



# Access Control List Enhancements on the Cisco 12000 Series Router

---

Part Number OL-15425-01, May 30, 2008

The Cisco 12000 series router filters IP packets using access control lists (ACLs) as a fundamental security feature. This document describes the following ACL enhancements for IPv4 traffic. These enhancements optimize the use of ACLs to control packet transmission and restrict network use by certain users or devices:

- Named ACLs—Allow you to identify ACLs for IPv4 traffic with a name and a number, and provide the following benefits:
  - Support access list entry (ACE) sequence-numbering, which allows you to apply sequence numbers to **permit** or **deny** statements and reorder, add, or remove these statements from a named IP access list. This feature eases your revisions to IP access lists. Earlier than this feature, you could only add, permit, or deny ACEs to the end of an access list. Adding an ACE in locations other than at the end of an access list required that you reconfigure the entire access list.)
  - Avoid the limitation on the maximum number of supported ACLs that exists for numbered ACLs.
- Time-based access control entries (ACEs)—Allow you to control the time during which IPv4 packets are permitted or denied access from specific network resources.
- Time-to-Live (TTL) access control entries—ACEs that specify a TTL value allow you to mitigate Denial of Service (DoS) attacks on the router from a variety of spoofed packets by permitting or denying IPv4 packets based on the TTL value in the packet header.

These ACL enhancements are supported:

- On the interfaces and subinterfaces of line cards with distributed switch engines (IP Services Engine/Engine 3 and Engine 5). Distributed switch engines perform high-speed switching of IP packets for all ports on a line card without using resources from the central switch engine of the route processor (RP).
- As IP receive ACLs to filter IPv4 packets traveling to the RP (only on engine 3 and engine 5) as described in [IP Receive ACL](#).



---

**Americas Headquarters:**  
**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

© 2008 Cisco Systems, Inc. All rights reserved.

- In quality-of-service (QoS) policies used for aggregate and distributed control plane policing of IPv4 packets sent to the RP from all line-card interfaces on the router (as described in [Control Plane Policing](#)).

#### Finding Feature Information

Your Cisco IOS software release may not support all of the features documented in this guide. To reach links to specific feature documentation in this module and to see a list of the releases in which each feature is supported, use the “[Feature Information for ACL Enhancements on the Cisco 12000 Series Router](#)” section on page 70.

#### Finding Support Information for Platforms and Cisco IOS Software Images

Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions that appear.

## Contents

- [Prerequisites for ACL Enhancements on the Cisco 12000 Series Router](#), page 2
- [Restrictions for ACL Enhancements on the Cisco 12000 Series Router](#), page 3
- [ACL Enhancements on the Cisco 12000 Series Router](#), page 5
- [Configuring ACL Enhancements on the Cisco 12000 Series Router](#), page 12
- [Configuration Examples for ACL Enhancements on the Cisco 12000 Series Router](#), page 27
- [Additional References](#), page 32
- [Command Reference](#), page 35
- [Feature Information for ACL Enhancements on the Cisco 12000 Series Router](#), page 70

## Prerequisites for ACL Enhancements on the Cisco 12000 Series Router

#### Modular Quality of Service Command-Line Interface

To use ACL enhancements in a service policy applied to control plane traffic, you should understand the concepts and general configuration procedure (class map and policy map) for applying quality-of-service (QoS) policies on a router.

For information about Cisco IOS QoS and the procedure for configuring QoS in your network using the modular QoS command-line interface (MQC), refer to [Cisco IOS Quality of Service Solutions Configuration Guide](#), Release 12.3.

# Restrictions for ACL Enhancements on the Cisco 12000 Series Router

## IPv6 Not Supported

The ACL enhancements described in this document—named ACLs, sequenced ACEs, time-based ACEs, and TTL-based ACEs—are supported only for IPv4 packet filtering. IPv6 traffic is not supported.

## Quality of Service

The ACL enhancements described in this document—named ACLs, sequenced ACEs, time-based ACEs, and TTL-based ACEs—are supported only in the following types of QoS policies:

- As IP receive ACLs to filter IPv4 packets destined for the RP (only on Engine 3 and Engine 5) as described in *IP Receive ACL*
- In quality-of-service (QoS) policies used for aggregate control plane policing and distributed control plane policing of IPv4 packets sent to the RP from all line-card interfaces on the router (as described in *Control Plane Policing*)

ACL enhancements are not supported in QoS policies applied to interfaces or subinterfaces.

## Supported Line Cards

The ACL enhancements described in this document—named ACLs, sequenced ACEs, time-based ACEs, and TTL-based ACEs—are supported only on the following engine types and interfaces on the Cisco 12000 series router:

- IP Services Engine (ISE or Engine 3) and Engine 5 line cards, except for:
  - ISE 2-port clear channel T3/E3 (serial) shared port adapters (SPAs)
  - ISE 4-port clear channel T3/E3 (serial) SPAs
  - ISE 2-port channelized T3 (CT3) to DS0 SPAs
  - ISE 4-port channelized T3 (CT3) to DS0 SPAs
- ATM, channelized, Fast Ethernet (FE), Gigabit Ethernet (GE), and Packet over SONET (POS) interfaces
- Link bundling (EtherChannel and POS channel) interfaces on E3 and E5 line cards
- Distributed control-plane policing interfaces on E3 and E5 line cards

## Time-based Access Control Entries

ACEs (**permit** and **deny** statements) that specify a time range are supported only for IPv4 traffic and only in extended (named or numbered) ACLs on ingress or egress interfaces on ISE/E3 or E5 line cards. By default, ACE with **time-range** option is active.

To configure a time-based ACE, you enter the **time-range** *time-range-name* parameter in a **permit** or **deny** statement.

The **time-range** configuration in a **permit** or **deny** statement is not supported in the following situations:

- When used in a reflexive ACL
- When used in a merged ACL
- On a Dial on Demand Routing (DDR) link




---

**Note** Only one absolute entry is allowed for each time-range entry. If the range time has both absolute and periodic values specified, then the periodic values are only evaluated *after* the start time is reached and *before* the end time is reached.

---

### Disabling Merging on the Line Card

To use time-based ACEs in a merged ACL, you must first disable merging on the Cisco 12000 series line card. To disable ACL merging, follow these steps:

- 
- Step 1** Enter the **hw-module slot *slot-number* tcam compile acl no-merge** command in global configuration mode.
- Step 2** Re-apply the ACL with time-based ACEs on the appropriate interface by entering the **ip access-group *access-list-name* {in | out}** command in interface configuration mode.




---

**Note** Time-based ACEs are not supported in Turbo ACLs.

---

The following restrictions apply to clock synchronization between the route processor (RP) and Cisco 12000 series line cards with switch engines:

- When there is a change in time on the RP clock, a delay in synchronizing the time on the line-card clock occurs. Time-based ACEs are enabled and disabled according to the slightly different time on the line-card clock.
- Time-based ACEs are supported in Stateful Switchover (SSO), route processor redundancy (RPR), and route processor redundancy plus (RPR+) mode. However, in RPR mode, the clocks on the active and standby RP are not synchronized.

In general, the exact time when a time-based ACE is enabled and disabled varies from the configured time depending on CPU utilization, number and length of ACLs being processed, and number of TCAM entries that correspond to an ACE. When you configure time ranges for **permit** and **deny** statements, take these factors into account.

### Time-to-Live Access Control Entries

ACEs (**permit** and **deny** statements) that specify a time-to-live (TTL) value are supported only for IPv4 traffic and only in extended (named or numbered) ACLs on ingress-direction interfaces on ISE/E3 or E5 line cards. TTL-based ACEs are not supported in the egress direction.

To fully implement TTL-based filtering for IPv4 traffic on a Cisco 12000 series router, you must configure:

- ACLs with TTL-based ACEs on the appropriate ISE (Engine 3) and Engine 5 interfaces. (Named ACLs are supported; see the [“Using Named Instead of Numbered Access Lists”](#) section on page 8.)
- Aggregate control plane policing for a class of ingress traffic that filters TTL values of 0 and 1

### NetFlow Compatibility

Legacy NetFlow and Flexible NetFlow features are not supported on an interface that uses an ACL with ACEs for TTL filtering. TTL filtering is performed on a per-packet basis, not per flow. For TTL-based filtering to work properly, you must disable NetFlow.




---

**Note** ACLs that specify a TTL value are not supported in Turbo ACLs.

---

### Using Named Access Control Lists

Named ACLs are supported only for IPv4 traffic:

- On the interfaces and subinterfaces of line cards with distributed switch engines (IP services Engine/Engine 3 and Engine 5). Distributed switch engines perform high-speed switching of IP packets for all ports on a line card without using resources from the central switch engine of the route processor (RP).
- As IP receive ACLs to filter IPv4 packets destined for the RP (only on Engine 3 and Engine 5) as described in [IP Receive ACL](#)
- In quality-of-service (QoS) policies used for aggregate control plane policing and distributed control plane policing of IPv4 packets sent to the RP from all line-card interfaces on the router (as described in [Control Plane Policing](#))



---

**Note** Named ACLs are not supported in Engine 2 and Engine 4/4+.

---

Named access lists are not recognized by releases earlier than Cisco IOS Release 11.2.

The name you specify for a named ACL with the **ip access-list** command can be either an alphanumeric string that starts with an alphabetic character or a number. No blank spaces are supported in an ACL name.

Unlike a numbered ACL configuration, you configure a named ACL in named access list configuration mode after you enter the **ip access-list** command. From this command mode, you must define the access conditions using the **deny** and **permit** commands.



**Caution**

---

Each standard and extended named ACL that you configure using the **ip access-list** command must have a unique name. You cannot use the same name for both a standard ACL and an extended ACL.

---

Named ACLs are not supported:

- In QoS policies applied to interfaces or subinterfaces on ISE and Engine 5 line cards.
- For policy-based routing (PBR).
- In dynamic and reflexive ACLs.

The **remark** command is not supported in a named ACL. The **remark** command allows you to enter a comment about a deny or permit statement.

## ACL Enhancements on the Cisco 12000 Series Router

To configure and use the ACL enhancements described in this document, you should understand the following concepts:

- [“Using Access Control Lists” section on page 6](#)
- [“Using Named Instead of Numbered Access Lists” section on page 8](#)
- [“Using Time-Based Access List Entries” section on page 10](#)
- [“Using Time-to-Live Access List Entries” section on page 10](#)

## Using Access Control Lists

Access control lists (ACLs, also referred to as *access lists*) perform packet filtering to control which packets move through the network. Packet filtering limits network traffic and restricts network use by certain users or devices.

Many commands accept a reference to an access list in the command syntax because 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
- Control label distribution
- Control peering selection
- 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

An access list is a sequential collection of **permit** and **deny** statements that apply to IP addresses and upper-level IP protocols. You create an access list by specifying an access list number or name and access conditions.

Many software commands accept an access list as part of their syntax. After you configure an access list, it is not in effect until the access list is referenced by a command that accepts an ACL. Multiple commands can reference the same access list. An access list can control traffic arriving at the router or leaving the router, but not traffic originating at the router.

1. Cisco IOS 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.

The first match determines whether the software accepts or rejects the packet.

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

2. If no conditions match, the software drops the packet. 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. Also, if you remove all entries from an ACL, it is considered to be void (null) and all traffic is permitted.

When access lists process packets, the following rules apply:

- If an access list is referenced by name or number 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 router. 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 router. 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.
- Cisco IOS software supports the following types of access lists for IP packet filtering:
- Standard IP access lists that use source addresses for matching operations.
- Extended IP access lists that use source and destination addresses for matching operations, and optional protocol type information for finer filtering.
- Dynamic extended IP access lists that grant access per user to a specific source or destination host basis through a user authentication process. You can allow user access through a firewall dynamically, without compromising security restrictions. Dynamic access lists and lock-and-key access are described in “[Configuring Lock-and-Key Security \(Dynamic Access Lists\)](#)” in the *Cisco IOS Security Configuration Guide*.
- Reflexive access lists that allow IP packets to be filtered based on session information. Reflexive access lists contain temporary entries, and are nested within an extended, named IP access list. For information on reflexive access lists, refer to the “[Configuring IP Session Filtering \(Reflexive Access Lists\)](#)” chapter in the *Cisco IOS Security Configuration Guide* and the “[Reflexive Access List Commands](#)” chapter in the *Cisco IOS Security Command Reference*.
- The Turbo Access Control List (Turbo ACL) feature processes access lists more expediently than conventional access lists. This feature enables a router to evaluate ACLs for more expedient packet classification and access checks.

ACLs are normally searched sequentially to find a matching rule, and ACLs are ordered specifically to take this factor into account. Because of the increasing needs and requirements for security filtering and packet classification, ACLs can expand to the point that searching the ACL adds a substantial amount of time and memory when packets are being forwarded. Moreover, the time taken by the router to search the list is not always consistent, adding a variable latency to the packet forwarding. A high CPU load is necessary for searching an access list with several entries.

For information about creating and using different types of access lists, refer to:

- “[Configuring IP Services](#)” chapter in *Cisco IOS IP Configuration Guide*, Release 12.3
- “[Access Control Lists: Overview and Guidelines](#)” chapter in *Cisco IOS Security Configuration Guide*, Release 12.3

In addition to using access lists for IP packet filtering, you can also configure context-based access control (CBAC). CBAC intelligently filters TCP and UDP packets based on application-layer protocol session information and can be used for intranets, extranets and internets. Without CBAC, traffic filtering is limited to access list implementations that examine packets at the network layer, or at most, the transport layer. CBAC inspects traffic that travels through the firewall to discover and manage state information for TCP and UDP sessions. For CBAC, refer to the “[Configuring Context-BAsed Access Control](#)” chapter in *Cisco IOS IP Configuration Guide*, Release 12.2

The following example shows how to create a numbered access list. In this example, network 36.0.0.0 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 36.0.0.0 address specify a particular host. Using access list 2, the Cisco IOS software accepts one address on subnet 48 and rejects all others on that subnet. The last line of the list shows that the software accepts addresses on all other network 36.0.0.0 subnets.

```
access-list 2 permit 36.48.0.3
access-list 2 deny 36.48.0.0 0.0.255.255
access-list 2 permit 36.0.0.0 0.255.255.255
interface ethernet 0
 ip access-group 2 in
```

## Using Named Instead of Numbered Access Lists

You can use access lists to:

- Control the transmission of packets on an interface
- Control vty access, and
- Restrict the contents of routing updates

When you define a standard or extended numbered IP access list, you use the standard version of the **access-list** command in global configuration mode. To define an IP access list by name, use the **ip access-list** command and enter named access list configuration mode to configure **permit** and **deny** ACEs.



### Note

When you create a named ACL **ip access-list** command, you can also specify a number instead of a text string as the name. A numbered ACL configured in named access list (NACL) configuration mode supports the ability to add, delete, and resequence ACEs using the **ip access-list** command while a traditional numbered ACL configured with the **access-list** command in global configuration mode does not support this feature.

Named access lists identify IP access lists with an alphanumeric string (a name) or number and are not restricted to a limited access-list range (for example, 1 to 99 or 100 to 199) as numbered lists are. By using named access lists, you have more flexibility and can configure more IP access lists in a router than if you use numbered access lists. Also, named ACLs support a simplified way to add, delete, and resequence ACEs using sequence numbers; for more information, see the [“Applying Sequence Numbers to Entries in a Named Access List”](#) section on page 9.

Before you configure named access lists, note the following guidelines:

- Access lists specified by name are not compatible with releases earlier than Cisco IOS Release 11.2.
- Not all access lists that accept a number accept a name. Access lists for packet filters and route filters on interfaces can use a name.
- A standard access list and an extended access list cannot have the same name.

Starting in Cisco IOS Release 12.0(33)S, named ACLs are downloaded to all line cards and named ACL checking is supported in the forwarding path. As a result, when you use named ACLs on ISE and engine 5 line cards, you can take advantage of the performance benefits of distributed Cisco Express Forwarding (CEF) switching.

You can use a named ACL for:

- IPv4 packet filtering
- IPv4 distributed control plane policing
- IPv4 receive ACLs
- IPv4 aggregate control plane policing
- IPv6 packet filtering
- IPv6 aggregate control plane policing

## Applying Sequence Numbers to Entries in a Named Access List

One of the main benefits in using an ACL created in named access list configuration mode using the **ip access-list** command is that you can take advantage of the IP Access List Entry Sequence Numbering feature. This feature is not supported in traditional numbered ACLs configured with the **access-list** command. ACE sequence numbering simplifies the process for adding and removing IP access list entries, and allows you to apply sequence numbers to unsequenced lists and resequence entries as needed to add new ones.

When you add a new entry to a named ACL, you can specify the sequence number so that each **deny** or **permit** statement is entered in its desired position in the access list. If necessary, you can resequence entries in the access list to create room to insert new entries.

Sequence numbering of ACEs in a named ACL is implemented as follows:

- For backward compatibility with previous releases, if you enter a statement without a sequence number applied:
  - If no entries exist in the ACL, the statement is assigned a sequence number of 10, and successive entries are increased by 10.
  - If sequenced entries exist in the ACL, the statement is assigned a sequence number of 10 more than the maximum sequence number. The maximum sequence number is 2147483647. If a generated sequence number exceeds this maximum number, the following message appears:  
`Exceeded maximum sequence number.`
- If you enter a statement that:
  - Matches an existing entry (except for the sequence number), the following error message appears:  
`Duplicate access rule.`
  - With an existing sequence number, the following error message appears:  
`Duplicate sequence number.`
- If you enter statements in a new access list from global configuration mode, sequence numbers for the access list are generated automatically.
- Distributed support is provided so that the sequence numbers of entries in the route processor (RP) and line card are in synchronization at all times.
- Sequence numbers are not nvgened; that is, the sequence numbers themselves are not saved and do not appear in the results displayed by the **show running-config** command. 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.

However, if Stateful Switchover (SSO) or route processor redundancy plus (RPR+) modes are configured on the router, the active and standby RPs are synchronized and sequence numbers in named ACLs are saved.

- Sequence numbering of ACEs is supported only in standard and extended IP access lists configured in named access list configuration mode with the **ip access-list** command. Old-style numbered access lists, which existed before named access lists and are configured with the **access-list** command, do not support the IP Access List Entry Sequence Numbering feature.

However, because you can specify a number as the name of a named access list, a numbered ACL that you configured in named access list (NACL) configuration mode supports the ability to add, delete, and resequence ACEs using sequence numbers.

For more information on how to use sequence numbering to simplify the management of named ACLs, refer to [IP Access List Entry Sequence Numbering](#).

## Using Time-Based Access List Entries

Time-based ACL entries (ACEs) allow you to control the time during which IPv4 packets from specific network resources are permitted or denied by the router. You configure a time-based ACE when you enter **permit** or **deny** statements in an ACL.

Time-based ACEs allow you to determine the usage of network resources for a specified time on specified weekdays or weekends. For example, the resource can be an application identified by its source or destination IP address, network mask, and port number.

To configure a time-based ACE, you enter the **time-range** *time-range-name* parameter in a **permit** or **deny** statement. For more information, see [Defining a Time Range for Time-Based Access Control Entries](#), page 22.

The exact time when a time-based ACE is enabled and disabled varies from the configured time depending on CPU utilization, number and length of ACLs being processed, and number of TCAM entries that correspond to an ACE. Take these factors into account when you configure time ranges for **permit** and **deny** statements.

The **time-range** configuration in a **permit** or **deny** statement is not supported when used in:

- A reflexive ACL
- An ACL that is merged with other ACLs from other software features applied to the interface

For a list of other restrictions in using time-based ACEs, see the [“Time-based Access Control Entries” section on page 3](#).



### Note

Time-based ACEs use the time and date kept on the software-based system clock on the router. You can either manually configure the time on the system clock or have it synchronized by the time kept on a Network Time Protocol (NTP) server. For information about how to set time-of-day services on a router, refer to [Performing Basic System Management](#).

## Using Time-to-Live Access List Entries

ACEs that specify a time-to-live (TTL) value allow you to mitigate Denial of Service (DoS) attacks on the router from a variety of spoofed packets by dropping suspicious packets that do not match a specified TTL value. You configure a time-to-live ACE when you enter **permit** or **deny** statements in an ACL.

ACEs that filter traffic based on a TTL value are useful in the following situations:

- Many DOS attacks seek to spoof the source address and source port, but not the time-to-live field in an IPv4 packet header. By specifying a TTL value in the **deny** statement, you can mitigate the attack.
- Many control plane protocols only communicate to their neighbors but listen to all network devices. To block attacks trying to enter as spoofed control-plane protocol packets, you can specify a TTL value in the **permit** statement of an ACL on a router that receives BGP network traffic. The TTL entry filters all packets to ensure receive path protection for BGP.

ACEs (**permit** and **deny** statements) that filter the TTL value in IPv4 packet headers are supported in extended named and numbered ACLs that are applied in the ingress direction on an ISE, E5, or control-plane interface. TTL-based ACEs are not supported in the egress direction.

You can use TTL-based ACEs in combination with other ACL features for:

- IPv4 packet filtering
- IPv4 distributed and aggregate control plane policing (see [Control Plane Policing](#))
- IPv4 receive ACLs (see [P Receive A CL](#))

The TTL value in the header of each IPv4 packet is checked against the specified TTL value, which can be from 0 to 255. Then a permit or deny decision is made when an ACE match is found. An ACL may also filter IPv4 packet headers according to other configured ACE values, such as Layer 3 protocol, source address, destination address, Layer 4 protocol, or port number.

**Note**

Legacy NetFlow and Flexible NetFlow features are not supported on an interface that uses an ACL with ACEs for TTL filtering. TTL filtering is performed on a per-packet basis, not per flow. You must disable NetFlow in order for TTL-based filtering to work properly.

**ACL Filtering of TTL Values 0 and 1 on Ingress Interfaces**

All possible TTL values (0 to 255) are examined in the distributed switching path on an ISE or Engine 5 line card. However, on an ingress interface, incoming IPv4 packets with a TTL value of 0 or 1 are punted to the process level of the route processor before the ingress ACL is checked. IPv4 packets with TTL values of 2 to 255 are first filtered by the ACL applied to the ingress interface; when a packet is matched, a permit or deny decision is made.

Packets with a TTL value of 0 or 1 always get punted to the process level because they are never forwarded out of the router. In the fast-path distributed packet switching performed on an ISE or Engine 5 line card, an ingress ACL configured to deny packets with TTL values of 0 or 1 still does not drop the packets. As a result, the CPU uses more resources, which may result in degraded router performance depending on the number of packets punted to the route processor.

To implement TTL-based filtering for IPv4 traffic in a way that fully prevents attacks on the CPU of the router, you must also configure ACEs that filter TTL values 0 and 1 in the ACL used for aggregate control plane policing on incoming IPv4 traffic.

For an example of how to configure both an ACL that filters TTL values 0 to 255 in incoming IPv4 packets and an ACL that filters TTL values 0 and 1 in all incoming IPv4 packets punted to the route processor, see the [“Configuring TTL-Based Access Control Entries: Example”](#) section on page 30.

# Configuring ACL Enhancements on the Cisco 12000 Series Router

This section documents the following procedures:

- “Configuring Named Access Control Lists” section on page 12
- “Verifying Named Access Control Lists” section on page 19
- “Configuring Time-Based Access Control Entries” section on page 20
- “Defining a Time Range for Time-Based Access Control Entries” section on page 22
- “Configuring TTL-Based Access Control Entries” section on page 23
- “Verifying Time-Based and TTL-Based Access Control Entries” section on page 26



**Note** For information on the use of any of the ACL commands described in this section, refer to *Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services*.

## Configuring Named Access Control Lists

This section describes how to create a named ACL for IPv4 packet filtering in named access list configuration mode. Procedures for configuring standard and extended access lists using names or numbers are described.

### Standard and Extended Named Access Lists

By using the procedures described in this section, you can create the following types of named ACLs:

- Standard access lists for IPv4 traffic that use source addresses for matching operations
- Extended access lists for IPv4 traffic that use source and destination addresses for matching operations, and optional protocol type information for finer granularity of control
- Standard or extended access lists with sequenced access list entries (**permit** and **deny** statements)

Named ACLs are not supported in dynamic and reflexive ACLs.

### Benefits

Starting in Cisco IOS Release 12.0(33)S, you can configure named ACLs on ISE and Engine 5 line cards that use distributed switch engines. Distributed named ACLs provide the following benefits:

- Named ACLs support the IP Access List Entry Sequence Numbering feature, which is not supported in traditional numbered ACLs configured with the **access-list** command. This feature allows you to apply sequence numbers to IP access list entries (**permit** and **deny** statements) and resequence the entries, and greatly simplifies the process for adding and removing access list entries.
- Named ACLs allow you to configure more IP access lists in a router than if you used numbered access lists.
- On ISE/Engine 3 and Engine 5 line cards, named ACLs are downloaded to all line cards and named ACL checking is supported in the forwarding path. As a result, filtering IPv4 packets using named ACLs takes advantage of the performance benefits of distributed CEF switching.

- Named ACLs on distributed switch-engine interfaces are supported in IPv4 receive ACLs and in the QoS policies applied to the control plane for distributed control plane policing. For more information, refer to [IP Receive ACL](#) and [Control Plane Policing](#).

## Configuring a Standard or Extended Named Access Control List

Perform this task if you want to create a standard or extended named ACL that uses only source addresses or source and destination addresses (as well as optional IPv4 information) to filter IPv4 traffic.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list {standard | extended} access-list-name**
4. *[sequence-number] permit source source-wildcard*  
or  
*[sequence-number] permit protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
5. *[sequence-number] deny source source-wildcard*  
or  
*[sequence-number] deny protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]*
6. Repeat Step 5 and/or Step 6 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-list access-list-name**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> <b>enable</b>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# <b>configure terminal</b>	Enters global configuration mode.

Command or Action	Purpose
<p><b>Step 3</b></p> <pre>ip access-list {standard   extended} access-list-name</pre> <p><b>Example:</b></p> <pre>Router(config)# ip access-list standard Internet_filter Router(config-std-nacl)#</pre>	<p>Defines an IP access list by name and enters named access list configuration mode.</p> <ul style="list-style-type: none"> <li>If you specify <b>standard</b>, you must also enter <b>permit</b> or <b>deny</b> statements using the standard access list syntax.</li> <li>If you specify <b>extended</b>, you must also enter <b>permit</b> or <b>deny</b> statements using the extended access list syntax.</li> <li>The valid values for <i>access-list-name</i> are an alphanumeric string or a number. (Because the name of an access list can be designated as a number, numbers are acceptable as names as long as they are entered in named access list configuration mode.)</li> </ul>
<p><b>Step 4</b></p> <pre>[sequence-number] permit source source-wildcard</pre> <p>or</p> <pre>[sequence-number] permit protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</pre> <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>Although this procedure configures a <b>permit</b> statement first, you could also configure a <b>deny</b> statement first, depending on the order of statements that you need.</li> <li>See the <a href="#">permit (IP), page 56</a> command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP).</li> <li>Enter a value for the (optional) <i>sequence-number</i> attribute to insert the permit statement in a numbered position in the access list.</li> <li>Use the <b>no sequence-number</b> command to delete a sequenced entry from the access list.</li> <li>The prompt indicates whether you are configuring a standard or extended named access list. If you specify <b>extended</b> in Step 3, the prompt for this step changes to <code>Router(config-ext-nacl)</code> to support extended <b>permit</b> command syntax.</li> </ul>

Command or Action	Purpose
<p><b>Step 5</b></p> <pre>[sequence-number] deny source source-wildcard</pre> <p>or</p> <pre>[sequence-number] deny protocol source source-wildcard destination destination-wildcard [precedence precedence] [tos tos] [log] [time-range time-range-name] [fragments]</pre> <p><b>Example:</b>  Router(config-std-nacl)# 105 deny 10.6.6.7 0.0.0 255</p>	<p>(Optional) Specifies a deny statement in named IP access list mode.</p> <ul style="list-style-type: none"> <li>Although this procedure configures a <b>permit</b> statement first, you could also configure a <b>deny</b> statement first, depending on the order of statements that you need.</li> <li>See the <a href="#">deny (IP), page 38</a> command for additional command syntax to permit upper layer protocols (ICMP, IGMP, TCP, and UDP).</li> <li>Enter a value for the (optional) <i>sequence-number</i> attribute to insert the deny statement in a numbered position in the access list.</li> <li>Use the <b>no sequence-number</b> command to delete an entry.</li> <li>The prompt indicates whether you are configuring a standard or extended named access list. If you specify <b>extended</b> in Step 3, the prompt for this step changes to Router(config-ext-nacl) to support extended <b>deny</b> command syntax.</li> </ul>
<p><b>Step 6</b></p> <p>Repeat Step 5 and Step 6 as necessary to insert additional permit and deny statements by sequence number (if necessary). Use the <b>no sequence-number</b> command to delete a sequenced entry.</p>	<p>Allows you to revise the access list.</p>
<p><b>Step 7</b></p> <pre>end</pre> <p><b>Example:</b>  Router(config-std-nacl)# end</p>	<p>(Optional) Exits the configuration mode and returns to privileged EXEC mode.</p>
<p><b>Step 8</b></p> <pre>show ip access-list access-list-name</pre> <p><b>Example:</b>  Router# show ip access-list Internet_filter</p>	<p>(Optional) Displays the contents of the IP access list.</p> <p>Review the output to see that the access list includes the new entry.</p> <pre>Router# show ip access-list Internet_filter  Standard IP access list Internet_filter 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</pre>

## Usage Notes

- When you configure a standard or extended named access list, note that, by default, the end of the access list contains an implicit deny statement that discards all packets that are not matched by preceding **permit** statements.
- When you configure a permit or deny statement with no sequence number, by default it is assigned a sequence number of 10 more than the last entry in the access list.

- When you configure a standard named access list, if you do not specify the mask for an IP host address by entering a value for *source-wildcard* in a permit or deny statement, 0.0.0.0 is used as the default mask.

After you create a standard or extended named ACL, you must apply it to an interface as described in the [“Applying a Named Access Control List to an Interface”](#) section on page 17.

After you configure a sequence of **permit** and **deny** statements in a named ACL, you can maintain the content of the access list as follows:

- To display the current sequenced contents of a named ACL, enter the **show ip access-list *access-list-name*** command.
- To delete a sequenced entry, re-enter the **ip access-list {standard | extended}** command. Then enter the **no *sequence-number* permit** or **no *sequence-number* deny** command to remove a specified access list entry.
- To add a sequenced entry, re-enter the **ip access-list {standard | extended}** command. Then enter a ***sequence-number* permit** or ***sequence-number* deny** command to specify the sequence number and access list entry.
- To apply sequence numbers to unsequenced ACEs or re-order the sequence of entries in a named ACL, follow the procedure described in the [“Resequencing Entries in a Named Access Control List”](#) section on page 16.

## Resequencing Entries in a Named Access Control List

Perform this task to apply sequence numbers to unsequenced access list entries or re-order the sequence of **permit** and **deny** statements in a named ACL.

### Restrictions

- The application of sequence numbers to entries in traditional numbered ACLs configured with the **access-list** command is not supported.  
However, because the name of an access list can be designated as a number, numbers are acceptable as names as long as they are entered in named access list configuration mode. For information, see the [“Configuring Named Access Control Lists”](#) section on page 12.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **ip access-list resequence *access-list-name* *starting-sequence-number* *increment***
4. **end**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> <code>enable</code>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<code>configure terminal</code>  <b>Example:</b> Router# <code>configure terminal</code>	Enters global configuration mode.
Step 3	<code>ip access-list-name resequence</code> <i>starting-sequence-number increment-list extended</i>  <b>Example:</b> Router(config)# <code>ip access-list resequence</code> <code>Internet_filter 100 15</code>	Resequences the specified IP access list using the starting sequence number and the increment of sequence numbers.  This example resequences an access list named <code>Internet_filter</code> . The starting sequence number is 100 and the increment is 15.
Step 4	<code>end</code>  <b>Example:</b> Router(config)# <code>end</code>	Exits the configuration mode and returns to privileged EXEC mode.

To apply a sequenced named ACL to an interface, follow the procedure described in the [“Applying a Named Access Control List to an Interface”](#) section on page 17.

To display the current sequenced contents of a named ACL, enter the `show ip access-list access-list-name` command.

To delete a sequenced entry:

1. Re-enter the `ip access-list {standard | extended}` command.
2. Enter the `no sequence-number permit` or `no sequence-number deny` command to remove a specified access list entry.

To add a sequenced entry:

1. Re-enter the `ip access-list {standard | extended}` command.
2. Enter the `sequence-number permit` or `sequence-number deny` command to insert an access list entry at a specified number in a sequence of access list entries.

## Applying a Named Access Control List to an Interface

After you create a named access list to filter IPv4 traffic on a line card with a distributed switch engine, you must apply the access list to an interface to activate it. You can apply the ACL to one or more interfaces, and on either outbound or inbound interfaces. To configure a named access list, perform the following task.



### Note

For information on the use of any of the ACL commands described in this section, refer to [Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services](#).

## SUMMARY STEPS

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

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> <b>enable</b>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>interface</b> <i>type number</i>  <b>Example:</b> Router# <b>interface pos 3/2</b>	Specifies an interface and enters interface configuration mode.
Step 4	<b>ip access-group</b> <i>access-list-name</i> { <b>in</b>   <b>out</b> }  <b>Example:</b> Router(config-if)# <b>ip access-group Internet_filter out</b>	Controls access to an interface by applying a named ACL on inbound or outbound traffic.
Step 5	<b>exit</b>  <b>Example:</b> Router(config-if)# <b>exit</b>	Exits global configuration mode.

## Usage Notes

- For inbound access lists, after receiving a packet, the Cisco IOS software checks the source address of the packet against the access list. If the access list permits the address, the software continues to process the packet. If the access list rejects the address, the software discards the packet and returns an ICMP host unreachable message.
- For outbound access lists, after receiving and routing a packet to a controlled interface, the software checks the source address of the packet against the access list. If the access list permits the address, the software sends the packet. If the access list rejects the address, the software discards the packet and returns an ICMP host unreachable message.
- When you apply an access list that has not yet been defined to an interface, the software acts as if the access list is not applied to the interface and accepts all packets. If you use undefined access lists as a means of security in your network, remember this behavior.

After you configure a named ACL, you can verify the configuration as described in the “[Verifying Named Access Control Lists](#)” section on page 19.

## Verifying Named Access Control Lists

To display the contents of all current IP named access lists on a router, perform the following task.



### Note

For information on the ACL commands in this section, refer to *Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services*.

### SUMMARY STEPS

1. **enable**
2. **show ip access-list** [*access-list-name*]
3. **exit**

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> <b>enable</b>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<b>show ip access-list</b> [ <i>access-list-name</i> ]  <b>Example:</b> Router# <b>show ip access-list Internetfilter</b>	Displays the contents of all currently configured IP access lists or (optionally) of a specified named access list.
Step 3	<b>exit</b>  <b>Example:</b> Router(config-if)# <b>exit</b>	Exits privileged EXEC mode.

### Usage Notes

- The **show ip access-list** command provides output identical to the **show access-lists** command, except that it is IP-specific and allows you to specify a particular access list.
- You can enter the **show ip access-list** command in either user EXEC or privileged EXEC mode.

## Configuring Time-Based Access Control Entries

This section describes how to configure **permit** and **deny** statements that filter IPv4 traffic for only a specified period of time in an extended numbered or named ACL.

### Benefits

A time-based access control entry (ACE) allows you to turn on and off the capacity to deny and permit IPv4 traffic received from, and sent to, network resources for a certain period of time.

### Restrictions

- ACEs (**permit** and **deny** statements) that specify a time range for when they are active are supported only for IPv4 traffic and only in extended (named or numbered) ACLs on ingress or egress interfaces.
- ACEs that specify a time range are not supported in a reflexive ACL.

### Prerequisites

To configure a time-based ACE, you enter the **time-range** *time-range-name* parameter in a **permit** or **deny** statement. However, before you enter this parameter, you must first configure a time range using the **time-range** command in global configuration mode and define a time period using the **absolute** or **periodic** commands or with a combination of these commands to define when the ACE is in effect.

For example:

```
time-range no-http
  periodic weekdays 8:00 to 18:00

time-range yes-cust010
  absolute start 12:00 1 January 2006
```

For information on this configuration procedure, see the [“Defining a Time Range for Time-Based Access Control Entries”](#) section on page 22.

For information on the **absolute**, **periodic**, and **time-range** command syntaxes, see [absolute](#), page 36, [periodic](#), page 53, and [time-range](#), page 68.

If no name for the time range period has been previously configured using the **time-range** command, the **deny** or **permit** statement which contains the **time-range** *time-range-name* keyword and argument is still active.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list extended** *access-list-number* [**dynamic** *dynamic-name* [*timeout* *minutes*]]  
**{deny | permit} ip** *source source-wildcard destination destination-wildcard* **{time-range**  
*time-range-name* **}** [**precedence** *precedence*] [**tos** *tos*] [**log** | **log-input**] [**t***tl operator ttl-value*]  
**[fragments]**  
 or  
**ip access-list extended** *access-list-name*

```
deny | permit ip source source-wildcard destination destination-wildcard {time-range
time-range-name} [precedence precedence] [tos tos] [log | log-input] [ttl operator ttl-value]
[fragments]
```

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; <b>enable</b></p>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# <b>configure terminal</b></p>	Enters global configuration mode.
Step 3	<pre>access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny   permit} ip source source-wildcard destination destination-wildcard {time-range time-range-name} [precedence precedence] [tos tos] [log   log-input] [ttl operator ttl-value] [fragments]</pre> <p>or</p> <pre>ip access-list extended access-list-name</pre> <pre>{deny   permit} ip source source-wildcard destination destination-wildcard {time-range time-range-name} [precedence precedence] [tos tos] [log   log-input] [ttl operator ttl-value] [fragments]</pre> <p><b>Example:</b> Router# <b>ip access-list extended dept4</b> Router(config-ext-nacl)# <b>permit tcp any any eq</b> <b>telnet time-range yes-telnet</b></p>	<p>Defines an extended numbered access list with a <b>deny</b> or <b>permit</b> statement that specifies a time range when the statement is active.</p> <p>or</p> <p>Defines an extended named access list and enters standard named access list configuration mode to define a <b>deny</b> or <b>permit</b> statement that specifies a time range when the statement is active.</p> <p>The <i>time-range-name</i> value that is specified with the <b>time-range</b> keyword is configured using the <b>time-range</b> command in global configuration mode (see the “<a href="#">Defining a Time Range for Time-Based Access Control Entries</a>” section on page 22).</p> <p>For detailed information on the <b>deny</b> and <b>permit</b> command syntax, see <a href="#">deny (IP)</a>, page 38 and <a href="#">permit (IP)</a>, page 56.</p>

After you create an extended named or numbered ACL that contains one or more time-based **deny** and **permit** statements, you can apply the ACL to an interface by using the **ip access-group** {*access-list-name* | *access-list-number*} {**in** | **out**} command, as described in the “[Applying a Named Access Control List to an Interface](#)” section on page 17.

## Defining a Time Range for Time-Based Access Control Entries

This section describes how to configure the time range specified in a time-based **permit** or **deny** statement in an extended numbered or named ACL.

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **time-range** *time-range-name*
4. **absolute** [*start time date*] [*end time date*]  
and/or  
**periodic** *days-of-the-week hh:mm to [days-of-the-week] hh:mm*

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> <b>enable</b>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# <b>configure terminal</b>	Enters global configuration mode.
Step 3	<b>time-range</b> <i>time-range-name</i>  <b>Example:</b> Router# <b>time-range yes-telnet</b>	Assigns a name to the time-range to be configured and enters time range configuration mode.  For more information on the command syntax, see <a href="#">time-range, page 68</a> .
Step 4	<b>absolute</b> [ <i>start time date</i> ] [ <i>end time date</i> ]  or <b>periodic</b> <i>days-of-the-week hh:mm to [days-of-the-week] hh:mm</i>  <b>Example:</b> Router(config-time-range)# <b>periodic weekdays 8:00 to 18:00</b>	Specifies when the time range will be in effect. Use some combination of these commands. Multiple periodic statements are allowed; only one absolute statement is allowed.  For more information on the command syntax, see <a href="#">absolute, page 36</a> and <a href="#">periodic, page 53</a> .

## Configuring TTL-Based Access Control Entries

This section describes how to configure **permit** and **deny** statements in an extended numbered or named ACL that filter IPv4 traffic based on the TTL value in IPv4 packet headers.

### Benefits

- An access control entry (ACE) that matches traffic according to a specified TTL value enhances a router's security by mitigating many types of DoS attacks and allowing only a restricted range of traffic from BGP neighbors in the receive path.
- You can use TTL-based filtering in combination with existing ACL features, such as:
  - IPv4 ACL filtering
  - IPv4 distributed and aggregate control plane policing (see [Control Plane Policing](#))
  - IPv4 receive ACLs (see [IP Receive ACL](#))

### Restrictions

- Starting in Cisco IOS Release 12.0(33)S, ACEs (**permit** and **deny** statements) that specify a time-to-live (TTL) value are supported only on the distributed switch engines of ISE/Engine 3 and Engine 5 line cards.
- On the Cisco 12000 series Internet router, TTL-based ACEs are supported only for IPv4 traffic and only in extended (named or numbered) ACLs on ingress or egress interfaces.
- ACEs that specify a TTL value are not supported in Turbo ACLs.
- All possible TTL values (0 to 255) are examined in the distributed switching path on an ISE or Engine 5 line card. However, on an ingress interface, incoming IPv4 packets with a TTL value of 0 or 1 are punted to the process level of the route processor before the ingress ACL is checked. IPv4 packets with TTL values of 2 to 255 are first filtered by the ACL applied to the ingress interface; when a packet is matched, a permit or deny decision is made.

Packets with a TTL value of 0 or 1 always get punted to the process level because they are never forwarded out of the router. In the fast-path distributed packet switching performed on an ISE or Engine 5 line card, an ingress ACL configured to deny packets with TTL values of 0 or 1 still does not drop the packets. As a result, the CPU uses more resources, which may result in degraded router performance depending on the number of packets punted to the route processor.

Therefore, to implement TTL-based filtering for IPv4 traffic in a way that fully prevents attacks on the CPU of the router, you must also configure ACEs that filter TTL values 0 and 1 in the ACL used to match IPv4 packets destined for the control plane in aggregate control plane policing.

For an example of how to configure both an ACL that filters TTL values 0 to 255 in incoming IPv4 packets and an ACL that filters only TTL values 0 and 1 in all IPv4 packets punted to the route processor, see the [“Configuring TTL-Based Access Control Entries: Example”](#) section on page 30.

## Configuring a TTL-Based Access Control Entry

Perform this task if you want to create **permit** or **deny** statements in an ACL to match IPv4 packets using a specified TTL value.



### Note

For information on the use of any of the ACL commands described in this section, refer to [Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services](#).

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **access-list extended** *access-list-number* [**dynamic** *dynamic-name* [*timeout* *minutes*]]  
 {**deny** | **permit**} **ip** *source source-wildcard destination destination-wildcard* {**t***tl operator ttl-value*}  
 [**pre***cedence precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]  
 or  
**ip access-list extended** *access-list-name*  
**deny** | **permit ip** *source source-wildcard destination destination-wildcard* {**t***tl operator ttl-value*}  
 [**pre***cedence precedence*] [**tos** *tos*] [**log** | **log-input**] [**time-range** *time-range-name*] [**fragments**]

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<pre>enable</pre> <p><b>Example:</b> Router&gt; <b>enable</b></p>	Enables privileged EXEC mode. If prompted, enter your password.
Step 2	<pre>configure terminal</pre> <p><b>Example:</b> Router# <b>configure terminal</b></p>	Enters global configuration mode.
Step 3	<pre>access-list access-list-number [dynamic dynamic-name [timeout minutes]] {deny   permit} ip source source-wildcard destination destination-wildcard {ttl operator ttl-value} [precedence precedence] [tos tos] [log   log-input] [time-range time-range-name] [fragments]</pre> <p>or</p> <pre>ip access-list extended access-list-name</pre> <pre>{deny   permit} ip source source-wildcard destination destination-wildcard {ttl operator ttl-value} [precedence precedence] [tos tos] [log   log-input] [time-range time-range-name] [fragments]</pre> <p><b>Example:</b> Router# <b>ip access-list extended input-acl</b> Router(config-ext-nacl)# deny ip any any tos 3 ttl eq 10 Router(config-ext-nacl)# deny ip any any tos 3 ttl eq 20 Router(config-ext-nacl)# deny ip any any ttl gt 154 fragments Router(config-ext-nacl)# permit ip any any precedence flash ttl neq 1 log </p>	<p>Defines an extended numbered access list with a <b>deny</b> or <b>permit</b> statement that specifies a TTL value for IPv4 packet filtering.</p> <p>or</p> <p>Defines an extended named access list and enters standard named access list configuration mode to define a <b>deny</b> or <b>permit</b> statement that specifies a TTL value for IPv4 packet filtering.</p> <p>The <b>ttl operator ttl-value</b> keyword and arguments compares the TTL value in an IPv4 packet to the TTL value specified in the <b>deny</b> or <b>permit</b> statement.</p> <ul style="list-style-type: none"> <li>• The <i>operator</i> can be <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), or <b>range</b> (inclusive range).</li> <li>• The <i>value</i> can range from 0 to 255.</li> <li>• If the operator is <b>range</b>, specify two values separated by a space.</li> <li>• For Release 12.0S, if the operator is <b>eq</b> or <b>neq</b>, only one TTL value can be specified.</li> </ul> <p>For all other releases, if the operator is <b>eq</b> or <b>neq</b>, as many as 10 TTL values can be specified, separated by a space.</p> <p>For detailed information on the <b>deny</b> and <b>permit</b> command syntax, see <a href="#">deny (IP)</a>, page 38 and <a href="#">permit (IP)</a>, page 56.</p>

After you create an extended named or numbered ACL that contains one or more TTL-based **deny** and **permit** statements, you can apply the ACL to an interface by using the **ip access-group** {*access-list-name* | *access-list-number*} {**in** | **out**} command, as described in the “[Applying a Named Access Control List to an Interface](#)” section on page 17.

If you use TTL-based **deny** or **permit** statements in an ACL applied to an ingress interface, you must perform an additional step to fully prevent attacks on the CPU of the router from incoming IPv4 traffic. You must also configure ACEs that filter TTL values 0 and 1 in the ACL used for aggregate control plane policing. For more information on the additional configuration required for packet filtering of the TTL values 0 and 1, see the [“Using Time-to-Live Access List Entries” section on page 10](#).

For an example of how to configure both an ACL that filters TTL values 0 to 255 in IPv4 packets on an ingress interface and an ACL that filters TTL values 0 and 1 in all incoming IPv4 packets punted to the route processor, see the [“Configuring TTL-Based Access Control Entries: Example” section on page 30](#).

## Verifying Time-Based and TTL-Based Access Control Entries

To verify the configuration of a **permit** or **deny** statement that matches IPv4 packets using a Time-to-Live value (TTL-based ACE) or that is configured to be active for a specified time period (time-based ACE), perform the following task.

**Note**

---

For information on ACL commands in this section, refer to [Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services](#).

---

### SUMMARY STEPS

1. **enable**
2. **show access-list** [*access-list-number*]  
Or  
**show ip access-list extended** *access-list-name*
3. **exit**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>	Enables privileged EXEC mode. If prompted, enter your password.
	<b>Example:</b> Router> <code>enable</code>	
Step 2	<code>show access-list [access-list-number]</code>  or <code>show ip access-list [access-list-name]</code>	Displays the contents of all currently configured IP access lists or (optionally) of a specified numbered access list.  or Displays the contents of all currently configured IP access lists or (optionally) of a specified named access list.
	<b>Example:</b> Router# <code>show ip access-list input-acl</code>	
Step 3	<code>exit</code>	Exits privileged EXEC mode.
	<b>Example:</b> Router(config-if)# <code>exit</code>	

## Usage Notes

- The `show ip access-list` command provides output identical to the `show access-lists` command, except that it is IP-specific and allows you to specify a particular access list.
- You can enter the `show ip access-list` command in either user EXEC or privileged EXEC mode.

## Configuration Examples for ACL Enhancements on the Cisco 12000 Series Router

This section provides the following configuration examples:

- [Configuring Named Access Control Lists: Example, page 27](#)
- [Verifying Named Access Control Lists: Example, page 28](#)
- [Configuring TTL-Based Access Control Entries: Example, page 30](#)
- [Configuring Named Access Control Lists: Example, page 27](#)
- [Verifying TTL-Based and Time-Based Access Control Entries: Example, page 32](#)

### Configuring Named Access Control Lists: Example

The following example shows how to configure a standard access list named `Internet_filter` and an extended access list named `marketing_group` and apply them to an engine 5 SPA interface (specified in the format `slot/subslot/port`):

```
ip access-list standard Internet_filter
    permit 1.2.3.4
    deny any
ip access-list extended marketing_group
```

```

permit tcp any 171.69.0.0 0.0.255.255 eq telnet
deny tcp any any
permit icmp any any
deny udp any 171.69.0.0 0.0.255.255 lt 1024
deny ip any any log
...
interface gigabitethernet0/2/1
 ip address 2.0.5.1 255.255.255.0
 ip access-group Internet_filter out
 ip access-group marketing_group in

```

## Verifying Named Access Control Lists: Example

This section contains examples that show how to verify a named access list called `Internet_filter` and display all configured IP access lists (standard and extended; named and numbered) using the **show ip access-list** command.

The following is sample output from the **show ip access-list** command when the name of a specific access list is requested:

```

show ip access-list Internetfilter

Extended IP access list Internetfilter
 permit tcp any 171.69.0.0 0.0.255.255 eq telnet
 deny tcp any any
 deny udp any 171.69.0.0 0.0.255.255 lt 1024
 deny ip any any log

```

The following is sample output from the **show ip access-list** command when all access lists are requested:

```

show ip 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

Extended IP access list 101
 deny udp any any eq ntp
 permit tcp any any
 permit udp any any eq tftp
 permit icmp any any
 permit udp any any eq domain

```

## Resequencing Entries in an Access List: Example

The following example shows how to resequence access list entries. 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 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
```

```
Router(config)# ip access-list extended 150
Router(config)# ip access-list resequence 150 1 2
Router(config)# end
```

```
Router# 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
```

## Adding Entries with Sequence Numbers: Example

In the following example, a new entry is added to a specified 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
```

## Entry without Sequence Number: Example

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 has a sequence number 10 higher than the last entry in the existing access list.

```
Router(config)# ip access-list standard 1
```

```

Router(config-std-nacl)# permit 1.1.1.1 0.0.0.255
Router(config-std-nacl)# permit 2.2.2.2 0.0.0.255
Router(config-std-nacl)# permit 3.3.3.3 0.0.0.255

Router# 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

Router(config)# ip access-list standard 1
Router(config-std-nacl)# permit 4.4.4.4 0.0.0.255
Router(config-std-nacl)# end

Router# 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 0.4.0.0, wildcard bits 0.0.0.255

```

## Configuring TTL-Based Access Control Entries: Example

The following example shows how to configure an extended named ACL with various ACEs that specify Time-to-Live values for IPv4 packet filtering on an ingress interface.

```

ip access-list extended input-acl
 permit ip host 171.69.1.129 any ttl lt 2
 permit ip any any ttl range 100 120
 deny ip host 10.1.1.1 any ttl gt 253
 ...
interface gigabitethernet0/2/1
 ip address 2.0.5.1 255.255.255.0
 ip access-group input-acl in

```

If you use TTL-based **deny** or **permit** statements in an ACL applied to an ingress interface, you must perform an additional step to fully prevent attacks on the CPU of the router from incoming IPv4 traffic. You must also configure ACEs that filter TTL values 0 and 1 in the ACL used for aggregate control plane policing. For more information on this behavior, see the [“ACL Filtering of TTL Values 0 and 1 on Ingress Interfaces” section on page 11](#).

The next example shows how to configure both:

- An ACL (cp-filter) that filters IPv4 packets so that packets with TTL values 0 and 1 are denied
- A quality-of-service (QoS) policy used to police all packets punted to the route processor and that uses the ACL with TTL filtering to match IPv4 packets with TTL values 0 and 1 so that they are dropped before reaching the control plane.

The QoS policy used to police control plane traffic received from all interfaces on all line cards on a router is also known as *aggregate control plane policing*. One aggregate QoS policy used for control plane policing is:

- Defined for the input control-plane interface and represents an aggregate for all ports on a router.
- Defined using the modular QoS command-line interface (MQC), including class maps and policy maps for both DoS protection and packet QoS.

For information about how to configure and use the Control Plane Policing feature, refer to [Control Plane Policing](#).

```
ip access-list extended cp-filter
  permit ip host 171.69.1.129 any ttl lt 10
  permit ip any any ttl range 100 120
  deny ip any any ttl eq 0
  deny ip any any ttl eq 1
  exit
class-map cp-filter-class
  match access-group cp-filter
  exit
policy-map cp-ttl-in
  class cp-filter-class
    police 80000 conform transmit exceed drop
  exit
  exit
control-plane slot 1
  service-policy input cp-ttl-in
  exit
```

## Configuring Time-Based Access Control Entries: Example

This section contains several examples of how to configure time-based **permit** and **deny** statements in extended named ACLs, including the required time-range configuration.

The following example shows how to configure a time-based ACE that denies HTTP traffic on a router from Monday through Friday between the hours of 8:00 AM and 6:00 PM.

```
time-range no-http
  periodic weekdays 8:00 to 18:00

ip access-list extended foo
  deny tcp any any eq http time-range no-http
```

The next example shows how to configure a time-based ACE that permits cust-gold traffic starting on a specific time and date and going forever since there is no ending time specification:

```
time-range yes-cust-gold
  absolute start 12:00 1 January 2001

ip access-list extended cust-gold
  permit foo any any time-range yes-cust-gold
```

The following example shows how to configure a time-based ACE that permits UDP traffic until 12:00 noon on December 31, 2006. After this date and time, UDP traffic is no longer allowed because the ACE is inactive.

```
time-range stop-udp
  absolute end 12:00 31 December 2006

ip access-list extended udp
  permit udp any any time-range stop-udp
```

This example shows how to configure a time-based ACE that permits Telnet traffic on Mondays, Tuesdays, and Fridays between 9:00 AM and 5:00 PM.

```
time-range yes-telnet
  periodic Monday Tuesday Friday 9:00 to 17:00

ip access-list extended foo
  permit tcp any any eq telnet time-range yes-telnet
```

The following example shows how to configure a time-based ACE that permits UDP traffic on weekends from 8:00 AM on January 1, 2006 to 18:00 on December 31, 2006.

```
time-range maybe-udp
  absolute start 8:00 1 January 1999 end 18:00 31 December 2001
  periodic weekends 00:00 to 23:59

ip access-list extended foo
  permit udp any any time-range maybe-udp
```

## Verifying TTL-Based and Time-Based Access Control Entries: Example

The following example shows how to verify the configuration of TTL-based and time-based ACEs in ACLs configured on a router:

```
show ip access-list

Extended IP access list 150
  deny ip any any tos 3 ttl eq 10
  deny ip any any tos 3 ttl eq 20
  deny ip any any ttl gt 154 fragments
  permit ip any any precedence flash ttl neq 1 log
Extended IP access list input-acl
  permit ip host 171.69.1.129 any ttl lt 2
  permit ip any any ttl range 100 120
  deny ip host 10.1.1.1 any ttl gt 253
Extended IP access list http-weekdays
  deny tcp any any eq http time-range no-http ([inactive])
```

## Additional References

The following sections provide references related to ACL enhancements on the Cisco 12000 series router.

## Related Documents

Related Topic	Document Title
Introduction and guidelines on using access control lists as part of a security solution	<i>Cisco IOS Security Configuration Guide, Release 12.2 Access Control Lists: Overview and Guidelines</i>
Configure access control lists to filter IP traffic	<i>Cisco IOS IP Configuration Guide, Release 12.3 Part 1: IP Addressing and Services Configuring IP Services</i> (“Filtering IP Packets Using Access Lists”)
IP ACL command syntax	<i>Cisco IOS IP Command Reference, Volume 1 of 4: Addressing and Services, Release 12.3</i>
Configure a quality-of-service filter that manages the traffic flow of control plane packets and protects the control plane of Cisco IOS routers	<i>Control Plane Policing</i>
Configure an IP receive ACL that filters traffic on distributed line cards before packets are received by the route processor and filters DoS attacks	<i>IP Receive ACL</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 releases, and feature sets, use Cisco MIB Locator at: <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 was not modified by this feature.	—

## Technical Assistance

Description	Link
The Cisco Technical Support & Documentation website contains thousands of pages of searchable technical content, including links to products, technologies, solutions, technical tips, and tools. Registered Cisco.com users can log in from this page to access more content.	<a href="http://www.cisco.com/techsupport">http://www.cisco.com/techsupport</a>

# Command Reference

This section documents only the following modified commands:

- [absolute](#), page 36
- [deny \(IP\)](#), page 38
- [ip access-list](#), page 49
- [ip access-list resequence](#), page 51
- [periodic](#), page 53
- [permit \(IP\)](#), page 56
- [time-range](#), page 68

# absolute

To specify an absolute time for a time-range, use the **absolute** command in time-range configuration mode. To remove the time limitation, use the **no** form of this command.

**absolute** [*start time date*] [*end time date*]

**no absolute**

## Syntax Description

<b>start time date</b>	(Optional) Absolute time and date that the <b>permit</b> or <b>deny</b> statement of the associated access list starts going into effect. The <i>time</i> is expressed in 24-hour notation, in the form of <i>hours:minutes</i> . For example, 8:00 is 8:00 AM and 20:00 is 8:00 PM. The <i>date</i> is expressed in the format <i>day month year</i> . The minimum start is 00:00 1 January 1993. If no start time and date are specified, the <b>permit</b> or <b>deny</b> statement is in effect immediately.
<b>end time date</b>	(Optional) Absolute time and date that the <b>permit</b> or <b>deny</b> statement of the associated access list is no longer in effect. Same <i>time</i> and <i>date</i> format as described for the <b>start</b> keyword. The end time and date must be after the start time and date. The maximum end time is 23:59 31 December 2035. If no end time and date are specified, the associated <b>permit</b> or <b>deny</b> statement is in effect indefinitely.

## Defaults

There is no absolute time when the time range is in effect.

## Command Modes

Time-range configuration

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.0(33)S	Support for this command was introduced on the Cisco 12000 series Internet router.

## Usage Guidelines

Time ranges are used by IP and Internetwork Packet Exchange (IPX) extended access lists. For more information on using these functions, see the *Cisco IOS IP Configuration Guide* and the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*. Time ranges are applied to the **permit** or **deny** statements found in these access lists.

The **absolute** command is one way to specify when a time range is in effect. Another way is to specify a periodic length of time with the **periodic** command. Use either of these commands after the **time-range** command, which enables time-range configuration mode and specifies a name for the time range. Only one **absolute** entry is allowed per **time-range** command.

If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** items are evaluated only after the **absolute start** time is reached, and are not further evaluated after the **absolute end** time is reached.

**Note**

All time specifications are interpreted as local time. To ensure that the time range entries take effect at the desired times, the software clock should be synchronized using the Network Time Protocol (NTP), or some other authoritative time source. For more information, refer to the “Performing Basic System Management” document on Cisco.com.

**Examples**

In the following example, an access list named ‘northeast’ references a time range named ‘xyz’. The access list and time range configuration permits traffic on Ethernet interface 0, starting at 12:00 noon on January 1, 2005 and going forever.

```
time-range xyz
  absolute start 12:00 1 January 2005
!
ip access-list extended northeast
  permit ip any any time-range xyz
!
interface ethernet 0
  ip access-group northeast in
```

The configuration sample permits UDP traffic until 12:00 noon on December 31, 2005. After that time, UDP traffic is no longer allowed out Ethernet interface 0.

```
time-range abc
  absolute end 12:00 31 December 2005
!
ip access-list extended northeast
  permit udp any any time-range abc
!
interface ethernet 0
  ip access-group northeast out
```

The configuration sample permits outgoing UDP traffic on Ethernet interface 0 on weekends only, from 8:00 AM on January 1, 2005, to 6:00 PM on December 31, 2006:

```
time-range weekend1
  absolute start 8:00 1 January 2005 end 18:00 31 December 2006
  periodic weekends 00:00 to 23:59
!
ip access-list extended northeast1
  permit udp any any time-range weekend1
!
interface ethernet 0
  ip access-group northeast1 out
```

**Related Commands**

Command	Description
<b>deny</b>	Sets conditions under which a packet does not pass a named access list.
<b>periodic</b>	Specifies a recurring (weekly) start and end time for a time range.
<b>permit</b>	Sets conditions under which a packet passes a named access list.
<b>time-range</b>	Enables time-range configuration mode and names a time range definition.

## deny (IP)

To set conditions in a named IP access list that will deny packets, use the **deny** command in access list configuration mode. To remove a deny condition from an access list, use the **no** form of this command.

```
[sequence-number] deny source [source-wildcard]
```

```
[sequence-number] deny protocol source source-wildcard destination destination-wildcard
[option option-name] [precedence precedence] [tos tos] [ttl operator value] [log]
[time-range time-range-name] [fragments]
```

```
no sequence-number
```

```
no deny source [source-wildcard]
```

```
no deny protocol source source-wildcard destination destination-wildcard
```

### Internet Control Message Protocol—ICMP

```
[sequence-number] deny icmp source source-wildcard destination destination-wildcard [icmp-type
[icmp-code] | icmp-message] [precedence precedence] [tos tos] [ttl operator value] [log]
[time-range time-range-name] [fragments]
```

### Internet Group Management Protocol—IGMP

```
[sequence-number] deny igmp source source-wildcard destination destination-wildcard
[igmp-type] [precedence precedence] [tos tos] [ttl operator value] [log] [time-range
time-range-name] [fragments]
```

### Transmission Control Protocol—TCP

```
[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] [ttl operator value] [log]
[time-range time-range-name] [fragments]
```

### User Datagram Protocol—UDP

```
[sequence-number] deny udp source source-wildcard [operator port [port]] destination
destination-wildcard [operator [port]] [precedence precedence] [tos tos] [ttl operator value]
[log] [time-range time-range-name] [fragments]
```

Syntax	Description
<i>sequence-number</i>	(Optional) Sequence number assigned to the deny statement. The sequence number causes the system to insert the statement in that numbered position in the access list.
<i>source</i>	Number of the network or host from which the packet is being sent. Three alternative ways to specify the source exist: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format.</li> <li>Use the <b>any</b> keyword as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host source</b> as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.</li> </ul>
<i>source-wildcard</i>	Wildcard bits to be applied to the source. Three alternative ways to specify the source wildcard exist: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore.</li> <li>Use the <b>any</b> keyword as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host source</b> as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.</li> </ul>
<i>protocol</i>	Name or number of an Internet protocol. The <i>protocol</i> argument can be one of the keywords <b>eigrp</b> , <b>gre</b> , <b>icmp</b> , <b>igmp</b> , <b>ip</b> , <b>ipinip</b> , <b>nos</b> , <b>ospf</b> , <b>tcp</b> , or <b>udp</b> , or an integer in the range from 0 to 255 representing an Internet protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the <b>ip</b> keyword. <p><b>Note</b> When the <b>icmp</b>, <b>igmp</b>, <b>tcp</b>, and <b>udp</b> keywords are entered, they must be followed with the specific command syntax that is shown for the ICMP, IGMP, TCP, and UDP forms of the <b>deny</b> command.</p>
<b>icmp</b>	Denies only ICMP packets. When you enter the <b>icmp</b> keyword, you must use the specific command syntax shown for the ICMP form of the <b>deny</b> command.
<b>igmp</b>	Denies only IGMP packets. When you enter the <b>igmp</b> keyword, you must use the specific command syntax shown for the IGMP form of the <b>deny</b> command.
<b>tcp</b>	Denies only TCP packets. When you enter the <b>tcp</b> keyword, you must use the specific command syntax shown for the TCP form of the <b>deny</b> command.
<b>udp</b>	Denies only UDP packets. When you enter the <b>udp</b> keyword, you must use the specific command syntax shown for the UDP form of the <b>deny</b> command.
<i>destination</i>	Number of the network or host to which the packet is being sent. Three alternative ways to specify the destination exist: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format.</li> <li>Use the <b>any</b> keyword as an abbreviation for the <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host destination</b> as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.</li> </ul>

<i>destination-wildcard</i>	<p>Wildcard bits to be applied to the destination. Three alternative ways to specify the destination wildcard exist:</p> <ul style="list-style-type: none"> <li>• Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore.</li> <li>• Use the <b>any</b> keyword as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>• Use <b>host destination</b> as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.</li> </ul>
<b>option</b> <i>option-name</i>	(Optional) Packets can be filtered by IP Options, as specified by a number from 0 to 255 or by the corresponding IP Option name, as listed in <a href="#">Table 1</a> in the “Usage Guidelines” section.
<b>precedence</b> <i>precedence</i>	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by a name.
<b>tos</b> <i>tos</i>	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by a name as listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.
<b>ttl</b> <i>operator value</i>	<p>(Optional) Compares the TTL value in the packet to the TTL value specified in this <b>deny</b> statement.</p> <ul style="list-style-type: none"> <li>• The <i>operator</i> can be <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), or <b>range</b> (inclusive range).</li> <li>• The <i>value</i> can range from 0 to 255.</li> <li>• If the operator is <b>range</b>, specify two values separated by a space.</li> <li>• For Release 12.0S, if the operator is <b>eq</b> or <b>neq</b>, only one TTL value can be specified.</li> <li>• For all other releases, if the operator is <b>eq</b> or <b>neq</b>, as many as 10 TTL values can be specified, separated by a space. If the TTL in the packet matches just one of the possibly 10 values, the entry is considered to be matched.</li> </ul>
<b>log</b>	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the <b>logging console</b> command.)
<b>time-range</b> <i>time-range-name</i>	(Optional) Name of the time range that applies to this <b>deny</b> statement. The name of the time range and its restrictions are specified by the <b>time-range</b> and <b>absolute</b> or <b>periodic</b> commands, respectively.
<b>fragments</b>	(Optional) The access list entry applies to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the <b>fragments</b> keyword, see the “ <a href="#">Access List Processing of Fragments</a> ” and “ <a href="#">Fragments and Policy Routing</a> ” sections in the “Usage Guidelines” section.
<i>icmp-type</i>	(Optional) ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
<i>icmp-code</i>	(Optional) ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255.

<i>icmp-message</i>	(Optional) ICMP packets can be filtered by an ICMP message type name or an ICMP message type and code name. The possible names are listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.
<i>igmp-type</i>	(Optional) IGMP packets can be filtered by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.
<i>operator</i>	(Optional) Compares source or destination ports. 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).  If the operator is positioned after the <i>source</i> and <i>source-wildcard</i> arguments, it must match the source port. If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i> arguments, it must match the destination port.  The <b>range</b> operator requires two port numbers. Up to ten port numbers can be entered for the <b>eq</b> (equal) and <b>neq</b> (not equal) operators. All other operators require one port number.
<i>port</i>	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.  TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
<b>established</b>	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bit set. The nonmatching case is that of the initial TCP datagram to form a connection.  <b>Note</b> The <b>established</b> keyword can be used only with the old command-line interface (CLI) format. To use the new CLI format, you must use the <b>match-any</b> or <b>match-all</b> keywords followed by the + or - keywords and <i>flag-name</i> argument.
<b>{match-any   match-all}</b>	(Optional) For the TCP protocol only: A match occurs if the TCP datagram has certain TCP flags set or not set. You use the <b>match-any</b> keyword to allow a match to occur if any of the specified TCP flags are present, or you can use the <b>match-all</b> keyword to allow a match to occur only if all of the specified TCP flags are present. You must follow the <b>match-any</b> and <b>match-all</b> keywords with the + or - keyword and the <i>flag-name</i> argument to match on one or more TCP flags.
<b>{+   -} flag-name</b>	(Optional) For the TCP protocol only: The + keyword allows IP packets if their TCP headers contain the TCP flags that are specified by the <i>flag-name</i> argument. The - keyword filters out IP packets that do not contain the TCP flags specified by the <i>flag-name</i> argument. You must follow the + and - keywords with the <i>flag-name</i> argument. TCP flag names can be used only when filtering TCP. Flag names for the TCP flags are as follows: <b>urg</b> , <b>ack</b> , <b>psh</b> , <b>rst</b> , <b>syn</b> , and <b>fin</b> .

**Defaults**

No specific conditions under which a packet is denied passing the named access list exist.

**Command Modes** Access list configuration

Command History	Release	Modification
	11.2	This command was introduced.
	12.0(1)T	The <b>time-range</b> <i>time-range-name</i> keyword and argument were added.
	12.0(11)	The <b>fragments</b> keyword was added.
	12.2(13)T	The <b>igrp</b> keyword was removed because the IGRP protocol is no longer available in Cisco IOS software.
	12.2(14)S	The <i>sequence-number</i> argument was added.
	12.2(15)T	The <i>sequence-number</i> argument was added.
	12.3(4)T	The <b>option</b> <i>option-name</i> keyword and argument were added. The <b>match-any</b> , <b>match-all</b> , <b>+</b> , and <b>-</b> keywords and the <i>flag-name</i> argument were added.
	12.3(7)T	Command functionality was modified to allow up to ten port numbers to be added after the <b>eq</b> and <b>neq</b> operators so that an access list entry can be created with noncontiguous ports.
	12.4(2)T	The <b>ttl operator value</b> keyword and arguments were added.
	12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.0(33)S	Support for the <b>ttl operator value</b> keyword and arguments and the <b>time-range</b> <i>time-range-name</i> keyword and argument were introduced on the Cisco 12000 series router.

**Usage Guidelines** Use this command following the **ip access-list** command to specify conditions under which a packet cannot pass the named access list.

#### log Keyword

A log message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Use the **ip access-list log-update** command to generate logging messages when the number of matches reaches a configurable threshold (rather than waiting for a 5-minute-interval). For more information, see the **ip access-list log-update** command.

The logging facility might drop some logging message packets if too many to be managed exist, or if there is more than one logging message to be managed in 1 second. This behavior prevents the router from crashing because of 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.

If you enable Cisco Express Forwarding (CEF) and then create an access list that uses the **log** keyword, the packets that match the access list are not CEF-switched. They are fast-switched. Logging disables CEF.

**time-range Keyword**

The **time-range** keyword allows you to identify a time range by name to specify when a **deny** statement is enabled. Before you enter this parameter, you must first configure a time range using the **time-range** command in global configuration mode and define a time period with the **absolute** and **periodic** commands; for example:

```
time-range no-http
  periodic weekdays 8:00 to 18:00

time-range yes-cust010
  absolute start 12:00 1 January 2006
```

The **time-range** configuration in a **deny** statement is not supported in the following situations:

- When used in a reflexive ACL
- When used in a merged ACL
- On an interface on which policy-based routing (PBR) is already configured.

To use a time-based **deny** statement in a merged ACL, you must first disable merging on the Cisco 12000 series line card. To disable ACL merging, follow these steps:

1. Enter the **hw-module slot slot-number tcam compile acl no-merge** command in global configuration mode.
2. Re-apply the ACL with time-based ACEs on the appropriate interface by entering the **ip access-group access-list-name {in | out}** command in interface configuration mode.

In general, the exact time when a time-based **deny** statement is enabled and disabled varies from the configured time depending on CPU utilization, number and length of ACLs being processed, and number of TCAM entries that correspond to an ACE. Take these factors into account when you configure time ranges for a **deny** statement.

**Access List Filtering of IP Options**

Access control lists can be used to filter packets with IP Options to prevent routers from being saturated with spurious packets containing IP Options. To see a complete table of all IP Options, including ones currently not in use, refer to the latest Internet Assigned Numbers Authority (IANA) information that is available from its URL: [www.iana.org](http://www.iana.org).

Cisco IOS software allows you to filter packets according to whether they contain one or more of the legitimate IP Options by entering either the IP Option value or the corresponding name for the *option-name* argument as shown in [Table 1](#).

**Table 1** IP Option Values and Names

IP Option Value or Name	Description
0 to 255	IP Options values.
add-ext	Match packets with Address Extension Option (147).
any-options	Match packets with any IP Option.
com-security	Match packets with Commercial Security Option (134).
dps	Match packets with Dynamic Packet State Option (151).
encode	Match packets with Encode Option (15).
eool	Match packets with End of Options (0).
ext-ip	Match packets with Extended IP Options (145).

**Table 1** IP Option Values and Names (continued)

IP Option Value or Name	Description
ext-security	Match packets with Extended Security Option (133).
finn	Match packets with Experimental Flow Control Option (205).
imitd	Match packets with IMI Traffic Descriptor Option (144).
lsr	Match packets with Loose Source Route Option (131).
mtup	Match packets with MTU Probe Option (11).
mtur	Match packets with MTU Reply Option (12).
no-op	Match packets with No Operation Option (1).
nsapa	Match packets with NSAP Addresses Option (150).
record-route	Match packets with Router Record Route Option (7).
router-alert	Match packets with Router Alert Option (148).
sdb	Match packets with Selective Directed Broadcast Option (149).
security	Match packets with Base Security Option (130).
ssr	Match packets with Strict Source Routing Option (137).
stream-id	Match packets with Stream ID Option (136).
timestamp	Match packets with Time Stamp Option (68).
traceroute	Match packets with Trace Route Option (82).
ump	Match packets with Upstream Multicast Packet Option (152).
visa	Match packets with Experimental Access Control Option (142).
zsu	Match packets with Experimental Measurement Option (10).

### Filtering IP Packets Based on TCP Flags

The access list entries that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have very specific groups of TCP flags set or not set. Users can select any desired combination of TCP flags with which to filter TCP packets. Users can configure access list entries in order to allow matching on a flag that is set and on a flag that is not set. Use the **+** and **-** keywords with a flag name to specify that a match is made based on whether a TCP header flag has been set. Use the **match-any** and **match-all** keywords to allow the packet if any or all, respectively, of the flags specified by the **+** or **-** keyword and *flag-name* argument have been set or not set.

### Access List Processing of Fragments

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

If the Access-List Entry Has...	Then...
No <b>fragments</b> keyword (the default behavior), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information: <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.</li> </ul> For an access list entry that contains Layer 3 and Layer 4 information: <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 managed 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,	The access list entry is applied only to noninitial fragments. The <b>fragments</b> keyword cannot be configured for an access list entry that contains any Layer 4 information.

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 includes the **fragments** keyword and applies to the subsequent fragments. In the cases in which multiple **deny** access list entries exist for the same host but with different Layer 4 ports, one **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.

**Note**

The **fragments** keyword cannot solve all cases that involve access lists and IP fragments. For TCP flow and when FO=1, the packets are not filtered even if the Layer 3 information matches and entry has deny statement.

### Fragments and Policy Routing

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list has entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy-routed, even if the first fragment is not policy-routed.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made, and it is more likely that policy routing occurs as intended.

### Creating an Access List Entry with Noncontiguous Ports

For Cisco IOS Release 12.3(7)T and later releases, you can specify noncontiguous ports on the same access control entry, which greatly reduces the number of access list entries required for the same source address, destination address, and protocol. If you maintain large numbers of access list entries, we recommend that you consolidate them when possible by using noncontiguous ports. You can specify up to ten port numbers following the **eq** and **neq** operators.

## Examples

The following example sets conditions for a standard access list named Internet filter:

```
ip access-list standard Internetfilter
  deny 192.168.34.0 0.0.0.255
  permit 172.16.0.0 0.0.255.255
  permit 10.0.0.0 0.255.255.255
! (Note: all other access implicitly denied.)
```

The following example denies HTTP traffic on Monday through Friday from 8:00 AM to 6:00 PM:

```
time-range no-http
  periodic weekdays 8:00 to 18:00
!
ip access-list extended strict
  deny tcp any any eq http time-range no-http
!
interface ethernet 0
  ip access-group strict in
```

The following example adds an entry with the sequence number 25 to extended IP access list 150:

```
ip access-list extended 150
  25 deny ip host 172.16.3.3 host 192.168.5.34
```

The following example removes the entry with the sequence number 25 from the extended access list example shown above:

```
no 25
```

The following example sets a deny condition for an extended access list named filter2. The access list entry specifies that a packet cannot pass the named access list if it contains the Strict Source Routing IP Option, which is represented by the IP option value **ssr**.

```
ip access-list extended filter2
  deny ip any any option ssr
```

The following example sets a deny condition for an extended access list named kmdfilter1. The access list entry specifies that a packet cannot pass the named access list if the RST and FIN TCP flags have been set for that packet:

```
ip access-list extended kmdfilter1
  deny tcp any any match-any +rst +fin
```

The following example shows several **deny** statements that can be consolidated into one access list entry with noncontiguous ports. The **show access-lists** command is entered to display a group of access list entries for the access list named abc.

```
Router# show access-lists abc

Extended IP access list abc
 10 deny tcp any eq telnet any eq 450
 20 deny tcp any eq telnet any eq 679
 30 deny tcp any eq ftp any eq 450
 40 deny tcp any eq ftp any eq 679
```

Because the entries are all for the same **deny** 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
 deny tcp any eq telnet ftp any eq 450 679
```

The following examples shows the creation of the consolidated access list entry:

```
Router# show access-lists abc

Extended IP access list abc
 10 deny tcp any eq telnet ftp any eq 450 679
```

The following access list filters IP packets containing Type of Service (ToS) level 3 with 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 not equal to 1, and sends log messages about such packets to the console. All other packets are denied.

```
ip access-list extended canton
 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
```

## Related Commands

Command	Description
<b>absolute</b>	Specifies an absolute time when a time range is in effect.
<b>access-list (IP extended)</b>	Defines an extended IP access list.
<b>access-list (IP standard)</b>	Defines a standard IP access list.
<b>ip access-group</b>	Controls access to an interface.
<b>ip access-list</b>	Defines an IP access list by name.
<b>ip access-list log-update</b>	Sets the threshold number of packets that cause a logging message.
<b>ip access-list resequence</b>	Applies sequence numbers to the access list entries in an access list.
<b>ip options</b>	Drops or ignores IP Options packets that are sent to the router.
<b>logging console</b>	Sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.

<b>Command</b>	<b>Description</b>
<b>match ip address</b>	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
<b>periodic</b>	Specifies a recurring (weekly) time range for functions that support the time-range feature.
<b>permit (IP)</b>	Sets conditions under which a packet passes a named IP access list.
<b>remark</b>	Writes a helpful comment (remark) for an entry in a named IP access list.
<b>show access-lists</b>	Displays a group of access-list entries.
<b>show ip access-list</b>	Displays the contents of all current IP access lists.
<b>time-range</b>	Specifies when an access list or other feature is in effect.

# ip access-list

To define an IP access list by name and enter named access list configuration mode, use the **ip access-list** command in global configuration mode. To remove a named IP access list, use the **no** form of this command.

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

```
no ip access-list {standard | extended} access-list-name
```

## Syntax Description

<b>standard</b>	Specifies a standard IP access list.
<b>extended</b>	Specifies an extended IP access list.
<i>access-list-name</i>	Name of the access list. Names cannot contain a space or quotation mark, and <b>must begin with an alphabetic character to prevent ambiguity with numbered access lists</b> .  Because the name of an access list can be designated as a number, numbers are acceptable as names as long as they are entered in named access list configuration mode.

## Defaults

No named IP access list is defined.

## Command Modes

Global configuration

## Command History

Release	Modification
11.2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

## Usage Guidelines

Use this command to configure a named IP access list as opposed to a numbered IP access list. This command will place the router in access-list configuration mode, where you must define the denied or permitted access conditions with the **deny** and **permit** commands.

Specifying the **standard** or **extended** keyword with the **ip access-list** command determines the prompt you get when you enter access-list configuration mode.

Use the **ip access-group** command to apply the access list to an interface.

Named access lists are not compatible with Cisco IOS releases earlier than Release 11.2.

## Examples

The following example defines a standard access list named Internet filter:

```
ip access-list standard Internetfilter
 permit 192.5.34.0 0.0.0.255
 permit 10.88.0.0 0.0.255.255
 permit 10.0.0.0 0.255.255.255
! (Note: all other access implicitly denied)
```

<b>Related Commands</b>	<b>Command</b>	<b>Description</b>
	<b>access list (IP extended)</b>	Defines an extended IP access list.
	<b>access list (IP standard)</b>	Defines a standard IP access list.
	<b>access-list remark</b>	Writes a helpful comment (remark) for an entry in a numbered access list.
	<b>deny (IP)</b>	Sets conditions for a named IP access list.
	<b>ip access-group</b>	Controls access to an interface.
	<b>permit (IP)</b>	Sets conditions for a named IP access list.
	<b>remark</b>	Writes a helpful comment (remark) for an entry in a named IP access list.
	<b>show ip access-list</b>	Displays the contents of all current IP access lists.

# ip access-list resequence

To apply sequence numbers to the access list entries in an access list, use the **ip access-list resequence** command in global configuration mode.

**ip access-list resequence** *access-list-name starting-sequence-number increment*

Syntax Description		
<i>access-list-name</i>	Name of the access list. Names cannot contain a space or quotation mark.	
<i>starting-sequence-number</i>	Access list entries will be resequenced using this initial value. The default value is 10. The range of possible sequence numbers is 1 through 2147483647.	
<i>increment</i>	The number by which the sequence numbers change. The default value is 10. For example, if the increment value is 5 and the beginning sequence number is 20, the subsequent sequence numbers are 25, 30, 25, 40, and so on.	

**Defaults** Disabled

**Command Modes** Global configuration

Command History	Release	Modification
	12.2(14)S	This command was introduced.
	12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.0(33)S	This command was integrated into Cisco IOS Release 12.0(33)S.

**Usage Guidelines** This command allows the **permit** and **deny** entries of a specified access list to be resequenced with an initial sequence number value determined by the *starting-sequence-number* argument, and continuing in increments determined by the *increment* argument. If the highest sequence number exceeds the maximum possible sequence number, then no sequencing occurs.

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 appears:

```
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 appears:

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 in synchronization at all times.

Sequence numbers are not saved in NVRAM. 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.

This command 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 as names as long as they are entered in named access list configuration mode.

### Examples

The following example resequences an access list named kmd1. The starting sequence number is 100, and the increment value is 5:

```
ip access-list resequence kmd1 100 5
```

### Related Commands

Command	Description
<b>deny (IP)</b>	Sets conditions under which a packet does not pass a named IP access list.
<b>permit (IP)</b>	Sets conditions under which a packet passes a named IP access list.

# periodic

To specify a recurring (weekly) time range for functions that support the time-range feature, use the **periodic** command in time-range configuration mode. To remove the time limitation, use the **no** form of this command.

**periodic** *days-of-the-week* *hh:mm* **to** [*days-of-the-week*] *hh:mm*

**no periodic** *days-of-the-week* *hh:mm* **to** [*days-of-the-week*] *hh:mm*

## Syntax Description

*days-of-the-week* The first occurrence of this argument is the starting day or day of the week that the associated time range is in effect. The second occurrence is the ending day or day of the week the associated statement is in effect.

This argument can be any single day or combinations of days: **Monday, Tuesday, Wednesday, Thursday, Friday, Saturday, and Sunday**. Other possible values are:

- **daily**—Monday through Sunday
- **weekdays**—Monday through Friday
- **weekend**—Saturday and Sunday

If the ending days of the week are the same as the starting days of the week, they can be omitted.

*hh:mm* The first occurrence of this argument is the starting hours:minutes that the associated time range is in effect. The second occurrence is the ending hours:minutes the associated statement is in effect.

The hours:minutes are expressed in a 24-hour clock. For example, 8:00 is 8:00 AM and 20:00 is 8:00 PM

**to** Entry of the **to** keyword is required to complete the range “from start-time to end-time.”

## Defaults

No recurring time range is defined.

## Command Modes

Time-range configuration (config-time-range)

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.0(33)S	Support for this command was introduced on the Cisco 12000 series Internet router.

## Usage Guidelines

For Cisco IOS Release 12.2(11)T, IP and Internetwork Packet Exchange (IPX) extended access lists are the only functions that can use time ranges. For information on using these functions, refer to the *Cisco IOS IP Configuration Guide* and the *Cisco IOS AppleTalk and Novell IPX Configuration Guide*.

The **periodic** command is one way to specify when a time range is in effect. Another way is to specify an absolute time period with the **absolute** command. Use either of these commands after the **time-range** global configuration command, which specifies the name of the time range. Multiple **periodic** entries are allowed per **time-range** command.

If the end days-of-the-week value is the same as the start value, they can be omitted.

If a **time-range** command has both **absolute** and **periodic** values specified, then the **periodic** items are evaluated only after the **absolute start** time is reached, and are not further evaluated after the **absolute end** time is reached.

**Note**

All time specifications are taken as local time. To ensure that the time range entries take effect at the desired times, you should synchronize the system software clock using Network Time Protocol (NTP).

Table 2 lists some typical settings for your convenience:

**Table 2** Typical Examples of periodic Command Syntax

If you want:	Configure this:
Monday through Friday, 8:00 AM to 6:00 PM only	<b>periodic weekday 8:00 to 18:00</b>
Every day of the week, from 8:00 AM to 6:00 PM only	<b>periodic daily 8:00 to 18:00</b>
Every minute from Monday 8:00 AM to Friday 8:00 PM	<b>periodic monday 8:00 to friday 20:00</b>
All weekend, from Saturday morning through Sunday night	<b>periodic weekend 00:00 to 23:59</b>
Saturdays and Sundays, from noon to midnight	<b>periodic weekend 12:00 to 23:59</b>

**Examples**

The following example configuration denies HTTP traffic on Monday through Friday from 8:00 AM to 6:00 PM:

```
Router# show startup-config
.
.
.
time-range no-http
  periodic weekdays 8:00 to 18:00
!
ip access-list extended strict
  deny tcp any any eq http time-range no-http
!
interface ethernet 0
  ip access-group strict in
.
.
.
```

The following example configuration permits Telnet traffic on Mondays, Tuesdays, and Fridays from 9:00 AM to 5:00 PM:

```
Router# show startup-config
.
.
.
time-range testing
```

```

periodic Monday Tuesday Friday 9:00 to 17:00
!
ip access-list extended legal
 permit tcp any any eq telnet time-range testing
!
interface ethernet 0
 ip access-group legal in
.
.
.

```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>absolute</b>	Specifies an absolute start and end time for a time range.
<b>access-list (extended)</b>	Defines an extended IP access list.
<b>deny (IP)</b>	Sets conditions under which a packet does not pass a named IP access list.
<b>permit (IP)</b>	Sets conditions under which a packet passes a named IP access list.
<b>time-range</b>	Enables time-range configuration mode and names a time range definition.

## permit (IP)

To set conditions to allow a packet to pass a named IP access list, use the **permit** command in access list configuration mode. To remove a permit condition from an access list, use the **no** form of this command.

```
[sequence-number] permit source [source-wildcard]
```

```
[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]
```

```
no sequence-number
```

```
no permit source [source-wildcard]
```

```
no 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]
```

### Internet Control Message Protocol (ICMP)

```
[sequence-number] permit icmp source source-wildcard destination destination-wildcard  
[icmp-type [icmp-code] | icmp-message] [precedence precedence] [tos tos] [ttl operator value]  
[log] [time-range time-range-name] [fragments]
```

### Internet Group Management Protocol (IGMP)

```
[sequence-number] permit igmp source source-wildcard destination destination-wildcard  
[igmp-type] [precedence precedence] [tos tos] [ttl operator value] [log] [time-range  
time-range-name] [fragments]
```

### Transmission Control Protocol (TCP)

```
[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] [ttl operator value] [log]  
[time-range time-range-name] [fragments]
```

### User Datagram Protocol (UDP)

```
[sequence-number] permit udp source source-wildcard [operator [port]] destination  
destination-wildcard [operator [port]] [precedence precedence] [tos tos] [ttl operator value]  
[log] [time-range time-range-name] [fragments]
```

Syntax	Description
<i>sequence-number</i>	(Optional) Sequence number assigned to the permit statement. The sequence number causes the system to insert the statement in that numbered position in the access list.
<i>source</i>	Number of the network or host from which the packet is being sent. There are three alternative ways to specify the source: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format.</li> <li>Use the <b>any</b> keyword as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host source</b> as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.</li> </ul>
<i>source-wildcard</i>	Wildcard bits to be applied to source. There are three alternative ways to specify the source wildcard: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore.</li> <li>Use the <b>any</b> keyword as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host source</b> as an abbreviation for a <i>source</i> and <i>source-wildcard</i> of <i>source</i> 0.0.0.0.</li> </ul>
<i>protocol</i>	Name or number of an Internet protocol. The <i>protocol</i> argument can be one of the keywords <b>eigrp</b> , <b>gre</b> , <b>icmp</b> , <b>igmp</b> , <b>ip</b> , <b>ipinip</b> , <b>nos</b> , <b>ospf</b> , <b>tcp</b> , or <b>udp</b> , or an integer in the range from 0 to 255 representing an Internet protocol number. To match any Internet protocol (including ICMP, TCP, and UDP), use the <b>ip</b> keyword. <p><b>Note</b> When the <b>icmp</b>, <b>igmp</b>, <b>tcp</b>, and <b>udp</b> keywords are entered, they must be followed with the specific command syntax that is shown for the ICMP, IGMP, TCP, and UDP forms of the <b>permit</b> command.</p>
<b>icmp</b>	Permits only ICMP packets. When you enter the <b>icmp</b> keyword, you must use the specific command syntax shown for the ICMP form of the <b>permit</b> command.
<b>igmp</b>	Permits only IGMP packets. When you enter the <b>igmp</b> keyword, you must use the specific command syntax shown for the IGMP form of the <b>permit</b> command.
<b>tcp</b>	Permits only TCP packets. When you enter the <b>tcp</b> keyword, you must use the specific command syntax shown for the TCP form of the <b>permit</b> command.
<b>udp</b>	Permits only UDP packets. When you enter the <b>udp</b> keyword, you must use the specific command syntax shown for the UDP form of the <b>permit</b> command.
<i>destination</i>	Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination: <ul style="list-style-type: none"> <li>Use a 32-bit quantity in four-part dotted-decimal format.</li> <li>Use the <b>any</b> keyword as an abbreviation for the <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>Use <b>host destination</b> as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.</li> </ul>

<i>destination-wildcard</i>	<p>Wildcard bits to be applied to the destination. There are three alternative ways to specify the destination wildcard:</p> <ul style="list-style-type: none"> <li>• Use a 32-bit quantity in four-part dotted-decimal format. Place 1s in the bit positions that you want to ignore.</li> <li>• Use the <b>any</b> keyword as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of 0.0.0.0 255.255.255.255.</li> <li>• Use <b>host destination</b> as an abbreviation for a <i>destination</i> and <i>destination-wildcard</i> of <i>destination</i> 0.0.0.0.</li> </ul>
<b>option</b> <i>option-name</i>	(Optional) Packets can be filtered by IP Options, as specified by a number from 0 to 255, or by the corresponding IP Option name, as listed in <a href="#">Table 3</a> in the “Usage Guidelines” section.
<b>precedence</b> <i>precedence</i>	(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7 or by a name.
<b>tos</b> <i>tos</i>	(Optional) Packets can be filtered by type of service (ToS) level, as specified by a number from 0 to 15, or by a name as listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.
<b>ttl</b> <i>operator value</i>	<p>(Optional) Compares the TTL value in the packet to the TTL value specified in this <b>permit</b> statement.</p> <ul style="list-style-type: none"> <li>• The <i>operator</i> can be <b>lt</b> (less than), <b>gt</b> (greater than), <b>eq</b> (equal), <b>neq</b> (not equal), or <b>range</b> (inclusive range).</li> <li>• The <i>value</i> can range from 0 to 255.</li> <li>• If the operator is <b>range</b>, specify two values separated by a space.</li> <li>• For Release 12.0S, if the operator is <b>eq</b> or <b>neq</b>, only one TTL value can be specified.</li> <li>• For all other releases, if the operator is <b>eq</b> or <b>neq</b>, as many as 10 TTL values can be specified, separated by a space.</li> </ul>
<b>log</b>	(Optional) Causes an informational logging message about the packet that matches the entry to be sent to the console. (The level of messages logged to the console is controlled by the <b>logging console</b> command.)
<b>time-range</b> <i>time-range-name</i>	(Optional) Name of the time range that applies to this <b>permit</b> statement. The name of the time range and its restrictions are specified by the <b>time-range</b> and <b>absolute</b> or <b>periodic</b> commands, respectively.
<b>fragments</b>	(Optional) The access list entry applies to noninitial fragments of packets; the fragment is either permitted or denied accordingly. For more details about the <b>fragments</b> keyword, see the “ <a href="#">Access List Processing of Fragments</a> ” and “ <a href="#">Fragments and Policy Routing</a> ” sections in the “Usage Guidelines” section.
<i>icmp-type</i>	(Optional) ICMP packets can be filtered by ICMP message type. The type is a number from 0 to 255.
<i>icmp-code</i>	(Optional) ICMP packets that are filtered by ICMP message type can also be filtered by the ICMP message code. The code is a number from 0 to 255.
<i>icmp-message</i>	(Optional) ICMP packets can be filtered by an ICMP message type name or an ICMP message type and code name. The possible names are listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.

<i>igmp-type</i>	(Optional) IGMP packets can be filtered by IGMP message type or message name. A message type is a number from 0 to 15. IGMP message names are listed in the “Usage Guidelines” section of the <b>access-list</b> (IP extended) command.
<i>operator</i>	(Optional) Compares source or destination ports. 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).  If the operator is positioned after the <i>source</i> and <i>source-wildcard</i> arguments, it must match the source port. If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i> arguments, it must match the destination port.  The <b>range</b> operator requires two port numbers. Up to ten port numbers can be entered for the <b>eq</b> (equal) and <b>neq</b> (not equal) operators. All other operators require one port number.
<i>port</i>	(Optional) The decimal number or name of a TCP or UDP port. A port number is a number from 0 to 65535. TCP and UDP port names are listed in the “Usage Guidelines” section of the <b>access-list (IP extended)</b> command.  TCP port names can be used only when filtering TCP. UDP port names can be used only when filtering UDP.
<b>established</b>	(Optional) For the TCP protocol only: Indicates an established connection. A match occurs if the TCP datagram has the ACK or RST bit set. The nonmatching case is that of the initial TCP datagram to form a connection.  <b>Note</b> The <b>established</b> keyword can be used only with the old command-line interface (CLI) format. To use the new CLI format, you must use the <b>match-any</b> or <b>match-all</b> keywords followed by the + or - keywords and <i>flag-name</i> argument.
{ <b>match-any</b>   <b>match-all</b> }	(Optional) For the TCP protocol only: A match occurs if the TCP datagram has certain TCP flags set or not set. You use the <b>match-any</b> keyword to allow a match to occur if any of the specified TCP flags are present, or you can use the <b>match-all</b> keyword to allow a match to occur only if all of the specified TCP flags are present. You must follow the <b>match-any</b> and <b>match-all</b> keywords with the + or - keyword and the <i>flag-name</i> argument to match on one or more TCP flags.
{ +   - } <i>flag-name</i>	(Optional) For the TCP protocol only: The + keyword matches IP packets if their TCP headers contain the TCP flags that are specified by the <i>flag-name</i> argument. The - keyword matches IP packets that do not contain the TCP flags specified by the <i>flag-name</i> argument. You must follow the + and - keywords with the <i>flag-name</i> argument. TCP flag names can be used only when filtering TCP. Flag names for the TCP flags are as follows: <b>urg</b> , <b>ack</b> , <b>psh</b> , <b>rst</b> , <b>syn</b> , and <b>fin</b> .

**Syntax Description**

No specific conditions under which a packet passes the named access list exist.

**Command Modes**

Access list configuration

Command History	Release	Modification
	11.2	This command was introduced.
	12.0(1)T	The <b>time-range</b> <i>time-range-name</i> keyword and argument were added.
	12.0(11)	The <b>fragments</b> keyword was added.
	12.2(13)T	The <b>igrp</b> keyword was removed because the IGRP protocol is no longer available in Cisco IOS software.
	12.2(14)S	The <i>sequence-number</i> argument was added.
	12.2(15)T	The <i>sequence-number</i> argument was added.
	12.3(4)T	The <b>option</b> <i>option-name</i> keyword and argument were added. The <b>match-any</b> , <b>match-all</b> , <b>+</b> and <b>-</b> keywords and the <i>flag-name</i> argument were added.
	12.3(7)T	Command functionality was modified to allow up to ten port numbers to be added after the <b>eq</b> and <b>neq</b> operators so that an access list entry can be created with noncontiguous ports.
	12.4	The <b>drip</b> keyword was added to specify the TCP port number used for OER communication.
	12.4(2)T	The <b>tll</b> <i>operator value</i> keyword and arguments were added.
	12.2(27)SBC	This command was integrated into Cisco IOS Release 12.2(27)SBC.
	12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
	12.0(33)S	Support for the <b>tll</b> <i>operator value</i> keyword and arguments and the <b>time-range</b> <i>time-range-name</i> keyword and argument were introduced on the Cisco 12000 series Internet router.

### Usage Guidelines

Use this command following the **ip access-list** command to define the conditions under which a packet passes the named access list.

The **time-range** keyword allows you to identify a time range by name. The **time-range**, **absolute**, and **periodic** commands specify when this **permit** statement is in effect.

#### log Keyword

A log message includes the access list number, whether the packet was permitted or denied; the protocol, whether it was TCP, UDP, ICMP, or a number; and, if appropriate, the source and destination addresses and source and destination port numbers. The message is generated for the first packet that matches, and then at 5-minute intervals, including the number of packets permitted or denied in the prior 5-minute interval.

Use the **ip access-list log-update** command to generate logging messages when the number of matches reaches a configurable threshold (rather than waiting for a 5-minute-interval). See the **ip access-list log-update** command for more information.

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 because of 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.

If you enable Cisco Express Forwarding (CEF) and then create an access list that uses the **log** keyword, the packets that match the access list are not CEF-switched. They are fast-switched. Logging disables CEF.

**time-range keyword**

The **time-range** keyword allows you to identify a time range by name to specify when a **permit** statement is enabled. Before you enter this parameter, you must first configure a time range using the **time-range** command in global configuration mode and define a time period with the **absolute** and **periodic** commands; for example:

```
time-range no-http
  periodic weekdays 8:00 to 18:00

time-range yes-cust010
  absolute start 12:00 1 January 2006
```

The **time-range** configuration in a **permit** statement is not supported in the following situations:

- When used in a reflexive ACL
- When used in a merged ACL
- On an interface on which policy-based routing (PBR) is already configured.

To use a time-based **permit** statement in a merged ACL, you must first disable merging on the Cisco 12000 series line card. To disable ACL merging, follow these steps:

1. Enter the **hw-module slot slot-number tcam compile acl no-merge** command in global configuration mode.
2. Re-apply the ACL with time-based ACEs on the appropriate interface by entering the **ip access-group access-list-name {in | out}** command in interface configuration mode.

In general, the exact time when a time-based **permit** statement is enabled and disabled varies from the configured time depending on CPU utilization, number and length of ACLs being processed, and number of TCAM entries that correspond to an ACE. Take these factors into account when you configure time ranges for a **permit** statement.

**Access List Filtering of IP Options**

Access control lists can be used to filter packets with IP Options to prevent routers from being saturated with spurious packets containing IP Options. To see a complete table of all IP Options, including ones currently not in use, refer to the latest Internet Assigned Numbers Authority (IANA) information that is available from their URL: [www.iana.org](http://www.iana.org).

Cisco IOS software allows you to filter packets according to whether they contain one or more of the legitimate IP Options by entering either the IP Option value or the corresponding name for the *option-name* argument as shown in [Table 3](#).

**Table 3** IP Option Values and Names

IP Option Value or Name	Description
0 to 255	IP Options values.
add-ext	Match packets with Address Extension Option (147).
any-options	Match packets with any IP Option.
com-security	Match packets with Commercial Security Option (134).
dps	Match packets with Dynamic Packet State Option (151).
encode	Match packets with Encode Option (15).
eool	Match packets with End of Options (0).
ext-ip	Match packets with Extended IP Options (145).

**Table 3** IP Option Values and Names (continued)

IP Option Value or Name	Description
ext-security	Match packets with Extended Security Option (133).
finn	Match packets with Experimental Flow Control Option (205).
imitd	Match packets with IMI Traffic Descriptor Option (144).
lsr	Match packets with Loose Source Route Option (131).
mtup	Match packets with MTU Probe Option (11).
mtur	Match packets with MTU Reply Option (12).
no-op	Match packets with the No Operation Option (1).
nsapa	Match packets with the NSAP Addresses Option (150).
record-route	Match packets with Router Record Route Option (7).
router-alert	Match packets with Router Alert Option (148).
sdb	Match packets with Selective Directed Broadcast Option (149).
security	Match packets with Base Security Option (130).
ssr	Match packets with Strict Source Routing Option (137).
stream-id	Match packets with Stream ID Option (136).
timestamp	Match packets with Time Stamp Option (68).
traceroute	Match packets with Trace Route Option (82).
ump	Match packets with Upstream Multicast Packet Option (152).
visa	Match packets with Experimental Access Control Option (142).
zsu	Match packets with Experimental Measurement Option (10).

### Filtering IP Packets Based on TCP Flags

The access list entries that make up an access list can be configured to detect and drop unauthorized TCP packets by allowing only the packets that have very specific groups of TCP flags set or not set. Users can select any desired combination of TCP flags with which to filter TCP packets. Users can configure access list entries in order to allow matching on a flag that is set and on a flag that is not set. Use the **+** and **-** keywords with a flag name to specify that a match is made based on whether a TCP header flag has been set. Use the **match-any** and **match-all** keywords to allow the packet if any or all, respectively, of the flags specified by the **+** or **-** keyword and *flag-name* argument have been set or not set.

### Permitting Optimized Edge Routing (OER) Communication

The **drip** keyword was introduced under the **tcp** keyword to support packet filtering in a network where OER is configured. The **drip** keyword specifies port 3949 that OER uses for internal communication. This option allows you to build a packet filter that permits communication between an OER master controller and border router(s). The **drip** keyword is entered following the TCP source, destination addresses, and the **eq** operator. See the example at the end of this command reference page.

### 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 behavior), and assuming all of the access-list entry information matches,	For an access list entry that contains only Layer 3 information: <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets, initial fragments, and noninitial fragments.</li> </ul> For an access list entry that contains Layer 3 and Layer 4 information: <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,	The access list entry is applied only to noninitial fragments. The <b>fragments</b> keyword cannot be configured for an access list entry that contains any Layer 4 information.

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** access list 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.

**Note**


---

The **fragments** keyword cannot solve all cases that involve access lists and IP fragments.

---

**Fragments and Policy Routing**

Fragmentation and the fragment control feature affect policy routing if the policy routing is based on the **match ip address** command and the access list has entries that match on Layer 4 through 7 information. It is possible that noninitial fragments pass the access list and are policy-routed, even if the first fragment is not policy-routed.

By using the **fragments** keyword in access list entries as described earlier, a better match between the action taken for initial and noninitial fragments can be made, and it is more likely that policy routing will occur as intended.

**Creating an Access List Entry with Noncontiguous Ports**

For Cisco IOS Release 12.3(7)T and later releases, you can specify noncontiguous ports on the same access control entry, which greatly reduces the number of access list entries required for the same source address, destination address, and protocol. If you maintain large numbers of access list entries, we recommend that you consolidate them when possible by using noncontiguous ports. You can specify up to ten port numbers following the **eq** and **neq** operators.

**Examples**

The following example sets conditions for a standard access list named Internetfilter:

```
ip access-list standard Internetfilter
 deny 192.168.34.0 0.0.0.255
 permit 172.16.0.0 0.0.255.255
 permit 10.0.0.0 0.255.255.255
! (Note: all other access implicitly denied).
```

The following example permits Telnet traffic on Mondays, Tuesdays, and Fridays from 9:00 AM to 5:00 PM:

```
time-range testing
 periodic Monday Tuesday Friday 9:00 to 17:00
!
ip access-list extended legal
 permit tcp any any eq telnet time-range testing
!
interface ethernet0
 ip access-group legal in
```

The following example sets a permit condition for an extended access list named filter2. The access list entry specifies that a packet may pass the named access list only if it contains the NSAP Addresses IP Option, which is represented by the IP Option value nsapa.

```
ip access-list extended filter2
 permit ip any any option nsapa
```

The following example sets a permit condition for an extended access list named kmdfilter1. The access list entry specifies that a packet can pass the named access list only if the RST IP flag has been set for that packet:

```
ip access-list extended kmdfilter1
 permit tcp any any match-any +rst
```

The following example sets a permit condition for an extended access list named `kmdfilter1`. The access list entry specifies that a packet can pass the named access list only if the RST and FIN TCP flags have been set for that packet:

```
ip access-list extended kmdfilter1
 permit tcp any any match-any +rst +fin
```

The following example shows how to verify the access list by using the `show access-lists` command and then to add an entry to an existing access list:

```
Router# show access-lists

Standard IP access list 1
 2 permit 10.0.0.0, wildcard bits 0.0.255.255
 5 permit 10.0.0.0, wildcard bits 0.0.255.255
10 permit 10.0.0.0, wildcard bits 0.0.255.255
20 permit 10.0.0.0, wildcard bits 0.0.255.255

ip access-list standard 1
 15 permit 10.0.0.0 0.0.255.255
```

The following examples shows how the entry with the sequence number of 20 is removed from the access list:

```
ip access-list standard 1
 no 20

!Verify that the list has been removed.
```

```
Router# show access-lists

Standard IP access list 1
10 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 0.4.0.0, wildcard bits 0.0.0.255
```

The following examples shows how, if a user tries to enter an entry that is a duplicate of an entry already on the list, no changes occur. The entry that the user is trying to add is a duplicate of the entry already in the access list with a sequence number of 20.

```
Router# show access-lists 101

Extended IP access list 101
 10 permit ip host 10.0.0.0 host 10.5.5.34
 20 permit icmp any any
 30 permit ip host 10.0.0.0 host 10.2.54.2
 40 permit ip host 10.0.0.0 host 10.3.32.3 log

ip access-list extended 101
 100 permit icmp any any

Router# show access-lists 101

Extended IP access list 101
 10 permit ip host 10.3.3.3 host 10.5.5.34
 20 permit icmp any any
 30 permit ip host 10.34.2.2 host 10.2.54.2
 40 permit ip host 10.3.4.31 host 10.3.32.3 log
```

The following example shows what occurs if a user tries to enter a new entry with a sequence number of 20 when an entry with a sequence number of 20 is already in the list. An error message appears, and no change is made to the access list.

```
Router# show access-lists 101
```

```

Extended IP access lists 101
  10 permit ip host 10.3.3.3 host 10.5.5.34
  20 permit icmp any any
  30 permit ip host 10.34.2.2 host 10.2.54.2
  40 permit ip host 10.3.4.31 host 10.3.32.3 log

ip access-lists extended 101
  20 permit udp host 10.1.1.1 host 10.2.2.2

```

%Duplicate sequence number.

```
Router# show access-lists 101
```

```

Extended IP access lists 101
  10 permit ip host 10.3.3.3 host 10.5.5.34
  20 permit icmp any any
  30 permit ip host 10.34.2.2 host 10.2.54.2
  40 permit ip host 10.3.4.31 host 10.3.32.3 log

```

The following example shows several **permit** statements that can be consolidated into one access list entry with noncontiguous ports. The **show access-lists** command is entered to display a group of access list entries for the access list named **aaa**.

```
Router# show access-lists aaa
```

```

Extended IP access lists aaa
  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 aaa
  no 10
  no 20
  no 30
  no 40
  permit tcp any eq telnet ftp any eq 450 679

```

The following example shows the creation of the consolidated access list entry:

```
Router# show access-lists aaa
```

```

Extended IP access list aaa
  10 permit tcp any eq telnet ftp any eq 450 679

```

The following access list filters IP packets containing Type of Service (ToS) level 3 with 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 not equal to 1, and sends log messages about such packets to the console. All other packets are denied.

```

ip access-list extended canton
  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

```

The following example shows how to configure a packet filter, for any TCP source and destination, that permits communication between an OER master controller and border router:

```
ip access-list extended 100
 permit any any tcp eq drip
 exit
```

Related Commands	Command	Description
	<b>absolute</b>	Specifies an absolute time when a time range is in effect.
	<b>access-list (IP extended)</b>	Defines an extended IP access list.
	<b>access-list (IP standard)</b>	Defines a standard IP access list.
	<b>deny (IP)</b>	Sets conditions under which a packet does not pass a named IP access list.
	<b>ip access-group</b>	Controls access to an interface.
	<b>ip access-list log-update</b>	Sets the threshold number of packets that cause a logging message.
	<b>ip access-list resequence</b>	Applies sequence numbers to the access list entries in an access list.
	<b>ip options</b>	Drops or ignores IP Options packets that are sent to the router.
	<b>logging console</b>	Sends system logging (syslog) messages to all available TTY lines and limits messages based on severity.
	<b>match ip address</b>	Distributes any routes that have a destination network number address that is permitted by a standard or extended access list, or performs policy routing on packets.
	<b>periodic</b>	Specifies a recurring (weekly) time range for functions that support the time-range feature.
	<b>show access-lists</b>	Displays a group of access-list entries.
	<b>show ip access-list</b>	Displays the contents of all current IP access lists.
	<b>time-range</b>	Specifies when an access list or other feature is in effect.

# time-range

To enable time-range configuration mode and define time ranges for functions (such as extended access lists), use the **time-range** command in global configuration mode. To remove the time limitation, use the **no** form of this command.

**time-range** *time-range-name*

**no time-range** *time-range-name*

## Syntax Description

*time-range-name* Desired name for the time range. The name cannot contain a space or quotation mark, and must begin with a letter.

## Defaults

None

## Command Modes

Global configuration

## Command History

Release	Modification
12.0(1)T	This command was introduced.
12.2(17a)SX	Support for this command was introduced on the Cisco 7600 series routers.
12.2(17d)SXB	Support for this command on the Supervisor Engine 2 was extended to 12.2(17d)SXB.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.0(33)S	Support for this command was introduced on the Cisco 12000 series internet router.

## Usage Guidelines

The **time-range** entries are identified by a name, which is referred to by one or more other configuration commands. Multiple time ranges can occur in a single access list or other feature.



### Note

In Cisco IOS 12.2SX releases, IP and IPX-extended access lists are the only types of access lists that can use time ranges.

After the **time-range** command, use the **periodic** time-range configuration command, the **absolute** time-range configuration command, or some combination of them to define when the feature is in effect. Multiple **periodic** commands are allowed in a time range; only one **absolute** command is allowed.



### Tip

To avoid confusion, use different names for time ranges and named access lists.

## Examples

The following example denies HTTP traffic on Monday through Friday from 8:00 AM to 6:00 PM. The example allows UDP traffic on Saturday and Sunday from 12:00 noon to midnight only.

```

time-range no-http
  periodic weekdays 8:00 to 18:00
!
time-range udp-yes
  periodic weekend 12:00 to 24:00
!
ip access-list extended strict
  deny tcp any any eq http time-range no-http
  permit udp any any time-range udp-yes
!
interface ethernet 0
  ip access-group strict in

```

**Related Commands**

<b>Command</b>	<b>Description</b>
<b>absolute</b>	Specifies an absolute start and end time for a time range.
<b>ip access-list</b>	Defines an IP access list by name.
<b>periodic</b>	Specifies a recurring (weekly) start and end time for a time range.
<b>permit (IP)</b>	Sets conditions under which a packet passes a named IP access list.

# Glossary


**Note**

See [Internetworking Terms and Acronyms](#) for a description of the terms used in this document.

## Feature Information for ACL Enhancements on the Cisco 12000 Series Router

[Table 4](#) lists the release history for this feature.

Not all commands may be available in your Cisco IOS software release. For release information about a specific command, see the command reference documentation.

Cisco IOS software images are specific to a Cisco IOS software release, a feature set, and a platform. Use Cisco Feature Navigator to find information about platform support and Cisco IOS software image support. Access Cisco Feature Navigator at <http://www.cisco.com/go/fn>. You must have an account on Cisco.com. If you do not have an account or have forgotten your username or password, click **Cancel** at the login dialog box and follow the instructions.


**Note**

[Table 4](#) lists only the Cisco IOS software release that introduced support for a given feature in a given Cisco IOS software release train. Unless noted otherwise, subsequent releases of that Cisco IOS software release train also support that feature.

**Table 4** Feature Information for ACL Enhancements on the Cisco 12000 Series Router

Feature Name	Releases	Feature Information
ACL Enhancements on the Cisco 12000 series router: <ul style="list-style-type: none"> <li>• IPv4 Named ACLs on distributed switch engines (IP Services Engine/Engine 3 and Engine 5)</li> <li>• ACL sequence numbering</li> <li>• Time-based ACEs</li> <li>• Time to Live (TTL)-based ACEs</li> </ul>	12.0(33)S	In Cisco IOS Release 12.0(33)S, these ACL enhancements were introduced on IP Services Engine (ISE/E3) and Engine 5 line cards on the Cisco 12000 series router. The following sections provide information about these ACL enhancements: <ul style="list-style-type: none"> <li>• <a href="#">Using Named Instead of Numbered Access Lists, page 8</a></li> <li>• <a href="#">Applying Sequence Numbers to Entries in a Named Access List, page 9</a></li> <li>• <a href="#">Using Time-Based Access List Entries, page 10</a></li> <li>• <a href="#">Using Time-to-Live Access List Entries, page 10</a></li> </ul> The following commands were introduced or modified by ACL Enhancements: absolute, deny (IP), ip access-list, ip access-list resequence, periodic, permit (IP), time-range.

CCDE, CCENT, Cisco Eos, Cisco Lumin, Cisco Nexus, Cisco StadiumVision, the Cisco logo, DCE, and Welcome to the Human Network are trademarks; Changing the Way We Work, Live, Play, and Learn is a service mark; and Access Registrar, Aironet, AsyncOS, Bringing the Meeting To You, Catalyst, CCDA, CCDP, CCIE, CCIP, CCNA, CCNP, CCSP, CCVP, Cisco, the Cisco Certified Internetwork Expert logo, Cisco IOS,

Cisco Press, Cisco Systems, Cisco Systems Capital, the Cisco Systems logo, Cisco Unity, Collaboration Without Limitation, EtherFast, EtherSwitch, Event Center, Fast Step, Follow Me Browsing, FormShare, GigaDrive, HomeLink, Internet Quotient, IOS, iPhone, iQ Expertise, the iQ logo, iQ Net Readiness Scorecard, iQuick Study, IronPort, the IronPort logo, LightStream, Linksys, MediaTone, MeetingPlace, MGX, Networkers, Networking Academy, Network Registrar, PCNow, PIX, PowerPanels, ProConnect, ScriptShare, SenderBase, SMARTnet, Spectrum Expert, StackWise, The Fastest Way to Increase Your Internet Quotient, TransPath, WebEx, and the WebEx logo are registered trademarks of Cisco Systems, Inc. and/or its affiliates in the United States and certain other countries.

---

All other trademarks mentioned in this document or Website 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. (0805R)

---

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

© 2008 Cisco Systems, Inc. All rights reserved.

