration mode.

	Command or Action	Purpose
<b>Step 4</b>	<p>Do one of the following:</p> <ul style="list-style-type: none"> <li>• <b>permit protocol</b> {<i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>} [<i>operator</i> [<i>port-number</i>]] {<i>destination-ipv6-prefix / prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>} [<i>operator</i> [<i>port-number</i>]] [<b>dest-option-type</b> [<i>doh-number</i>   <i>doh-type</i>]] [<b>dscp</b> <i>value</i>] [<b>flow-label</b> <i>value</i>] [<b>fragments</b>] [<b>log</b>] [<b>log-input</b>] [<b>mobility</b>] [<b>mobility-type</b> [<i>mh-number</i>   <i>mh-type</i>]] [<b>routing</b>] [<b>routing-type</b> <i>routing-number</i>] [<b>sequence</b> <i>value</i>] [<b>time-range</b> <i>name</i>]</li> <li>• <b>deny protocol</b> {<i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>} [<i>operator</i> <i>port-number</i>]] {<i>destination-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>} [<i>operator</i> [<i>port-number</i>]] [<b>dest-option-type</b> [<i>doh-number</i>   <i>doh-type</i>]] [<b>dscp</b> <i>value</i>] [<b>flow-label</b> <i>value</i>] [<b>fragments</b>] [<b>log</b>] [<b>log-input</b>] [<b>mobility</b>] [<b>mobility-type</b> [<i>mh-number</i>   <i>mh-type</i>]] [<b>routing</b>] [<b>routing-type</b> <i>routing-number</i>] [<b>sequence</b> <i>value</i>] [<b>time-range</b> <i>name</i>] [<b>undetermined-transport</b>]</li> </ul> <p><b>Example:</b></p> <pre>Device(config-ipv6-acl)# permit ipv6 host 2001:DB8:0:4::32 any eq telnet</pre> <p><b>Example:</b></p> <pre>Device(config-ipv6-acl)# deny ipv6 host 2001:DB8:0:6::6/32 any</pre>	Specifies permit or deny conditions for an IPv6 ACL.

## Applying an IPv6 ACL to the Virtual Terminal Line

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **line** [*aux* | *console* | *tty* | *vty*] *line-number*[*ending-line-number*]
4. **ipv6 access-class** *ipv6-access-list-name* {**in** | **out**}

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>	Enables privileged EXEC mode.

	Command or Action	Purpose
	<b>Example:</b> Device> enable	<ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>line [aux  console  tty  vty]</b> <i>line-number[ending-line-number]</i>  <b>Example:</b> Device(config)# line vty 0 4	Identifies a specific line for configuration and enters line configuration mode.  <ul style="list-style-type: none"> <li>In this example, the <b>vty</b> keyword is used to specify the virtual terminal lines for remote console access.</li> </ul>
<b>Step 4</b>	<b>ipv6 access-class <i>ipv6-access-list-name</i> {in  out}</b>  <b>Example:</b> Device(config-line)# ipv6 access-class cisco in	Filters incoming and outgoing connections to and from the device based on an IPv6 ACL.

## Configuration Examples for IPv6 Access Control Lists

### Example: Verifying IPv6 ACL Configuration

In this example, the **show ipv6 access-list** command is used to verify that IPv6 ACLs are configured correctly:

```
Device> show ipv6 access-list
```

```
IPv6 access list inbound
  permit tcp any any eq bgp reflect tcptraffic (8 matches) sequence 10
  permit tcp any any eq telnet reflect tcptraffic (15 matches) sequence 20
  permit udp any any reflect udptraffic sequence 30

IPv6 access list tcptraffic (reflexive) (per-user)
  permit tcp host 2001:DB8:1::32 eq bgp host 2001:DB8:2::32 eq 11000 timeout 300 (time
left 243) sequence 1
  permit tcp host 2001:DB8:1::32 eq telnet host 2001:DB8:2::32 eq 11001 timeout 300 (time
left 296) sequence 2

IPv6 access list outbound
  evaluate udptraffic
  evaluate tcptraffic
```

## Example: Creating and Applying an IPv6 ACL

The following example shows how to restrict HTTP access to certain hours during the day and log any activity outside of the permitted hours:

```
Device# configure terminal
Device(config)# time-range lunchtime
Device(config-time-range)# periodic weekdays 12:00 to 13:00
Device(config-time-range)# exit
Device(config)# ipv6 access-list OUTBOUND
Device(config-ipv6-acl)# permit tcp any any eq www time-range lunchtime
Device(config-ipv6-acl)# deny tcp any any eq www log-input
Device(config-ipv6-acl)# permit tcp 2001:DB8::/32 any
Device(config-ipv6-acl)# permit udp 2001:DB8::/32 any
Device(config-ipv6-acl)# end
```

## Example: Controlling Access to a vty

In the following example, incoming connections to the virtual terminal lines 0 to 4 are filtered based on the IPv6 access list named `acl1`:

```
ipv6 access-list acl1
 permit ipv6 host 2001:DB8:0:4::2/32 any
!
line vty 0 4
 ipv6 access-class acl1 in
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
IP access list commands	<i>Cisco IOS Security Command Reference</i>
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for IPv6 Access Control Lists

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 14: Feature Information for IPv6 Access Control Lists**

Feature Name	Releases	Feature Information
IPv6 Services: Extended Access Control Lists	Cisco IOS XE Release 2.1	Standard IPv6 ACL functionality was extended to support traffic filtering based on IPv6 option headers and optional, upper-layer protocol type information for finer granularity of control.



## Configuring Template ACLs

When user profiles are configured using RADIUS Attribute 242 or vendor-specific attribute (VSA) Cisco-AVPairs, similar per-user access control lists (ACLs) may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

In networks where each subscriber has its own ACL, it is common for the ACL to be the same for each user except for the user's IP address. The Template ACLs feature groups ACLs with many common access control elements (ACEs) into a single ACL that saves system resources.

- [Finding Feature Information, page 155](#)
- [Prerequisites for Template ACLs, page 156](#)
- [Restrictions for Template ACLs, page 156](#)
- [Information About Configuring Template ACLs, page 156](#)
- [How to Configure Template ACLs, page 160](#)
- [Configuration Examples for Template ACLs, page 161](#)
- [Additional References, page 162](#)
- [Feature Information for ACL Template, page 163](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Prerequisites for Template ACLs

- Cisco ASR 1000 series routers
- Cisco IOS XE Release 2.4 or a later release

## Restrictions for Template ACLs

Template ACLs are activated only for per-user ACLs configured through RADIUS Attribute 242 or VSA Cisco-AVPairs (ip:inacl/outacl). No other ACL types are processed by the Template ACL feature.

Template ACL functionality is available only for IPv4 ACLs.

Template ACL functionality is not available for the following types of per-user ACLs:

- Time-based ACLs
- Dynamic ACLs
- Evaluate ACLs
- Reflexive ACLs
- ACLs configured on ISG IP sessions
- IPv6 ACLs

### Disabling the Template ACL Feature

When the Template ACL feature is disabled, the system replaces all existing template ACL instances with ACLs. If the system does not have enough resources (in particular TCAM resources) to setup the required number of ACLs, the system generates an error message, and the request to disable the Template ACLs feature fails.

## Information About Configuring Template ACLs

### Template ACL Feature Design

When the service provider uses AAA servers to configure individual ACLs for each authorized session using with RADIUS attribute 242 or VSA Cisco-AVPairs, the number of sessions can easily exceed the maximum ACL number allowed by the system.

In networks where each subscriber has an ACL, it is common for the ACL to be the same for each user except for the user's IP address. Template ACLs alleviate this problem by grouping ACLs with many common ACEs into a single ACL that compiles faster and saves system resources.

The Template ACL feature is enabled by default, and ACLs set up using the RADIUS attribute 242 or VSA Cisco-AVPairs are considered for template status.

When the Template ACL feature is enabled, the system scans and evaluates all configured per-session ACLs and then creates all required template ACLs.



### Disabling Template ACLs

When the Template ACL feature is disabled, the system replaces all existing template ACL instances with ACLs. If the system does not have enough resources (in particular TCAM resources) to setup the required number of ACLs, the system generates an error message, and the request to disable the Template ACL feature fails.

Therefore, before you disable the Template ACL feature, use the **show access-list template summary** command to view the number of template ACLs in the system and ascertain if this number exceeds the system limitations.

When the template ACL feature is disabled, no new ACLs are considered for templating.

## Multiple ACLs

When the Template ACL feature is enabled, the system can identify when two per-user ACLs are similar, and the system consolidates the two per-user ACLs into one template ACL.

For example, the following example shows two ACLs for two separate users:

```
ip access-list extended Virtual-Access1.1#1 (PeerIP: 10.1.1.1)
permit igmp any host 10.1.1.1
permit icmp host 10.1.1.1 any
deny ip host 10.31.66.36 host 10.1.1.1
deny tcp host 10.1.1.1 host 10.31.66.36
permit udp any host 10.1.1.1
permit udp host 10.1.1.1 any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
ip access-list extended Virtual-Access1.1#2 (PeerIP: 10.13.11.2)
permit igmp any host 10.13.11.2
permit icmp host 10.13.11.2 any
deny ip host 10.31.66.36 host 10.13.11.2
deny tcp host 10.13.11.2 host 10.31.66.36
permit udp any host 10.13.11.2
permit udp host 10.13.11.2 any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
```

With the Template ACL feature is enabled, the system recognizes that these two ACLs are similar, and creates a template ACL as follows:

```
ip access-list extended Template_1
permit igmp any host <PeerIP>
permit icmp host <PeerIP> any
deny ip host 10.31.66.36 host <PeerIP>
deny tcp host <PeerIP> 10.31.66.36
permit udp any host <PeerIP>
permit udp host <PeerIP> any
permit udp any host 192.168.2.1
permit udp any host 192.168.222.1
permit icmp host 10.55.15.4 host 192.168.2.1
permit udp 10.22.11.0 0.0.0.255 host 192.168.211.2
permit tcp any host 192.168.222.1
permit ip host 10.55.15.4 host 192.168.2.1
permit tcp 10.22.11.0 0.0.0.255 host 192.168.211.2
```

In this example, the peer IP address is associated as follows:

- Virtual-Access1.1#1 10.1.1.1
- Virtual-Access1.1#2 10.13.11.2

The two ACLs are consolidated into one template ACL and are referenced as follows:

Virtual-Access1.1#1 maps to Template\_1(10.1.1.1)

Virtual-Access1.1#2 maps to Template\_1(10.13.11.2)

## VSA Cisco-AVPairs

Template ACL processing occurs for ACLs that are configured using Cisco-AVPairs. Only AVPairs that are defined using the ACL number are considered for the templating process.

To be considered for templating, AVPairs for incoming ACLs must conform to the following format:

ip:inacl#number={standard-access-control-list | extended-access-control-list}

For example: ip:inacl#10=deny ip any 10.13.16.0 0.0.0.255

To be considered for templating, AVPairs for outgoing ACLs must conform to the following format:

ip:outacl#number={standard-access-control-list | extended-access-control-list}

For example: ip:outacl#200=permit ip any any

For more information on Cisco-AVPairs, see the Cisco Vendor-Specific AVPair Attributes section of the *Cisco IOS ISG RADIUS CoA Interface Guide*.

## RADIUS Attribute 242

Template ACL processing occurs for ACLs that are configured using RADIUS attribute 242. Attribute 242 has the following format for an IP data filter:

Ascend-Data-Filter = "ip <dir> <action> [dstip <dest\_ipaddr\subnet\_mask>] [srcip <src\_ipaddr\subnet\_mask>] [<proto> [dstport <cmp> <value>] [srcport <cmp> <value>] [<est>]]"

The table below describes the elements in an attribute 242 entry for an IP data filter.

**Table 15: IP Data Filter Syntax Elements**

Element	Description
<b>ip</b>	Specifies an IP filter.
<b>&lt;dir&gt;</b>	Specifies the filter direction. Possible values are <b>in</b> (filtering packets coming into the router) or <b>out</b> (filtering packets going out of the router).
<b>&lt;action&gt;</b>	Specifies the action the router should take with a packet that matches the filter. Possible values are <b>forward</b> or <b>drop</b> .

Element	Description
<b>dstip &lt;dest_ipaddr/subnet_mask&gt;</b>	Enables destination-IP-address filtering. Applies to packets whose destination address matches the value of <b>&lt;dest_ipaddr&gt;</b> . If a subnet mask portion of the address is present, the router compares only the masked bits. If you set <b>&lt;dest_ipaddr&gt;</b> to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
<b>srcip &lt;src_ipaddr/subnet_mask&gt;</b>	Enables source-IP-address filtering. Applies to packets whose source address matches the value of <b>&lt;src_ipaddr&gt;</b> . If a subnet mask portion of the address is present, the router compares only the masked bits. If you set <b>&lt;src_ipaddr&gt;</b> to 0.0.0.0, or if this keyword is not present, the filter matches all IP packets.
<b>&lt;proto&gt;</b>	Specifies a protocol specified as a name or a number. Applies to packets whose protocol field matches this value. Possible names and numbers are <b>icmp (1)</b> , <b>tcp (6)</b> , <b>udp (17)</b> , and <b>ospf (89)</b> . If you set this value to zero (0), the filter matches any protocol.
<b>dstport &lt;cmp&gt; &lt;value&gt;</b>	<p>Enables destination-port filtering. This keyword is valid only when <b>&lt;proto&gt;</b> is set to <b>tcp (6)</b> or <b>udp (17)</b>. If you do not specify a destination port, the filter matches any port.</p> <p><b>&lt;cmp&gt;</b> defines how to compare the specified <b>&lt;value&gt;</b> to the actual destination port. This value can be <b>&lt;</b>, <b>=</b>, <b>&gt;</b>, or <b>!</b>.</p> <p><b>&lt;value&gt;</b> can be a name or a number. Possible names and numbers are <b>ftp-data (20)</b>, <b>ftp (21)</b>, <b>telnet (23)</b>, <b>nameserver (42)</b>, <b>domain (53)</b>, <b>tftp (69)</b>, <b>gopher (70)</b>, <b>finger (79)</b>, <b>www (80)</b>, <b>kerberos (88)</b>, <b>hostname (101)</b>, <b>nntp (119)</b>, <b>ntp (123)</b>, <b>exec (512)</b>, <b>login (513)</b>, <b>cmd (514)</b>, and <b>talk (517)</b>.</p>
<b>srcport &lt;cmp&gt; &lt;value&gt;</b>	<p>Enables source-port filtering. This keyword is valid only when <b>&lt;proto&gt;</b> is set to <b>tcp(6)</b> or <b>udp (17)</b>. If you do not specify a source port, the filter matches any port.</p> <p><b>&lt;cmp&gt;</b> defines how to compare the specified <b>&lt;value&gt;</b> to the actual destination port. This value can be <b>&lt;</b>, <b>=</b>, <b>&gt;</b>, or <b>!</b>.</p> <p><b>&lt;value&gt;</b> can be a name or a number. Possible names and numbers are <b>ftp-data (20)</b>, <b>ftp (21)</b>, <b>telnet(23)</b>, <b>nameserver(42)</b>, <b>domain(53)</b>, <b>tftp(69)</b>, <b>gopher(70)</b>, <b>finger(79)</b>, <b>www(80)</b>, <b>kerberos (88)</b>, <b>hostname (101)</b>, <b>nntp (119)</b>, <b>ntp(123)</b>, <b>exec (512)</b>, <b>login (513)</b>, <b>cmd (514)</b>, and <b>talk (517)</b>.</p>
<b>&lt;est&gt;</b>	When set to 1, specifies that the filter matches a packet only if a TCP session is already established. This argument is valid only when <b>&lt;proto&gt;</b> is set to <b>tcp (6)</b> .

"RADIUS Attribute 242 IP Data Filter Entries" shows four attribute 242 IP data filter entries.

### RADIUS Attribute 242 IP Data Filter Entries

```
Ascend-Data-Filter="ip in drop"
Ascend-Data-Filter="ip out forward tcp"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16
dstport!=telnet"
Ascend-Data-Filter="ip out forward tcp dstip 10.0.200.3/16 srcip 10.0.200.25/16 icmp"
```

## How to Configure Template ACLs

If ACLs are configured using RADIUS Attribute 242 or VSA Cisco-AVPairs, template ACLs are enabled by default.

### Configuring the Maximum Size of Template ACLs

By default, template ACL status is limited to ACLs with 100 or fewer rules. However, you can set this limit to a lower number. To set the maximum number of rules that an ACL may have in order to be considered as a template ACL, perform the steps in this section:

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list template *number***
4. **exit**
5. **show access-list template summary**

#### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>access-list template <i>number</i></b>  <b>Example:</b> Router(config)# access-list template 50	Enables template ACL processing.  Only ACLs with the specified number of rules (or fewer rules) will be considered for template status.

	Command or Action	Purpose
<b>Step 4</b>	<b>exit</b>  <b>Example:</b> Router(config)# exit	Exits global configuration mode.
<b>Step 5</b>	<b>show access-list template summary</b>  <b>Example:</b> Router# show access-list template summary	(Optional) Displays summary information about template ACLs.

## Troubleshooting Tips

The following commands can be used to troubleshoot the Template ACL feature:

- **show access-list template**
- **show platform hardware qfp active classification class-group-manager class-group client acl all**
- **show platform hardware qfp active feature acl {control | node acl-node-id}**
- **show platform software access-list**

# Configuration Examples for Template ACLs

## Example Maximum Size of Template ACLs

The following example shows how to set the maximum number of rules that an ACL may have in order to be considered for template status to 50. Only ACLs whose number of rules is the same as or smaller than 50 are considered for template status.

```
Router> enable

Router# configure terminal

Router(config)# access-list template 50
Router(config)# exit
```

## Example Showing ACL Template Summary Information

The following example shows how to view summary information for all ACLs in the system. The output from the command includes the following information:

- Maximum number of rules per template ACL

- Number of discovered active templates
- Number of ACLs replaced by those templates
- Number of elements in the Red-Black tree

```
Router# show access-list template summary
Maximum rules per template ACL = 100
Templates active = 9
Number of ACLs those templates represent = 14769
Number of tree elements = 13
```

### Red-Black Tree Elements

The number of tree elements is the number of elements in the Red-Black tree. Each template has 1 unique entry in the Red-Black tree. The system calculates a cyclic redundancy check (CRC) over each ACL masking out the peer IP address and puts the CRC into the Red-Black tree. For example:

Your system has 9 templates (representing 14769 ACLs), and 13 tree elements. If each template has only 1 unique entry in the Red-Black tree, then the additional 4 tree elements means that your system contains 4 per-user ACLs that are not templated.

## Example Showing ACL Template Tree Information

The following example shows how to view Red-Black tree information for all ACLs in the system.

The output from the command includes the following information:

- Name of the ACL on the Red-Black tree
- The original CRC32 value
- Number of users of the ACL
- Calculated CRC32 value

```
Router# show access-list template tree
ACL name      OrigCRC      Count  CalcCRC
4Temp_1073741891108  59DAB725    98  59DAB725
```

## Additional References

### Related Documents

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
IP access list commands	<i>Cisco IOS Security Command Reference</i>
Configuring IP access lists	"Creating an IP Access List and Applying It to an Interface"

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for ACL Template

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 16: Feature Information for ACL Template**

Feature Name	Releases	Feature Information
Template ACLs	12.2(28)SB 12.2(31)SB2 Cisco IOS XE Release 2.4	<p>In 12.2(28)SB, this feature was introduced on the Cisco 10000 series router.</p> <p>In 12.2(31)SB2, support was added for the PRE3.</p> <p>In Cisco IOS XE Release 2.4, this feature was implemented on the Cisco ASR 1000 series routers.</p> <p>The following commands were introduced or modified:<b>access-list template, show access-list template</b></p>







## IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp\_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.
- [Finding Feature Information, page 165](#)
- [Information About IPv6 ACL—Template ACL, page 166](#)
- [How to Enable IPv6 ACL—Template ACL, page 166](#)
- [Configuration Examples for IPv6 ACL—Template ACL, page 167](#)
- [Additional References, page 168](#)
- [Feature Information for IPv6 ACL—Template ACL, page 169](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To

find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Information About IPv6 ACL—Template ACL

### IPv6 Template ACL

When user profiles are configured using vendor-specific attribute (VSA) Cisco AV-pairs, similar per-user IPv6 ACLs may be replaced by a single template ACL. That is, one ACL represents many similar ACLs. By using IPv6 template ACLs, you can increase the total number of per-user ACLs while minimizing the memory and Ternary Content Addressable Memory (TCAM) resources needed to support the ACLs.

The IPv6 Template ACL feature can create templates using the following ACL fields:

- IPv6 source and destination addresses
- TCP and UDP, including all associated ports (0 through 65535)
- ICMP neighbor discovery advertisements and solicitations
- IPv6 DSCP with specified DSCP values

ACL names are dynamically generated by this feature; for example:

- 6Temp\_#152875854573--Example of a dynamically generated template name for a template ACL parent
- Virtual-Access2.32135#152875854573--Example of a child ACL or an ACL that has not yet been made part of a template.

## How to Enable IPv6 ACL—Template ACL

### Enabling IPv6 Template Processing

#### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list template** *[number-of-rules]*
4. **exit**
5. **show access-list template** {**summary** | *aclname* | **exceed** *number* | **tree**}

## DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Router> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>access-list template</b> [ <i>number-of-rules</i> ]  <b>Example:</b> Router(config)# access-list template 50	Enables template ACL processing.  <ul style="list-style-type: none"> <li>The example in this task specifies that ACLs with 50 or fewer rules will be considered for template ACL status.</li> <li>The <i>number-of-rules</i> argument default is 100.</li> </ul>
<b>Step 4</b>	<b>exit</b>  <b>Example:</b> Router(config)# exit	Exits global configuration mode and places the router in privileged EXEC mode.
<b>Step 5</b>	<b>show access-list template</b> { <b>summary</b>   <i>aclname</i>   <b>exceed</b> <i>number</i>   <b>tree</b> }  <b>Example:</b> Router# show access-list template summary	Displays information about ACL templates.

## Configuration Examples for IPv6 ACL—Template ACL

### Example: IPv6 Template ACL Processing

In this example, the contents of ACL1 and ACL2 are the same, but the names are different:

```

ipv6 access-list extended ACL1 (PeerIP: 2001:1::1/64)
permit igmp any 2003:1::1/64
permit icmp 2002:5::B/64 any
permit udp any host 2004:1::5
permit udp any host 2002:2BC::a
permit icmp host 2001:BC::7 host 2003:3::7

```

```

ipv6 access-list extended ACL2 (PeerIP: 2007:2::7/64)
permit igmp any 2003:1::1/64
permit icmp 2002:5::B/64 any
permit udp any host 2004:1::5
permit udp any host 2002:2BC::a
permit icmp host 2001:BC::7 host 2003:3::7

```

The template for these ACLs is as follows:

```

ipv6 access-list extended Template_1
permit igmp any 2003:1::1/64
permit icmp 2002:5::B/64 any
permit udp any host 2004:1::5
permit udp any host 2002:2BC::a
permit icmp host 2001:BC::7 host 2003:3::7

```

## Additional References

### Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
IPv6 commands	<i>Cisco IOS IPv6 Command Reference</i>
Cisco IOS IPv6 features	<a href="#">Cisco IOS IPv6 Feature Mapping</a>

### Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

### MIBs

MIB	MIBs Link
	<p>To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:</p> <p><a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a></p>

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for IPv6 ACL—Template ACL

*Table 17: Feature Information for IPv6 ACL—Template ACL*

Feature Name	Releases	Feature Information
IPv6 ACL—Template ACL	Cisco IOS XE Release 3.2S	This feature allows similar per-user IPv6 ACLs to be replaced by a single template ACL.  The following commands were introduced or modified: <b>access-list template</b> , <b>show access-list template</b> .





## IPv4 ACL Chaining Support

ACL Chaining, also known as Multi-Access Control List, allows you to split access control lists (ACLs). This module describes how with the IPv4 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

- [Finding Feature Information, page 171](#)
- [Restrictions for IPv4 ACL Chaining Support, page 171](#)
- [Information About IPv4 ACL Chaining Support, page 172](#)
- [How to Configure IPv4 ACL Chaining Support, page 172](#)
- [Configuration Examples for IPv4 ACL Chaining Support, page 174](#)
- [Additional References for IPv4 ACL Chaining Support, page 174](#)
- [Feature Information for IPv4 ACL Chaining Support, page 175](#)

## Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Restrictions for IPv4 ACL Chaining Support

- A single access control List (ACL) cannot be used for both common and regular ACLs for the same target in the same direction.
- ACL chaining applies to only security ACLs. It is not supported for feature policies, such as Quality of Service (QoS), Firewall Services Module (FW) and Policy Based Routing (PBR).

- Per-target statistics are not supported for common ACLs.

# Information About IPv4 ACL Chaining Support

## ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs
- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

## IPv4 ACL Chaining Support

IPv4 ACL Chaining Support allows you to split the Access control list (ACL) into common and customer-specific ACLs and attach both ACLs to a common session. In this way, only one copy of the common ACL is attached to Ternary Content Addressable Memory (TCAM) and shared by all users, thereby making it easier to maintain the common ACEs.

The IPv4 ACL Chaining feature allows two IPV4 ACLs to be active on an interface per direction:

- Common
- Regular
- Common and Regular

**Note**

If you configure both common and regular ACLs on an interface, the common ACL is considered over a regular ACL.

## How to Configure IPv4 ACL Chaining Support

ACL chaining is supported by extending the **ip traffic filter** command.

The **ip traffic filter** command is not additive. When you use this command, it replaces earlier instances of the command.



For more information, refer to the *IPv6 ACL Chaining with a Common ACL* section in the Security Configuration Guide: Access Control Lists Configuration Guide.

## Configuring an Interface to Accept Common ACL

Perform this task to configure the interface to accept a common Access control list (ACL) along with an interface-specific ACL:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface *type number***
4. **ip access-group {common {*common-access-list-name* {*regular-access-list* | **acl**}} {**in** | **out**}}**
5. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface <i>type number</i></b>  <b>Example:</b> Device(config)# interface gigabitethernet 0/0/0	Configures an interface (in this case a gigabitethernet interface) and enters the interface configuration mode.
<b>Step 4</b>	<b>ip access-group {common {<i>common-access-list-name</i> {<i>regular-access-list</i>   <b>acl</b>}} {<b>in</b>   <b>out</b>}}</b>  <b>Example:</b> Device(config)# ipv4 access-group common acl-p acl1 in	Configures the interface to accept a common ACL along with the interface-specific ACL.
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

# Configuration Examples for IPv4 ACL Chaining Support

This section provides configuration examples of Common Access Control List (ACL).

## Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an Access Control List (ACL) configured on the interface without explicitly deleting the ACL:

```
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl ACL2 in
end
```

This example shows how common ACL cannot be replaced on interfaces without deleting it explicitly from the interface:

```
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl1 ACL1 in
end
change the common acl to C_acl2
interface gigabitethernet 0/0/0
no ipv4 access-group common C_acl1 ACL1 in
end
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl2 ACL1 in
end
```

**Note**

When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.

**Note**

If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how the interface ACL is removed:

```
interface gigabitethernet 0/0/0
ipv4 access-group common C_acl1 ACL1 in
end
```

## Additional References for IPv4 ACL Chaining Support

### Related Documents

Related Topic	Document Title
IPv6 ACL Chaining Support	<a href="#">Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S</a>

Related Topic	Document Title
Cisco IOS commands	<a href="#">Cisco IOS Master Command List, All Releases</a>
Security commands	<ul style="list-style-type: none"> <li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li> <li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li> </ul>

### Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<a href="http://www.cisco.com/support">http://www.cisco.com/support</a>

## Feature Information for IPv4 ACL Chaining Support

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 18: Feature Information for IPv4 ACL Chaining Support**

Feature Name	Releases	Feature Information
IPv4 ACL Chaining Support	Cisco IOS XE Release 3.11S	<p>The IPv4 ACL Chaining Support feature describes how you can explicitly split Access control lists (ACLs) into common and user-specific ACLs and bind both ACLs to a session for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.</p> <p>The following commands were introduced or modified: <b>ip access-group</b> command.</p>



## IPv6 ACL Chaining with a Common ACL

ACL Chaining, also known as Multi-Access Control List (ACL), allows you to split ACLs. This document describes how with the IPv6 ACL Chaining Support feature, you can explicitly split ACLs into common and user-specific ACLs and bind both ACLs to a target for traffic filtering on a device. In this way, the common ACLs in Ternary Content Addressable Memory (TCAM) are shared by multiple targets, thereby reducing the resource usage.

- [Finding Feature Information, page 177](#)
- [Information About IPv6 ACL Chaining with a Common ACL, page 177](#)
- [How to Configure IPv6 ACL Chaining with a Common ACL, page 178](#)
- [Configuration Examples for IPv6 ACL Chaining with a Common ACL, page 180](#)
- [Additional References for IPv6 ACL Chaining with a Common ACL, page 181](#)
- [Feature Information for IPv6 ACL Chaining with a Common ACL, page 182](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Information About IPv6 ACL Chaining with a Common ACL

### ACL Chaining Overview

The packet filter process supports only a single Access control list (ACL) to be applied per direction and per protocol on an interface. This leads to manageability and scalability issues if there are common ACL entries

needed on many interfaces. Duplicate Access control entries (ACEs) are configured for all those interfaces, and any modification to the common ACEs needs to be performed for all ACLs.

A typical ACL on the edge box for an Internet Service Provider (ISP) has two sets of ACEs:

- Common ISP specific ACEs
- Customer/interface specific ACEs

The purpose of these address blocks is to deny access to ISP's protected infrastructure networks and anti-spoofing protection by allowing only customer source address blocks. This results in configuring unique ACL per interface and most of the ACEs being common across all ACLs on a device. ACL provisioning and modification is very cumbersome, hence, any changes to the ACE impacts every target.

## IPv6 ACL Chaining with a Common ACL

With IPv6 ACL Chaining, you can configure a traffic filter with the following:

- Common ACL
- Specific ACL
- Common and Specific ACL

Each Access control list (ACL) is matched in a sequence. For example, if you have specified both the ACLs - a common and a specific ACL, the packet is first matched against the common ACL; if a match is not found, it is then matched against the specific ACL.



### Note

Any IPv6 ACL may be configured on a traffic filter as a common or specific ACL. However, the same ACL cannot be specified on the same traffic filter as both common and specific.

## How to Configure IPv6 ACL Chaining with a Common ACL

### Before You Begin

IPv6 ACL chaining is configured on an interface using an extension of the existing IPv6 traffic-filter command: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] [**in** | **out**]



### Note

You may choose to configure either of the following:

- Only a common ACL. For example: **ipv6 traffic-filter common** *common-acl*
- Only a specific ACL. For example: **ipv6 traffic-filter** *common-acl*
- Both ACLs. For example: **ipv6 traffic-filter common** *common-acl specific-acl*

The **ipv6 traffic-filter** command is not additive. When you use the command, it replaces earlier instances of the command. For example, the command sequence: **ipv6 traffic-filter** [**common** *common-acl*] [*specific-acl*] **in** **ipv6 traffic-filter** [*specific-acl*] **in** binds a common ACL to the traffic filter, removes the common ACL and then binds a specific ACL.

## Configuring the IPv6 ACL to an Interface

Perform this task to configure the interface to accept a common access control list (ACL) along with an interface-specific ACL:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*}
4. **ipv6 traffic filter** {*common-access-list-name* {**in** | **out**}}
5. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode. Enter your password if prompted.
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>interface</b> <i>type number</i> }	Specifies the interface type and number, and enters interface configuration mode.
	<b>Example:</b>  Device(config)# interface gigabitethernet 0/0/0	
<b>Step 4</b>	<b>ipv6 traffic filter</b> { <i>common-access-list-name</i> { <b>in</b>   <b>out</b> }}	Applies the specified IPv6 access list to the interface specified in the previous step.
	<b>Example:</b>  Device(config)# ipv6 traffic-filter outbound out	
<b>Step 5</b>	<b>end</b>  <b>Example:</b> Device(config-if)# end	(Optional) Exits the configuration mode and returns to privileged EXEC mode.

# Configuration Examples for IPv6 ACL Chaining with a Common ACL

You may configure the following combinations in no particular order:

- A common ACL, for example: **ipv6 traffic-filter common *common-acl* in**
- A specific ACL, for example: **ipv6 traffic-filter *specific-acl* in**
- Both ACLs, for example: **ipv6 traffic-filter common *common-acl* *specific-acl* in**

## Example: Configuring an Interface to Accept a Common ACL

This example shows how to replace an access control list (ACL) configured on the interface without explicitly deleting the ACL:

```
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl ACL1 in
end
replace interface acl ACL1 by ACL2
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl ACL2 in
end
```

This example shows how to delete a common ACL from an interface. A common ACL cannot be replaced on interfaces without deleting it explicitly from the interface.

```
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl1 ACL1 in
end
change the common acl to C_acl2
interface gigabitethernet 0/0/0
no ipv6 access-group common C_acl1 ACL1 in
end
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl2 ACL1 in
end
```



### Note

When reconfiguring a common ACL, you must ensure that no other interface on the line card is attached to the common ACL.



### Note

If both common ACL and interface ACL are attached to an interface and only one of the above is reconfigured on the interface, then the other is removed automatically.

This example shows how to remove the interface ACL:

```
interface gigabitethernet 0/0/0
ipv6 access-group common C_acl1 ACL1 in
end
```



# Additional References for IPv6 ACL Chaining with a Common ACL

## Related Documents

Related Topic	Document Title
IPv4 ACL Chaining Support	<a href="#">Security Configuration Guide: Access Control Lists, Cisco IOS XE Release 3S</a>
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
Security commands	<ul style="list-style-type: none"><li>• <a href="#">Cisco IOS Security Command Reference: Commands A to C</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands D to L</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands M to R</a></li><li>• <a href="#">Cisco IOS Security Command Reference: Commands S to Z</a></li></ul>

## Technical Assistance

Description	Link
<p>The Cisco Support website provides extensive online resources, including documentation and tools for troubleshooting and resolving technical issues with Cisco products and technologies.</p> <p>To receive security and technical information about your products, you can subscribe to various services, such as the Product Alert Tool (accessed from Field Notices), the Cisco Technical Services Newsletter, and Really Simple Syndication (RSS) Feeds.</p> <p>Access to most tools on the Cisco Support website requires a Cisco.com user ID and password.</p>	<a href="http://www.cisco.com/support">http://www.cisco.com/support</a>

## Feature Information for IPv6 ACL Chaining with a Common ACL

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 19: Feature Information for IPv6 ACL Chaining with a Common ACL**

Feature Name	Releases	Feature Information
IPv6 ACL Chaining with a Common ACL	Cisco IOS XE Release 3.11S	<p>The ACL Chaining feature, also known as Multi-ACLs, allows you to explicitly split IPv6 traffic filter access control lists (ACLs) into common and per-session ACLs. In this way, the common access control entries (ACEs) that are used reduces resource usage of each ACL entry per session in the Ternary Content Addressable Memory (TCAM).</p> <p>The following commands were introduced or modified: <b>ip access-group common</b>.</p>



## IPv6 ACL Extensions for Hop by Hop Filtering

The IPv6 ACL Extensions for Hop by Hop Filtering feature allows you to control IPv6 traffic that might contain hop-by-hop extension headers. You can configure an access control list (ACL) to deny all hop-by-hop traffic or to selectively permit traffic based on protocol.

- [Finding Feature Information, page 183](#)
- [Information About IPv6 ACL Extensions for Hop by Hop Filtering, page 183](#)
- [How to Configure IPv6 ACL Extensions for Hop by Hop Filtering, page 184](#)
- [Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering, page 185](#)
- [Additional References, page 186](#)
- [Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering, page 187](#)

### Finding Feature Information

Your software release may not support all the features documented in this module. For the latest caveats and feature information, see [Bug Search Tool](#) and the release notes for your platform and software release. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the feature information table at the end of this module.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

## Information About IPv6 ACL Extensions for Hop by Hop Filtering

### ACLs and Traffic Forwarding

IPv6 access control lists (ACLs) determine what traffic is blocked and what traffic is forwarded at device interfaces. ACLs allow filtering based on source and destination addresses, inbound and outbound to a specific interface. Use the **ipv6 access-list** command to define an IPv6 ACL, and the **deny** and **permit** commands to configure its conditions.

The IPv6 ACL Extensions for Hop by Hop Filtering feature implements RFC 2460 to support traffic filtering in any upper-layer protocol type.

# How to Configure IPv6 ACL Extensions for Hop by Hop Filtering

## Configuring IPv6 ACL Extensions for Hop by Hop Filtering

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ipv6 access-list *access-list-name***
4. **permit protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**reflect** *name*] [**timeout** *value*] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*]
5. **deny protocol** {*source-ipv6-prefix/prefix-length* | **any** | **host** *source-ipv6-address* | **auth**} [*operator* [*port-number*]] {*destination-ipv6-prefix/prefix-length* | **any** | **host** *destination-ipv6-address* | **auth**} [*operator* [*port-number*]] [**dest-option-type** [*header-number* | *header-type*]] [**dscp** *value*] [**flow-label** *value*] [**fragments**] [**hbh**] [**log**] [**log-input**] [**mobility**] [**mobility-type** [*mh-number* | *mh-type*]] [**routing**] [**routing-type** *routing-number*] [**sequence** *value*] [**time-range** *name*] [**undetermined-transport**]
6. **end**

### DETAILED STEPS

	Command or Action	Purpose
<b>Step 1</b>	<b>enable</b>  <b>Example:</b> Device> enable	Enables privileged EXEC mode.  <ul style="list-style-type: none"> <li>• Enter your password if prompted.</li> </ul>
<b>Step 2</b>	<b>configure terminal</b>  <b>Example:</b> Device# configure terminal	Enters global configuration mode.
<b>Step 3</b>	<b>ipv6 access-list <i>access-list-name</i></b>  <b>Example:</b> Device(config)# ipv6 access-list hbh-acl	Defines an IPv6 ACL and enters IPv6 access list configuration mode.
<b>Step 4</b>	<b>permit protocol</b> { <i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>   <b>auth</b> } [ <i>operator</i> [ <i>port-number</i> ]] { <i>destination-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>	Sets permit conditions for the IPv6 ACL.

	Command or Action	Purpose
	<b>auth</b> { <i>operator</i> [ <i>port-number</i> ] } [ <b>dest-option-type</b> [ <i>header-number</i>   <i>header-type</i> ] ] [ <b>dscp</b> <i>value</i> ] [ <b>flow-label</b> <i>value</i> ] [ <b>fragments</b> ] [ <b>hbh</b> ] [ <b>log</b> ] [ <b>log-input</b> ] [ <b>mobility</b> ] [ <b>mobility-type</b> [ <i>mh-number</i>   <i>mh-type</i> ] ] [ <b>reflect</b> <i>name</i> ] [ <b>timeout</b> <i>value</i> ] ] [ <b>routing</b> ] [ <b>routing-type</b> <i>routing-number</i> ] [ <b>sequence</b> <i>value</i> ] [ <b>time-range</b> <i>name</i> ]  <b>Example:</b> Device(config-ipv6-acl)# permit icmp any any dest-option-type	
<b>Step 5</b>	<b>deny</b> <i>protocol</i> { <i>source-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>source-ipv6-address</i>   <b>auth</b> } [ <i>operator</i> [ <i>port-number</i> ] ] { <i>destination-ipv6-prefix/prefix-length</i>   <b>any</b>   <b>host</b> <i>destination-ipv6-address</i>   <b>auth</b> } [ <i>operator</i> [ <i>port-number</i> ] ] [ <b>dest-option-type</b> [ <i>header-number</i>   <i>header-type</i> ] ] [ <b>dscp</b> <i>value</i> ] [ <b>flow-label</b> <i>value</i> ] [ <b>fragments</b> ] [ <b>hbh</b> ] [ <b>log</b> ] [ <b>log-input</b> ] [ <b>mobility</b> ] [ <b>mobility-type</b> [ <i>mh-number</i>   <i>mh-type</i> ] ] [ <b>routing</b> ] [ <b>routing-type</b> <i>routing-number</i> ] [ <b>sequence</b> <i>value</i> ] [ <b>time-range</b> <i>name</i> ] [ <b>undetermined-transport</b> ]  <b>Example:</b> Device(config-ipv6-acl)# deny icmp any any dest-option-type	Sets deny conditions for the IPv6 ACL.
<b>Step 6</b>	<b>end</b>  <b>Example:</b> Device (config-ipv6-acl)# end	Returns to privileged EXEC configuration mode.

## Configuration Example for IPv6 ACL Extensions for Hop by Hop Filtering

### Example: IPv6 ACL Extensions for Hop by Hop Filtering

```

Device(config)# ipv6 access-list hbh_acl
Device(config-ipv6-acl)# permit tcp any any hbh
Device(config-ipv6-acl)# permit tcp any any
Device(config-ipv6-acl)# permit udp any any
Device(config-ipv6-acl)# permit udp any any hbh
Device(config-ipv6-acl)# permit hbh any any
Device(config-ipv6-acl)# permit any any
Device(config-ipv6-acl)# hardware statistics
Device(config-ipv6-acl)# exit

! Assign an IP address and add the ACL on the interface.

Device(config)# interface FastEthernet3/1
Device(config-if)# ipv6 address 1001::1/64

```

```

Device(config-if)# ipv6 traffic-filter hbh_acl in
Device(config-if)# exit
Device(config)# exit
Device# clear counters
Clear "show interface" counters on all interfaces [confirm]
Device#

```

! Verify the configurations.

```
Device# show running-config interface FastEthernet3/1
```

Building configuration...

```

Current configuration : 114 bytes
!
interface FastEthernet3/1
no switchport
ipv6 address 1001::1/64
ipv6 traffic-filter hbh_acl
end

```

## Additional References

### Related Documents

Related Topic	Document Title
IPv6 addressing and connectivity	<i>IPv6 Configuration Guide</i>
Cisco IOS commands	<a href="#">Cisco IOS Master Commands List, All Releases</a>
IPv6 commands	<i>Cisco IOS IPv6 Command Reference</i>
Cisco IOS IPv6 features	<a href="#">Cisco IOS IPv6 Feature Mapping</a>

### Standards and RFCs

Standard/RFC	Title
RFCs for IPv6	<i>IPv6 RFCs</i>

### MIBs

MIB	MIBs Link
	To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL: <a href="http://www.cisco.com/go/mibs">http://www.cisco.com/go/mibs</a>

**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.	<a href="http://www.cisco.com/cisco/web/support/index.html">http://www.cisco.com/cisco/web/support/index.html</a>

## Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering

The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train also support that feature.

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to [www.cisco.com/go/cfn](http://www.cisco.com/go/cfn). An account on Cisco.com is not required.

**Table 20: Feature Information for IPv6 ACL Extensions for Hop by Hop Filtering**

Feature Name	Releases	Feature Information
IPv6 ACL Extensions for Hop by Hop Filtering	Cisco IOS Release XE 3.4S Cisco IOS Release XE 3.5S Cisco IOS Release XE 3.6S Cisco IOS Release XE 3.3SG	Allows you to control IPv6 traffic that might contain hop-by-hop extension headers.  The following commands were introduced or modified: <b>deny</b> (IPv6), <b>permit</b> (IPv6).

