



# IPSec NAT Transparency

---

The IPSec NAT Transparency feature introduces support for IP Security (IPSec) traffic to travel through Network Address Translation (NAT) or Port Address Translation (PAT) points in the network by addressing many known incompatibilities between NAT and IPSec.

## Feature Specifications for the IPSec NAT Transparency feature

---

### Feature History

Release	Modification
12.2(13)T	This feature was introduced.

---

### Supported Platforms

For platforms supported in Cisco IOS Release 12.2(13)T, consult Cisco Feature Navigator.

---

## Determining Platform Support Through Cisco Feature Navigator

Cisco IOS software is packaged in feature sets that are supported on specific platforms. To get updated information regarding platform support for this feature, access Cisco Feature Navigator. Cisco Feature Navigator dynamically updates the list of supported platforms as new platform support is added for the feature.

Cisco Feature Navigator is a web-based tool that enables you to determine which Cisco IOS software images support a specific set of features and which features are supported in a specific Cisco IOS image. You can search by feature or release. Under the release section, you can compare releases side by side to display both the features unique to each software release and the features in common.

To access Cisco Feature Navigator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to [cco-locksmith@cisco.com](mailto:cco-locksmith@cisco.com). An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register>

Cisco Feature Navigator is updated regularly when major Cisco IOS software releases and technology releases occur. For the most current information, go to the Cisco Feature Navigator home page at the following URL:

<http://www.cisco.com/go/fn>

**Availability of Cisco IOS Software Images**

Platform support for particular Cisco IOS software releases is dependent on the availability of the software images for those platforms. Software images for some platforms may be deferred, delayed, or changed without prior notice. For updated information about platform support and availability of software images for each Cisco IOS software release, refer to the online release notes or, if supported, Cisco Feature Navigator.

## Contents

- [Restrictions for IPSec NAT Transparency, page 2](#)
- [Information About IPSec NAT Transparency, page 2](#)
- [How to Configure NAT and IPSec, page 7](#)
- [Configuration Examples for IPSec and NAT, page 9](#)
- [Additional References, page 9](#)
- [Command Reference, page 11](#)
- [Glossary, page 22](#)

## Restrictions for IPSec NAT Transparency

Although this feature addresses many incompatibilities between NAT and IPSec, the following problems still exist:

**Internet Key Exchange (IKE) IP Address and NAT**

This incompatibility applies only when IP addresses are used as a search key to find a preshared key. Modification of the IP source or destination addresses by NAT or reverse NAT results in a mismatch between the IP address and the preshared key.

**Embedded IP Addresses and NAT**

Because the payload is integrity protected, any IP address enclosed within IPSec packets cannot be translated by NAT. Protocols that use embedded IP addresses include FTP, Internet Relay Chat (IRC), Simple Network Management Protocol (SNMP), Lightweight Directory Access Protocol (LDAP), H.323, and Session Initiation Protocol (SIP).

## Information About IPSec NAT Transparency

To configure the IPSec NAT Transparency feature, you must understand the following concepts:

- [Benefit of IPSec NAT Transparency, page 3](#)
- [Feature Design of IPSec NAT Traversal, page 3](#)
- [NAT Keepalives, page 6](#)

## Benefit of IPSec NAT Transparency

Before the introduction of this feature, a standard IPSec virtual private network (VPN) tunnel would not work if there were one or more NAT or PAT points in the delivery path of the IPSec packet. This feature makes NAT IPSec-aware, thereby, allowing remote access users to build IPSec tunnels to home gateways.

## Feature Design of IPSec NAT Traversal

The IPSec NAT Transparency feature introduces support for IPSec traffic to travel through NAT or PAT points in the network by encapsulating IPSec packets in a User Datagram Protocol (UDP) wrapper, which allows the packets to travel across NAT devices. The following sections define the details of NAT traversal:

- [IKE Phase 1 Negotiation: NAT Detection](#)
- [IKE Phase 2 Negotiation: NAT Traversal Decision](#)
- [UDP Encapsulation of IPSec Packets for NAT Traversal](#)
- [UDP Encapsulated Process for Software Engines: Transport Mode and Tunnel Mode ESP Encapsulation](#)

### IKE Phase 1 Negotiation: NAT Detection

During Internet Key Exchange (IKE) phase 1 negotiation, two types of NAT detection occur before IKE Quick Mode begins—NAT support and NAT existence along the network path.

To detect NAT support, you should exchange the vendor identification (ID) string with the remote peer. During Main Mode (MM) 1 and MM 2 of IKE phase 1, the remote peer sends a vendor ID string payload to its peer to indicate that this version supports NAT traversal. Thereafter, NAT existence along the network path can be determined.

Detecting whether NAT exists along the network path allows you to find any NAT device between two peers and the exact location of NAT. A NAT device can translate the private IP address and port to public value (or from public to private). This translation changes the IP address and port if the packet goes through the device. To detect whether a NAT device exists along the network path, the peers should send a payload with hashes of the IP address and port of both the source and destination address from each end. If both ends calculate the hashes and the hashes match, each peer knows that a NAT device does not exist on the network path between them. If the hashes do not match (that is, someone translated the address or port), then each peer needs to perform NAT traversal to get the IPSec packet through the network.

The hashes are sent as a series of NAT discovery (NAT-D) payloads. Each payload contains one hash; if multiple hashes exist, multiple NAT-D payloads are sent. In most environments, there are only two NAT-D payloads—one for the source address and port and one for the destination address and port. The destination NAT-D payload is sent first, followed by the source NAT-D payload, which implies that the receiver should expect to process the local NAT-D payload first and the remote NAT-D payload second. The NAT-D payloads are included in the third and fourth messages in Main Mode and in the second and third messages in Aggressive Mode (AM).

## IKE Phase 2 Negotiation: NAT Traversal Decision

While IKE phase 1 detects NAT support and NAT existence along the network path, IKE phase 2 decides whether or not the peers at both ends will use NAT traversal. Quick Mode (QM) security association (SA) payload in QM1 and QM2 is used to for NAT traversal negotiation.

Because the NAT device changes the IP address and port number, incompatibilities between NAT and IPSec can be created. Thus, exchanging the original source address bypasses any incompatibilities.

## UDP Encapsulation of IPSec Packets for NAT Traversal

In addition to allowing IPSec packets to traverse across NAT devices, UDP encapsulation also addresses many incompatibility issues between IPSec and NAT and PAT. The resolved issues are as follows:

### Incompatibility Between IPSec ESP and PAT—Resolved

If PAT found a legislative IP address and port, it would drop the Encapsulating Security Payload (ESP) packet. To prevent this scenario, UDP encapsulation is used to hide the ESP packet behind the UDP header. Thus, PAT treats the ESP packet as a UDP packet, processing the ESP packet as a normal UDP packet.

### Incompatibility Between Checksums and NAT—Resolved

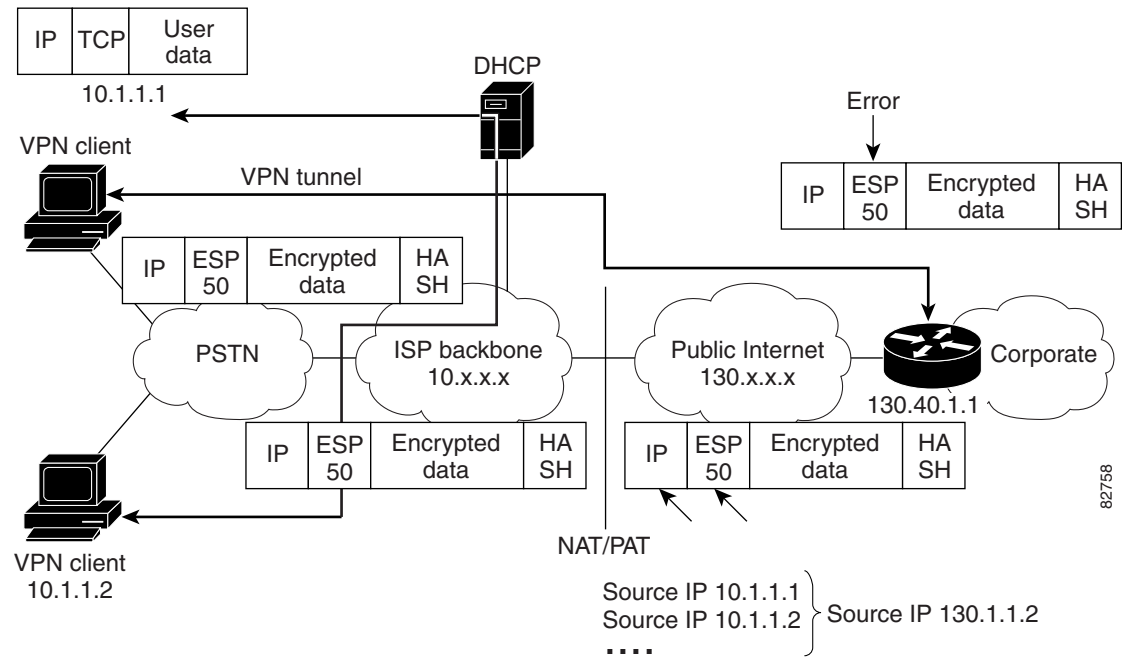
In the new UDP header, the checksum value is always assigned to zero. This value prevents an intermediate device from validating the checksum against the packet checksum, thereby, resolving the TCP UDP checksum issue because NAT changes the IP source and destination addresses.

### Incompatibility Between Fixed IKE Destination Ports and PAT—Resolved

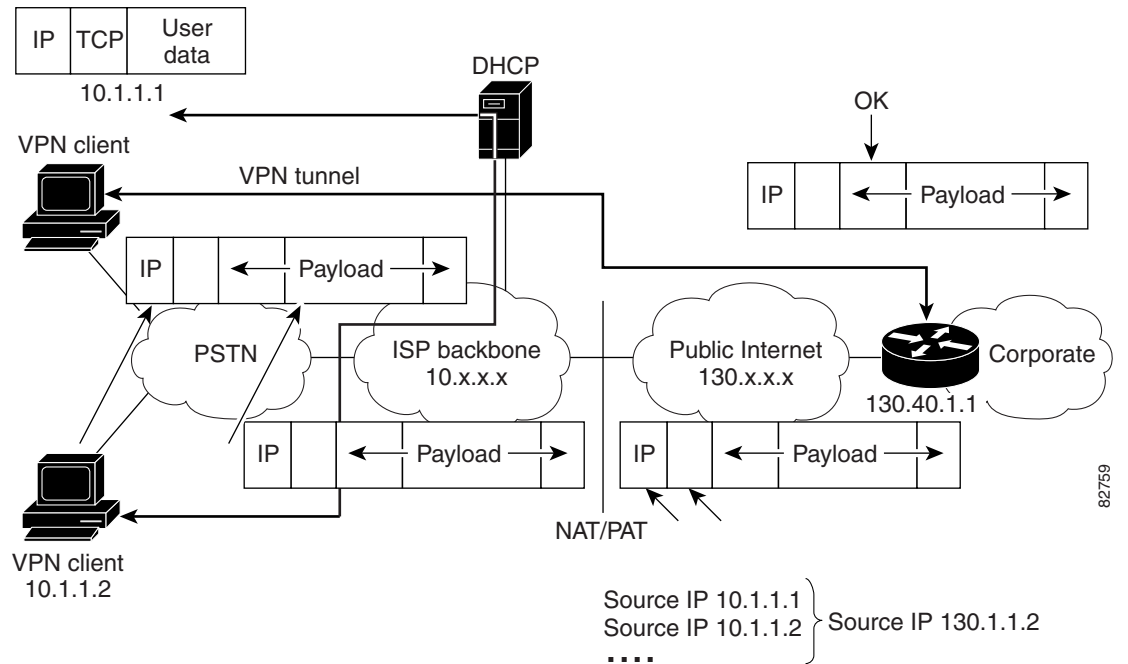
PAT changes the port address in the new UDP header for translation and leaves the original payload unchanged.

To see how UDP encapsulation helps to send IPSec packets see [Figure 1](#) and [Figure 2](#).

**Figure 1 Standard IPSec Tunnel Through a NAT/PAT Point (No UDP Encapsulation)**



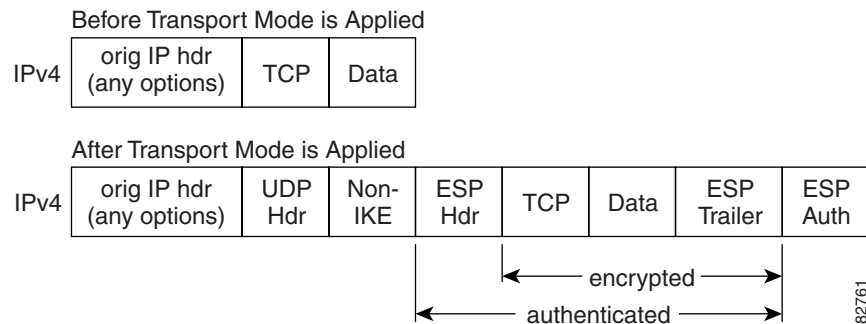
**Figure 2 IPSec Packet with UDP Encapsulation**



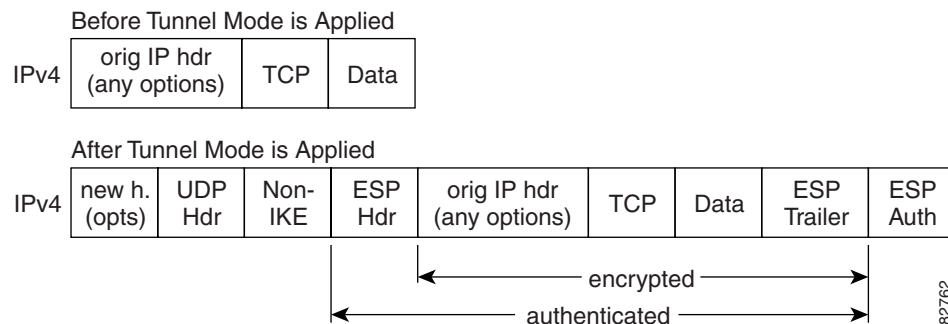
## UDP Encapsulated Process for Software Engines: Transport Mode and Tunnel Mode ESP Encapsulation

After the IPSec packet is encrypted by a hardware accelerator or a software crypto engine, a UDP header and a non-IKE marker (which is 8 bytes in length) are inserted between the original IP header and ESP header. The total length, protocol, and checksum fields are changed to match this modification. [Figure 3](#) shows an IPSec packet before and after transport mode is applied; [Figure 4](#) shows an IPSec packet before and after tunnel mode is applied.

**Figure 3 Transport Mode—IPSec Packet Before and After ESP Encapsulation**



**Figure 4 Tunnel Mode—IPSec Packet Before and After ESP Encapsulation**



## NAT Keepalives

NAT keepalives are enabled to keep the dynamic NAT mapping alive during a connection between two peers. NAT keepalives are UDP packets with an unencrypted payload of 1 byte. Although the current dead peer detection (DPD) implementation is similar to NAT keepalives, there is a slight difference: DPD is used to detect peer status, while NAT keepalives are sent if the IPSec entity did not send or receive the packet at a specified period of time—valid range is between 5 to 3600 seconds.

If NAT keepalives are enabled (via the `crypto isamkp nat keepalive` command), users should ensure that the idle value is shorter than the NAT mapping expiration time, which is 20 seconds.

# How to Configure NAT and IPSec

This section contains the following procedures:

- [Configuring NAT Traversal, page 7](#) (optional)
- [Disabling NAT Traversal, page 7](#) (optional)
- [Configuring NAT Keepalives, page 8](#) (optional)
- [Verifying IPSec Configuration, page 8](#) (optional)

## Configuring NAT Traversal

NAT Traversal is a feature that is auto detected by VPN devices. There are no configuration steps for a router running Cisco IOS Release 12.2(13)T. If both VPN devices are NAT-T capable, NAT Traversal is auto detected and auto negotiated.

## Disabling NAT Traversal

You may wish to disable NAT traversal if you already know that your network uses IPSec-awareness NAT (spi-matching scheme). To disable NAT traversal, use the following commands:

### SUMMARY STEPS:

1. `enable`
2. `configure terminal`
3. `no crypto ipsec nat-transparency udp-encapsulation`

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<code>enable</code>  <b>Example:</b> Router> enable	Enables higher privilege levels, such as privileged EXEC mode.  Enter your password if prompted.
Step 2	<code>configure terminal</code>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<code>no crypto ipsec nat-transparency udp-encapsulation</code>  <b>Example:</b> Router(config)# no crypto ipsec nat-transparency udp-encapsulation	Disables NAT traversal.

## Configuring NAT Keepalives

To configure your router to send NAT keepalives, use the following commands:

### SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto isakmp nat keepalive *seconds***

### DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables higher privilege levels, such as privileged EXEC mode.  Enter your password if prompted.
Step 2	<b>configure terminal</b>  <b>Example:</b> Router# configure terminal	Enters global configuration mode.
Step 3	<b>crypto isakmp nat keepalive <i>seconds</i></b>  <b>Example:</b> Router(config)# crypto isakmp nat keepalive 20	Allows an IPSec node to send NAT keepalive packets. <ul style="list-style-type: none"> <li>• <i>seconds</i>—The number of seconds between keepalive packets; range is between 5 to 3,600 seconds.</li> </ul>

## Verifying IPSec Configuration

To verify your configuration, perform the following optional steps:

### SUMMARY STEPS

1. **enable**
2. **show crypto ipsec sa [map *map-name* | address | identity] [detail]**

## DETAILED STEPS

	Command or Action	Purpose
Step 1	<b>enable</b>  <b>Example:</b> Router> enable	Enables higher privilege levels, such as privileged EXEC mode.  Enter your password if prompted.
Step 2	<b>show crypto ipsec sa</b> [ <b>map</b> <i>map-name</i>   <b>address</b>   <b>identity</b> ] [ <b>detail</b> ]  <b>Example:</b> Router# show crypto ipsec sa	Displays the settings used by current SAs.

## Configuration Examples for IPSec and NAT

This section provides the following configuration example:

- [NAT Keepalives Configuration Example, page 9](#)

### NAT Keepalives Configuration Example

The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
 authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
 set peer 56.0.0.1
 set transform-set t2
 match address 101
```

## Additional References

The following sections provide additional references related to IPSec NAT Transparency:

- [Related Documents, page 10](#)
- [Standards, page 10](#)
- [MIBs, page 10](#)
- [RFCs, page 11](#)
- [Technical Assistance, page 11](#)

## Related Documents

Related Topic	Document Title
Additional NAT configuration tasks.	The chapter “Configuring IP Addressing” in the <i>Cisco IOS IP Configuration Guide</i> , Release 12.2
Additional NAT commands	The chapter “IP Addressing Commands” in the <i>Cisco IOS IP Command Reference, Volume 1 of 3: Addressing and Services</i> , Release 12.2
Additional IPSec configuration tasks	The chapter “Configuring IPSec Network Security” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.2
Additional IPSec commands	The chapter “IPSec Network Security Commands” in the <i>Cisco IOS Security Command Reference</i> , Release 12.2
Information on IKE phase 1 and phase 2, Aggressive Mode, and Main Mode.	The chapter “Configuring Internet Key Exchange Security Protocol” in the <i>Cisco IOS Security Configuration Guide</i> , Release 12.2
Additional information on IKE dead peer detection.	<i>Easy VPN Server</i> , Cisco IOS Release 12.2(8)T feature module

## Standards

Standards	Title
None	—

## MIBs

MIBs	MIBs Link
None	To obtain lists of supported MIBs by platform and Cisco IOS release, and to download MIB modules, go to the Cisco MIB website on Cisco.com at the following URL: <a href="http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml">http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml</a>

To locate and download MIBs for selected platforms, Cisco IOS releases, and feature sets, use Cisco MIB Locator found at the following URL:

<http://tools.cisco.com/ITDIT/MIBS/servlet/index>

If Cisco MIB Locator does not support the MIB information that you need, you can also obtain a list of supported MIBs and download MIBs from the Cisco MIBs page at the following URL:

<http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml>

To access Cisco MIB Locator, you must have an account on Cisco.com. If you have forgotten or lost your account information, send a blank e-mail to [cco-locksmith@cisco.com](mailto:cco-locksmith@cisco.com). An automatic check will verify that your e-mail address is registered with Cisco.com. If the check is successful, account details with a new random password will be e-mailed to you. Qualified users can establish an account on Cisco.com by following the directions found at this URL:

<http://www.cisco.com/register>

## RFCs

RFCs <sup>1</sup>	Title
RFC 2402	<i>IP Authentication Header</i>
RFC 2406	<i>IP Encapsulating Security Payload (ESP)</i>

1. Not all supported RFCs are listed.

## Technical Assistance

Description	Link
Technical Assistance Center (TAC) home page, containing 30,000 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 even more content.	<a href="http://www.cisco.com/public/support/tac/home.shtml">http://www.cisco.com/public/support/tac/home.shtml</a>

## Command Reference

This section documents new and modified commands. All other commands used with this feature are documented in the Cisco IOS Release 12.2 Tcommand reference publications.

### New Command

- [crypto isamkp nat keepalive](#)

### Modified Commands

- [access-list \(IP extended\)](#)
- [show crypto ipsec sa](#)

# crypto isakmp nat keepalive

To allow an IP Security (IPSec) node to send Network Address Translation (NAT) keepalive packets, use the **crypto isakmp nat keepalive** command in global configuration mode. To disable NAT keepalive packets, use the **no** form of this command.

**crypto isakmp nat keepalive** *seconds*

**no crypto isakmp nat keepalive**

<b>Syntax Description</b>	<i>seconds</i>	Number of seconds between keepalive packets; range is between 5 to 3600 seconds.
---------------------------	----------------	--

**Defaults** NAT keepalive packets are not sent.

**Command Modes** Global configuration

<b>Command History</b>	<b>Release</b>	<b>Modification</b>
	12.2(13)T	This command was introduced.

**Usage Guidelines** The **crypto isakmp nat keepalive** command allows users to keep the dynamic NAT mapping alive during a connection between two peers. A NAT keepalive beat is sent if IPSec does not send or receive a packet within a specified time period—valid range is between 5 to 3600 seconds.

If this command is enabled, users should ensure that the idle value is shorter than than the NAT mapping expiration time, which is 20 seconds.

**Examples** The following example shows how to enable NAT keepalives to be sent every 20 seconds:

```
crypto isakmp policy 1
  authentication pre-share
crypto isakmp key 1234 address 56.0.0.1
crypto isakmp nat keepalive 20
!
!
crypto ipsec transform-set t2 esp-des esp-sha-hmac
!
crypto map test2 10 ipsec-isakmp
  set peer 56.0.0.1
  set transform-set t2
  match address 101
```

## access-list (IP extended)

To define an extended IP access list, use the extended version of the **access-list** global configuration command. To remove the access lists, use the **no** form of this command.

### User Datagram Protocol (UDP)

For UDP, you can also use the following syntax:

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

### Syntax Description

<i>access-list-number</i>	Number of an access list. This is a decimal number from 100 to 199 or from 2000 to 2699.
<b>dynamic</b> <i>dynamic-name</i>	(Optional) Identifies this access list as a dynamic access list. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the <i>Cisco IOS Security Configuration Guide</i> .
<b>timeout</b> <i>minutes</i>	(Optional) Specifies the absolute length of time, in minutes, that a temporary access list entry can remain in a dynamic access list. The default is an infinite length of time and allows an entry to remain permanently. Refer to lock-and-key access documented in the “Configuring Lock-and-Key Security (Dynamic Access Lists)” chapter in the <i>Cisco IOS Security Configuration Guide</i> .
<b>deny</b>	Denies access if the conditions are matched.
<b>permit</b>	Permits access if the conditions are matched.
<i>protocol</i>	Name or number of an Internet protocol. It can be one of the keywords <b>eigrp</b> , <b>gre</b> , <b>icmp</b> , <b>igmp</b> , <b>igrp</b> , <b>ip</b> , <b>ipinip</b> , <b>nos</b> , <b>ospf</b> , <b>pim</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. Some protocols allow further qualifiers described below.
<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>	<p>Wildcard bits to be applied to source. Each wildcard bit 0 indicates the corresponding bit position in the source. Each wildcard bit set to 1 indicates that both a 0 bit and a 1 bit in the corresponding position of the IP address of the packet will be considered a match to this access list entry.</p> <p>There are three alternative ways to specify the source 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 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> <p>Wildcard bits set to 1 need not be contiguous in the source wildcard. For example, a source wildcard of 0.255.0.64 would be valid.</p>
<i>destination</i>	<p>Number of the network or host to which the packet is being sent. There are three alternative ways to specify the destination:</p> <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 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>precedence</b> <i>precedence</i>	<p>(Optional) Packets can be filtered by precedence level, as specified by a number from 0 to 7, or by name as listed in the section “Usage Guidelines.”</p>
<b>tos</b> <i>tos</i>	<p>(Optional) Packets can be filtered by type of service level, as specified by a number from 0 to 15, or by name as listed in the section “Usage Guidelines.”</p>

<b>log</b>	<p>(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.)</p> <p>The 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.</p> <p>The logging facility might drop some logging message packets if there are too many to be handled or if there is more than one logging message to be handled in 1 second. This behavior prevents the router from crashing due to too many logging packets. Therefore, the logging facility should not be used as a billing tool or an accurate source of the number of matches to an access list.</p>
<b>log-input</b>	(Optional) Includes the input interface and source MAC address or VC in the logging output.
<b>time-range</b> <i>time-range-name</i>	(Optional) Name of the time range that applies to this statement. The name of the time range and its restrictions are specified by the <b>time-range</b> command.
<i>operator</i>	<p>(Optional) Compares source or destination ports. Possible operands 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).</p> <p>If the operator is positioned after the <i>source</i> and <i>source-wildcard</i>, it must match the source port.</p> <p>If the operator is positioned after the <i>destination</i> and <i>destination-wildcard</i>, it must match the destination port.</p> <p>The <b>range</b> operator requires two port numbers. All other operators require one port number.</p>
<i>port</i>	<p>(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 section “Usage Guidelines.” TCP port names can only be used when filtering TCP. UDP port names can only be used when filtering UDP.</p> <p>TCP port names can only be used when filtering TCP. UDP port names can only be used when filtering UDP.</p>
<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.

**Defaults**

An extended access list defaults to a list that denies everything. An extended access list is terminated by an implicit deny statement.

**Command Modes**

Global configuration

Command History	Release	Modification
	10.0	This command was introduced.
	10.3	The following keywords and arguments were added: <ul style="list-style-type: none"> <li>• <i>source</i></li> <li>• <i>source-wildcard</i></li> <li>• <i>destination</i></li> <li>• <i>destination-wildcard</i></li> <li>• <b>precedence</b> <i>precedence</i></li> <li>• <i>icmp-type</i></li> <li>• <i>icm-code</i></li> <li>• <i>icmp-message</i></li> <li>• <i>igmp-type</i></li> <li>• <i>operator</i></li> <li>• <i>port</i></li> <li>• <b>established</b></li> </ul>
	11.1	The <b>dynamic</b> <i>dynamic-name</i> keyword and argument were added.
	11.1	The <b>timeout</b> <i>minutes</i> keyword and argument were added.
	11.2	The <b>log-input</b> keyword was added.
	12.0(1)T	The <b>time-range</b> <i>time-range-name</i> keyword and argument were added.
	12.0(11) and 12.1(2)	The <b>fragments</b> keyword was added.
	12.2(13)T	The <b>non500-isakmp</b> keyword was added to the list of UDP port names.

### Usage Guidelines

You can use access lists to control the transmission of packets on an interface, control vty access, and restrict the contents of routing updates. The Cisco IOS software stops checking the extended access list after a match occurs.

Fragmented IP packets, other than the initial fragment, are immediately accepted by any extended IP access list. Extended access lists used to control vty access or restrict the contents of routing updates must not match against the TCP source port, the type of service (ToS) value, or the precedence of the packet.



#### Note

After a numbered access list is created, any subsequent additions (possibly entered from the terminal) are placed at the end of the list. In other words, you cannot selectively add or remove access list command lines from a specific numbered access list.

The following is a list of precedence names:

- **critical**
- **flash**
- **flash-override**
- **immediate**
- **internet**



- **network**
- **priority**
- **routine**

The following is a list of UDP port names that can be used instead of port numbers. Refer to the current assigned numbers RFC to find a reference to these protocols. Port numbers corresponding to these protocols can also be found if you type a **?** in the place of a port number.

- **biff**
- **bootpc**
- **bootps**
- **discard**
- **dnsix**
- **domain**
- **echo**
- **mobile-ip**
- **nameserver**
- **netbios-dgm**
- **netbios-ns**
- **non500-isamkmp**
- **ntp**
- **rip**
- **snmp**
- **snmptrap**
- **sunrpc**
- **syslog**
- **tacacs-ds**
- **talk**
- **tftp**
- **time**
- **who**
- **xmcp**

#### **Access List Processing of Fragments**

The behavior of access-list entries regarding the use or lack 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,	<p>For an access-list entry containing only Layer 3 information:</p> <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets, initial fragments and noninitial fragments.</li> </ul> <p>For an access list entry containing Layer 3 and Layer 4 information:</p> <ul style="list-style-type: none"> <li>• The entry is applied to nonfragmented packets and initial fragments. <ul style="list-style-type: none"> <li>– If the entry is a <b>permit</b> statement, the packet or fragment is permitted.</li> <li>– If the entry is a <b>deny</b> statement, 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, the noninitial fragment is permitted.</li> <li>– If the entry is a <b>deny</b> statement, the next access-list entry is processed.</li> </ul> </li> </ul> <p> <b>Note</b> The <b>deny</b> statements are handled differently for noninitial fragments versus nonfragmented or initial fragments.</p>
...the <b>fragments</b> keyword, and assuming all of the access-list entry information matches,	<p>The access-list entry is applied only to noninitial fragments.</p> <p> <b>Note</b> The <b>fragments</b> keyword cannot be configured for an access-list entry that contains any Layer 4 information.</p>

Be aware that you should not simply 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 where 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 involving 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 had 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 was not policy routed or the reverse.

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 policy routing will occur as intended.

**Examples**

In the following example, serial interface 0 is part of a Class B network with the address 128.88.0.0, and the address of the mail host is 128.88.1.2. The **established** keyword is used only for the TCP protocol to indicate an established connection. A match occurs if the TCP datagram has the ACK or RST bits set, which indicates that the packet belongs to an existing connection.

```
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.0.0 0.0.255.255 established
access-list 102 permit tcp 0.0.0.0 255.255.255.255 128.88.1.2 0.0.0.0 eq 25
interface serial 0
 ip access-group 102 in
```

The following example permits Domain Naming System (DNS) packets and ICMP echo and echo reply packets:

```
access-list 102 permit tcp any 128.88.0.0 0.0.255.255 established
access-list 102 permit tcp any host 128.88.1.2 eq smtp
access-list 102 permit tcp any any eq domain
access-list 102 permit udp any any eq domain
access-list 102 permit icmp any any echo
access-list 102 permit icmp any any echo-reply
```

The following examples show how wildcard bits are used to indicate the bits of the prefix or mask that are relevant. Wildcard bits are similar to the bitmasks that are used with normal access lists. Prefix or mask bits corresponding to wildcard bits set to 1 are ignored during comparisons and prefix or mask bits corresponding to wildcard bits set to 0 are used in comparison.

The following example permits 192.108.0.0 255.255.0.0 but denies any more specific routes of 192.108.0.0 (including 192.108.0.0 255.255.255.0):

```
access-list 101 permit ip 192.108.0.0 0.0.0.0 255.255.0.0 0.0.0.0
access-list 101 deny ip 192.108.0.0 0.0.255.255 255.255.0.0 0.0.255.255
```

The following example permits 131.108.0/24 but denies 131.108/16 and all other subnets of 131.108.0.0:

```
access-list 101 permit ip 131.108.0.0 0.0.0.0 255.255.255.0 0.0.0.0
access-list 101 deny ip 131.108.0.0 0.0.255.255 255.255.0.0 0.0.255.255
```

The following example uses a time range to deny HTTP traffic on Monday through Friday from 8:00 a.m. to 6:00 p.m.:

```
time-range no-http
 periodic weekdays 8:00 to 18:00
 !
access-list 101 deny tcp any any eq http time-range no-http
 !
interface ethernet 0
 ip access-group 101 in
```

# show crypto ipsec sa

To display the settings used by current security associations (SAs), use the **show crypto ipsec sa** EXEC command.

**show crypto ipsec sa** [**map** *map-name* | **address** | **identity**] [**detail**]

Syntax Description		
<b>map</b> <i>map-name</i>	(Optional) Displays any existing security associations created for the crypto map set named <i>map-name</i> .	
<b>address</b>	(Optional) Displays the all existing security associations, sorted by the destination address (either the local address or the address of the IP Security remote peer) and then by protocol (Authentication Header or Encapsulation Security Protocol).	
<b>identity</b>	(Optional) Displays only the flow information. It does not show the security association information.	
<b>detail</b>	(Optional) Displays detailed error counters. (The default is the high level send/receive error counters.)	

**Command Modes** EXEC

Command History	Release	Modification
	11.3 T	This command was introduced.
	12.2(13)T	The “remote crypto endpt” and “ in use settings” fields were modified to support NAT traversal.

**Usage Guidelines** If no keyword is used, all SAs are displayed. They are sorted first by interface, and then by traffic flow (for example, source/destination address, mask, protocol, port). Within a flow, the SAs are listed by protocol (ESP/AH) and direction (inbound/outbound).

**Examples** The following is sample output for the **show crypto ipsec sa** command:

```
Router# show crypto ipsec sa

interface:FastEthernet0
  Crypto map tag:testtag, local addr. 10.2.80.161

  local ident (addr/mask/prot/port):(10.2.80.161/255.255.255.255/0/0)
  remote ident (addr/mask/prot/port):(100.0.0.1/255.255.255.255/0/0)
  current_peer:100.0.0.1:4500
    PERMIT, flags={origin_is_acl,}
    #pkts encaps:109, #pkts encrypt:109, #pkts digest 109
    #pkts decaps:109, #pkts decrypt:109, #pkts verify 109
    #pkts compressed:0, #pkts decompressed:0
    #pkts not compressed:0, #pkts compr. failed:0, #pkts decompress failed:0
    #send errors 90, #rcv errors 0
```

```
local crypto endpt.:10.2.80.161, remote crypto endpt.:100.0.0.1:4500
path mtu 1500, media mtu 1500
current outbound spi:23945537
```

```
inbound esp sas:
spi:0xF423E273(4095992435)
  transform:esp-des esp-sha-hmac ,
  in use settings ={Tunnel UDP-Encaps, }
  slot:0, conn id:200, flow_id:1, crypto map:testtag
  sa timing:remaining key lifetime (k/sec):(4607996/2546)
  IV size:8 bytes
  replay detection support:Y
```

```
inbound ah sas:
```

```
inbound pcp sas:
```

```
outbound esp sas:
spi:0x23945537(596923703)
  transform:esp-des esp-sha-hmac ,
  in use settings ={Tunnel UDP-Encaps, }
  slot:0, conn id:201, flow_id:2, crypto map:testtag
  sa timing:remaining key lifetime (k/sec):(4607998/2519)
  IV size:8 bytes
  replay detection support:Y
```

```
outbound ah sas:
```

```
outbound pcp sas:
```

# Glossary

**IKE**—Internet Key Exchange. Hybrid protocol that implements Oakley key exchange and Skeme key exchange inside the Internet Security Association Key Management Protocol (ISAKMP) framework. Although IKE can be used with other protocols, its initial implementation is with IPSec. IKE provides authentication of the IPSec peers, negotiates IPSec keys, and negotiates IPSec security associations (SAs).

**IPSec**—IP Security. Framework of open standards developed by the Internet Engineering Task Force (IETF). IPSec provides security for transmission of sensitive information over unprotected networks such as the Internet. IPSec acts at the network layer, protecting and authenticating IP packets between participating IPSec devices (“peers”), such as Cisco routers.

**NAT**—Network Address Translation. Translates a private IP address used inside the corporation to a public, routable address for use on the outside of the corporation, such as the Internet. NAT is considered a one-to-one mapping of addresses from private to public.

**PAT**—Port Address Translation. Like NAT, PAT also translated private IP address to public, routable addresses. Unlike NAT, PAT provides a many-to-one mapping of private addresses to a public address; each instance of the public address is associated with a particular port number to provide uniqueness. PAT can be used in environments where the cost of obtaining a range of public addresses is too expensive for an organization.

**Note**

---

Refer to the [Internetworking Terms and Acronyms](#) for terms not included in this glossary.

---



