crypto aaa attribute list through crypto ipsec transform-set

- crypto aaa attribute list, on page 3
- crypto ca authenticate, on page 6
- crypto ca cert validate, on page 8
- crypto ca certificate chain, on page 9
- crypto ca certificate map, on page 11
- crypto ca certificate query (ca-trustpoint), on page 14
- crypto ca certificate query (global), on page 16
- crypto ca crl request, on page 17
- crypto ca enroll, on page 19
- crypto ca export pem, on page 22
- crypto ca export pkcs12, on page 25
- crypto ca identity, on page 27
- crypto ca import, on page 28
- crypto ca import pem, on page 29
- crypto ca import pkcs12, on page 31
- crypto ca profile enrollment, on page 33
- crypto ca trusted-root, on page 35
- crypto ca trustpoint, on page 36
- crypto call admission limit, on page 38
- crypto connect vlan, on page 40
- crypto ctcp, on page 42
- crypto dynamic-map, on page 44
- crypto-engine, on page 47
- crypto engine accelerator, on page 48
- crypto engine aim, on page 51
- crypto engine compliance shield disable, on page 52
- crypto engine em, on page 53
- crypto engine mode vrf, on page 54
- crypto engine nm, on page 56
- crypto engine onboard, on page 57
- crypto engine slot, on page 58

- crypto engine slot (interface), on page 59
- crypto gdoi ks, on page 62
- crypto gdoi gm, on page 64
- crypto gdoi group, on page 66
- crypto identity, on page 67
- crypto ikev2 authorization policy, on page 69
- crypto ikev2 certificate-cache, on page 71
- crypto ikev2 cluster, on page 72
- crypto ikev2 cookie-challenge, on page 74
- crypto ikev2 cts, on page 75
- crypto ikev2 diagnose, on page 80
- crypto ikev2 dpd, on page 81
- crypto ikev2 fragmentation, on page 83
- crypto ikev2 http-url, on page 84
- crypto ikev2 keyring, on page 85
- crypto ikev2 limit, on page 88
- crypto ikev2 name mangler, on page 90
- crypto ikev2 nat, on page 92
- crypto ikev2 policy, on page 93
- crypto ikev2 profile, on page 96
- crypto ikev2 proposal, on page 100
- crypto ikev2 redirect, on page 103
- crypto ikev2 window, on page 104
- crypto ipsec client ezvpn (global), on page 105
- crypto ipsec client ezvpn (interface), on page 110
- crypto ipsec client ezvpn connect, on page 113
- crypto ipsec client ezvpn xauth, on page 114
- crypto ipsec transform-set default, on page 116
- crypto ipsec df-bit (global), on page 118
- crypto ipsec df-bit (interface), on page 119
- crypto ipsec fragmentation (global), on page 121
- crypto ipsec fragmentation (interface), on page 122
- crypto ipsec ike sa-strength-enforcement, on page 124
- crypto ipsec ipv4-deny, on page 126
- crypto ipsec nat-transparency, on page 128
- crypto ipsec optional, on page 130
- crypto ipsec optional retry, on page 131
- crypto ipsec profile, on page 132
- crypto ipsec security-association dummy, on page 134
- crypto ipsec security-association idle-time, on page 135
- crypto ipsec security-association lifetime, on page 137
- crypto ipsec security-association multi-sn, on page 140
- crypto ipsec security-association replay disable, on page 141
- crypto ipsec security-association replay window-size, on page 142
- crypto ipsec server send-update, on page 143
- crypto ipsec transform-set, on page 144

crypto aaa attribute list

To define an authentication, authorization, and accounting (AAA) attribute list of per-user attributes on a local Easy VPN server, use the **crypto aaa attribute list**command in crypto isakmp group configuration mode. To remove the AAA attribute list, use the **no** form of this command.

crypto aaa attribute list *list-name*
no crypto aaa attribute list *list-name*

Syntax Description

list-name	Name of the local attribute list.

Command Default

A local attribute list is not defined.

Command Modes

Crypto isakmp group configuration

Command History

Release	Modification
12.4(9)T	This command was introduced.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

There is no limit to the number of lists that can be defined (except for NVRAM storage limits).

Examples

The following example shows that per-user attributes have been defined on a local Easy VPN AAA server:

```
!
aaa new-model
!
!
aaa authentication login default local
aaa authentication login noAAA none
aaa authorization network default local
!
aaa attribute list per-group
 attribute type inacl "per-group-acl" service ike protocol ip mandatory
!
aaa session-id common
!
resource policy
!
ip subnet-zero
!
!
ip cef
```

```
!
!
username example password 0 example
!
!
crypto isakmp policy 3
 encr aes
 authentication pre-share
 group 14
crypto isakmp xauth timeout 90
!
crypto isakmp client configuration group PerUserAAA
 key cisco
 pool dpool
 crypto aaa attribute list per-group
!
crypto isakmp profile vi
 match identity group PerUserAAA
 isakmp authorization list default
 client configuration address respond
 client configuration group PerUserAAA
 virtual-template 1
!
!
crypto ipsec transform-set set esp-aes esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set isakmp-profile vi
!
!
interface GigabitEthernet0/0
 description 'EzVPN Peer'
 ip address 192.168.1.1 255.255.255.128
 duplex full
 speed 100
 media-type rj45
 no negotiation auto
!
interface GigabitEthernet0/1
 no ip address
 shutdown
 duplex auto
 speed auto
 media-type rj45
 no negotiation auto
interface Virtual-Template1 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
!
ip local pool dpool 10.5.0.1 10.5.0.10
ip classless
!
no ip http server
no ip http secure-server
!
!
ip access-list extended per-group-acl
 permit tcp any any
 deny   icmp any any
logging alarm informational
logging trap debugging
!
```

```
control-plane
!
gatekeeper
 shutdown
!
line con 0
line aux 0
 stopbits 1
line vty 0 4
!
!
end
```

Related Commands

Command	Description
crypto isakmp client configuration group	Specifies to which group a policy profile will be defined.

crypto ca authenticate

Note This command was replaced by the **crypto pki authenticate** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To authenticate the certification authority (by getting the certificate of the CA), use the **crypto ca authenticate** command in global configuration mode.

crypto ca authenticate *name*

Syntax Description

name	Specifies the name of the CA. This is the same name used when the CA was declared with the **crypto ca identity** command .

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command is required when you initially configure CA support at your router.

This command authenticates the CA to your router by obtaining the self-signed certificate of the CA that contains the public key of the CA. Because the CA signs its own certificate, you should manually authenticate the public key of the CA by contacting the CA administrator when you perform this command.

If you are using RA mode (using the **enrollment mode ra** command) when you issue the **crypto ca authenticate** command, then registration authority signing and encryption certificates will be returned from the CA as well as the CA certificate.

This command is not saved to the router configuration. However. the public keys embedded in the received CA (and RA) certificates are saved to the configuration as part of the RSA public key record (called the “RSA public key chain”).

If the CA does not respond by a timeout period after this command is issued, the terminal control will be returned so it will not be tied up. If this happens, you must re-enter the command. Cisco IOS software will not recognize CA certificate expiration dates set for beyond the year 2038. If the validity period of the CA certificate is set to expire after the year 2038, the following error message will be displayed when authentication with the CA server is attempted:

error retrieving certificate :incomplete chain

The following messages are displayed when you attempt to debug the error:

CRYPTO_PKI: Unable to read CA/RA certificates.

PKI-3-GETCARACERT Failed to receive RA/CA certificates.

If you receive an error message similar to this one, check the expiration date of your CA certificate. If the expiration date of your CA certificate is set after the year 2038, you must reduce the expiration date by a year or more.

Examples

In the following example, the router requests the certificate of the CA. The CA sends its certificate and the router prompts the administrator to verify the certificate of the CA by checking the CA certificate's fingerprint. The CA administrator can also view the CA certificate's fingerprint, so you should compare what the CA administrator sees to what the router displays on the screen. If the fingerprint on the router's screen matches the fingerprint viewed by the CA administrator, you should accept the certificate as valid.

```
Router(config)#
crypto ca authenticate myca
Certificate has the following attributes:
Fingerprint: 0123 4567 89AB CDEF 0123
Do you accept this certificate? [yes/no] y
#
```

Related Commands

Command	Description
debug crypto pki transactions	Displays debug messages for the trace of interaction (message type) between the CA and the router.
show crypto pki certificates	Displays information about your certificate, the certificate of the CA, and any RA certificates.

crypto ca cert validate

Note This command was replaced by the **crypto pki cert validate** command effective with Cisco IOS Release 12.3(8)T and 12.2(18)SXE.

To determine if a trustpoint has been successfully authenticated, a certificate has been requested and granted, and if the certificate is currently valid, use the **crypto ca cert validate** command in global configuration mode.

crypto ca cert validate *trustpoint*

Syntax Description

trustpoint	The trustpoint to be validated.

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
12.3(8)T	This command was introduced.

Usage Guidelines

The **crypto ca cert validate** command validates the router's own certificate for a given trustpoint. Use this command as a sanity check after enrollment to verify that the trustpoint is properly authenticated, a certificate has been requested and granted for the trustpoint, and that the certificate is currently valid. A certificate is valid if it is signed by the trustpoint certification authority (CA), not expired, and so on.

Examples

The following examples show the possible output from the **crypto ca cert validate** command:

```
Router(config)# crypto ca cert validate ka
Validation Failed: trustpoint not found for ka
Router(config)# crypto ca cert validate ka
Validation Failed: can't get local certificate chain
Router(config)# crypto ca cert validate ka
Certificate chain has 2 certificates.
Certificate chain for ka is valid
Router(config)# crypto ca cert validate ka
Certificate chain has 2 certificates.
Validation Error: no certs on chain
Router(config)# crypto ca cert validate ka
Certificate chain has 2 certificates.
Validation Error: unspecified error
```

Related Commands

Command	Description
crypto pki trustpoint	Declares the certification authority that the router should use.
show crypto pki trustpoints	Displays the trustpoints that are configured in the router.

crypto ca certificate chain

Note This command was replaced by the **crypto pki certificate chain** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To enter the certificate chain configuration mode, use the **crypto ca certificate chain**command in global configuration mode. (You need to be in certificate chain configuration mode to delete certificates.)

crypto ca certificate chain *name*

Syntax Description

name	Specifies the name of the CA. Use the same name as when you declared the CA using the **crypto pki trustpoint**command.

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command puts you into certificate chain configuration mode. When you are in certificate chain configuration mode, you can delete certificates using the **certificate** command.

Examples

The following example deletes the router’s certificate. In this example, the router had a general-purpose RSA key pair with one corresponding certificate. The show command is used to determine the serial number of the certificate to be deleted.

```
Router# show crypto ca certificates
Certificate
  Subject Name
    Name: myrouter.example.com
    IP Address: 10.0.0.1
  Status: Available
  Certificate Serial Number: 0123456789ABCDEF0123456789ABCDEF
  Key Usage: General Purpose

CA Certificate
  Status: Available
  Certificate Serial Number: 3051DF7123BEE31B8341DFE4B3A338E5F
  Key Usage: Not Set
Router# configure terminal
Rrouter(config)# crypto ca certificate chain myca
Router(config-cert-chain)# no certificate 0123456789ABCDEF0123456789ABCDEF
% Are you sure you want to remove the certificate [yes/no]? yes
% Be sure to ask the CA administrator to revoke this certificate.
Router(config-cert-chain)# exit
```

Related Commands

Command	Description
certificate	Adds certificates manually.

crypto ca certificate map

Note This command was replaced by the **crypto pki certificate map** command effective with Cisco IOS Release 12.3(7)T, 12.2(18)SXD, and 12.2(18)SXE.

To define certificate-based access control lists (ACLs), use the **crypto ca certificate map** command in ca-certificate-map configuration mode. To remove the certificate-based ACLs, use the no form of this command.

crypto ca certificate map *label sequence-number*
no crypto ca certificate map *label sequence-number*

Syntax Description

label	A user-specified label that is referenced within the **crypto ca trustpoint** command.
sequence-number	A number that orders the ACLs with the same label. ACLs with the same label are processed from lowest to highest sequence number. When an ACL is matched, processing stops with a successful result.

Command Default

No default behavior or value.

Command Modes

Ca-certificate-map configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

Issuing this command places the router in CA certificate map configuration mode where you can specify several certificate fields together with their matching criteria. The general form of these fields is as follows:

```
field-name match-criteria match-value
```

The *field-name* in the above example is one of the certificate fields. Field names are similar to the names used in the International Telecommunication Union Telecommunication Standardization Sector (ITU-T) X.509 standard. The **name**field is a special field that matches any subject name or related name field in the certificate, such as the **alt-subject-name, subject-name**, and **unstructured-subject-name** fields.

- **alt-subject-name** -- Case-insensitive string.
- **expires-on** --Date field in the format dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss.
- **issuer-name** -- Case-insensitive string.
- **name** -- Case-insensitive string.
- **subject-name** --Case-insensitive string.
- **unstructured-subject-name** -- Case-insensitive string.
- **valid-start** --Date field in the format dd mm yyyy hh:mm:ss or mmm dd yyyy hh:mm:ss.

Note The time portion is optional in both the **expires-on** date and **valid-start** field and defaults to 00:00:00 if not specified. The time is interpreted according to the time zone offset configured for the router. The string **utc** can be appended to the date and time when they are configured as Universal Time, Coordinated (UTC) rather than local time.

The *match-criteria* in the example is one of the following logical operators:

- **eq** --equal (valid for name and date fields)
- **ne** --not equal (valid for name and date fields)
- **co** --contains (valid only for name fields)
- **nc** --does not contain (valid only for name fields)
- **lt** --less than (valid only for date fields)
- **ge** --greater than or equal to (valid only for date fields)

The *match-value* is a case-insensitive string or a date.

Examples

The following example shows how to configure a certificate-based ACL that will allow any certificate issued by Cisco Systems to an entity within the cisco.com domain. The label is Cisco, and the sequence is 10.

```
crypto ca certificate map Cisco 10
 issuer-name co Cisco Systems
 unstructured-subject-name co cisco.com
```

The following example accepts any certificate issued by Cisco Systems for an entity with DIAL or organizationUnit component ou=WAN. This certificate-based ACL consists of two separate ACLs tied together with the common label Group. Because the check for DIAL has a lower sequence number, it is performed first. Note that the string “DIAL” can occur anywhere in the subjectName field of the certificate, but the string WAN must be in the organizationUnit component.

```
crypto ca certificate map Group 10
 issuer-name co Cisco Systems
 subject-name co DIAL
crypto ca certificate map Group 20
 issuer-name co Cisco Systems
 subject-name co ou=WAN
```

Case is ignored in string comparisons; therefore, DIAL in the previous example will match dial, DIAL, Dial, and so on. Also note that the component identifiers (o=, ou=, cn=, and so on) are not required unless it is desirable that the string to be matched occurs in a specific component of the name. (Refer to the ITU-T security standards for more information about certificate fields and components such as ou=.)

If a component identifier is specified in the match string, the exact string, including the component identifier, must appear in the certificate. This requirement can present a problem if more than one component identifier is included in the match string. For example, “ou=WAN,o=Cisco Systems” will not match a certificate with the string “ou=WAN,ou=Engineering,o=Cisco Systems” because the “ou=Engineering” string separates the two desired component identifiers.

To match both “ou=WAN” and “o=Cisco Systems” in a certificate while ignoring other component identifiers, you could use this certificate map:

```
crypto ca certificate map Group 10
 subject-name co ou=WAN
 subject-name co o=Cisco
```

Any space character proceeding or following the equal sign (=) character in component identifiers is ignored. Therefore “o=Cisco” in the proceeding example will match “o = Cisco,” “o= Cisco,” “o =Cisco,” and so on.

Related Commands

Command	Description
crypto pki trustpoint	Declares the CA that your router should use.

crypto ca certificate query (ca-trustpoint)

Note This command was replaced by the **crypto pki certificate query (ca-trustpoint)**command effective with Cisco IOS Release 12.3(7)T, 12.2(18)SXD, and 12.2(18)SXE.

To specify that certificates should not be stored locally but retrieved from a certification authority (CA) trustpoint, use the **crypto ca certificate query** command in ca-trustpoint configuration mode. To cause certificates to be stored locally per trustpoint, use the no form of this command.

crypto ca certificate query
no crypto ca certificate query

Syntax Description

This command has no arguments or keywords.

Command Default

CA trustpoints are stored locally in the router's NVRAM.

Command Modes

Ca-trustpoint configuration

Command History

Release	Modification
12.2(8)T	This command was introduced.

Usage Guidelines

Normally, certain certificates are stored locally in the router's NVRAM, and each certificate uses a moderate amount of memory. To save NVRAM space, you can use this command to put the router into query mode, preventing certificates from being stored locally; instead, they are retrieved from a specified CA trustpoint when needed. This will save NVRAM space but could result in a slight performance impact.

The **crypto ca certificate query** command is a subcommand for each trustpoint; thus, this command can be disabled on a per-trustpoint basis.

Before you can configure this command, you must enable the **crypto pki trustpoint**command , which puts you in ca-trustpoint configuration mode.

Note This command replaces the **crypto ca certificate query**command in global configuration mode. Although you can still enter the global configuration command, the configuration mode and command will be written back as ca-trustpoint.

Examples

The following example shows how to prevent certificates and certificate revocation lists (CRLs) from being stored locally on the router; instead, they are retrieved from the "ka" trustpoint when needed.

```
crypto ca trustpoint ka
 .
 .
```

```
.
crypto ca certificate query
```

Related Commands

Command	Description
crypto pki trustpoint	Declares the CA that your router should use.

crypto ca certificate query (global)

The **crypto ca certificate query**command in global configuration mode is replaced by the crypto ca certificate query command in ca-trustpoint configuration mode. See the **crypto ca certificate query** command for more information.

crypto ca crl request

Note Effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE, this command was replaced by the **crypto pki crl request** command.

To request that a new certificate revocation list (CRL) be obtained immediately from the certification authority, use the **crypto ca crl request**command in global configuration mode.

crypto ca crl request *name*

Syntax Description

name	Specifies the name of the CA. This is the same name used when the CA was declared with the **crypto pki trustpoint**command.

Command Default

Normally, the router requests a new CRL when it is verifying a certificate and there is no CRL cached.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.
12.3(7)T	This command was replaced by the **crypto pki crl request** command.

Usage Guidelines

A CRL lists all the certificates of the network device that have been revoked. Revoked certificates will not be honored by your router; therefore, any IPSec device with a revoked certificate cannot exchange IP Security traffic with your router.

The first time your router receives a certificate from a peer, it will download a CRL from the CA. Your router then checks the CRL to make sure the certificate of the peer has not been revoked. (If the certificate appears on the CRL, it will not accept the certificate and will not authenticate the peer.)

A CRL can be reused with subsequent certificates until the CRL expires. If your router receives the certificate of a peer after the applicable CRL has expired, it will download the new CRL.

If your router has a CRL which has not yet expired, but you suspect that the contents of the CRL are out of date, use the **crypto ca crl request** command to request that the latest CRL be immediately downloaded to replace the old CRL.

This command is not saved to the configuration.

Note This command should be used only after the trustpoint is enrolled.

Examples

The following example immediately downloads the latest CRL to your router:

```
crypto ca crl request
```

crypto ca enroll

Note This command was replaced by the **crypto pki enroll** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To obtain the certificate(s) of your router from the certification authority, use the **crypto ca enroll** command in global configuration mode. To delete a current enrollment request, use the **no** form of this command.

crypto ca enroll *name*
no crypto ca enroll *name*

Syntax Description

name	Specifies the name of the CA. Use the same name as when you declared the CA using the **crypto pki trustpoint**command.

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.

Usage Guidelines

This command requests certificates from the CA for all of your router's RSA key pairs. This task is also known as enrolling with the CA. (Technically, enrolling and obtaining certificates are two separate events, but they both occur when this command is issued.)

Your router needs a signed certificate from the CA for each RSA key pairs of your router; if you previously generated general purpose keys, this command will obtain the one certificate corresponding to the one general purpose RSA key pair. If you previously generated special usage keys, this command will obtain two certificates corresponding to each of the special usage RSA key pairs.

If you already have a certificate for your keys you will be unable to complete this command; instead, you will be prompted to remove the existing certificate first. (You can remove existing certificates with the **no certificate** command.)

The **crypto ca enroll** command is not saved in the router configuration.

Note If your router reboots after you issue the **crypto ca enroll** command but before you receive the certificate(s), you must reissue the command.

Responding to Prompts

When you issue the **crypto ca enroll** command, you are prompted a number of times.

First, you are prompted to create a challenge password. This password can be up to 80 characters in length. This password is necessary in the event that you ever need to revoke your router’s certificate(s). When you ask the CA administrator to revoke your certificate, you must supply this challenge password as a protection against fraudulent or mistaken revocation requests.

Note This password is not stored anywhere, so you need to remember this password.

If you lose the password, the CA administrator may still be able to revoke the router’s certificate but will require further manual authentication of the router administrator identity.

You are also prompted to indicate whether or not your router’s serial number should be included in the obtained certificate. The serial number is not used by IP Security or Internet Key Exchange but may be used by the CA to either authenticate certificates or to later associate a certificate with a particular router. (Note that the serial number stored is the serial number of the internal board, not the one on the enclosure.) Ask your CA administrator if serial numbers should be included. If you are in doubt, include the serial number.

Normally, you would not include the IP address because the IP address binds the certificate more tightly to a specific entity. Also, if the router is moved, you would need to issue a new certificate. Finally, a router has multiple IP addresses, any of which might be used with IPSec.

If you indicate that the IP address should be included, you will then be prompted to specify the interface of the IP address. This interface should correspond to the interface that you apply your crypto map set to. If you apply crypto map sets to more than one interface, specify the interface that you name in the **crypto map local-address** command.

Examples

In the following example, a router with a general-purpose RSA key pair requests a certificate from the CA. When the router displays the certificate fingerprint, the administrator verifies this number by calling the CA administrator, who checks the number. The fingerprint is correct, so the router administrator accepts the certificate.

There can be a delay between when the router administrator sends the request and when the certificate is actually received by the router. The amount of delay depends on the CA method of operation.

```
Router(config)# crypto ca enroll myca
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this
   password to the CA Administrator in order to revoke your certificate.
   For security reasons your password will not be saved in the configuration.
   Please make a note of it.
Password: <mypassword>
Re-enter password: <mypassword>
% The subject name in the certificate will be: myrouter.example.com
% Include the router serial number in the subject name? [yes/no]: yes
% The serial number in the certificate will be: 03433678
% Include an IP address in the subject name [yes/no]? yes
Interface: ethernet0/0
Request certificate from CA [yes/no]? yes
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The ’show crypto pki certificates’ command will also show the fingerprint.
```

Some time later, the router receives the certificate from the CA and displays the following confirmation message:

```
Router(config)#   Fingerprint: 01234567 89ABCDEF FEDCBA98 75543210
%CRYPTO-6-CERTRET: Certificate received from Certificate Authority
Router(config)#
```

If necessary, the router administrator can verify the displayed Fingerprint with the CA administrator.

If there is a problem with the certificate request and the certificate is not granted, the following message is displayed on the console instead:

```
%CRYPTO-6-CERTREJ: Certificate enrollment request was rejected by Certificate Authority
```

The subject name in the certificate is automatically assigned to be the same as the RSA key pair's name. In the above example, the RSA key pair was named "myrouter.example.com." (The router assigned this name.)

Requesting certificates for a router with special usage keys would be the same as the previous example, except that two certificates would have been returned by the CA. When the router received the two certificates, the router would have displayed the same confirmation message:

```
%CRYPTO-6-CERTRET: Certificate received from Certificate Authority
```

Related Commands

Command	Description
debug crypto pki messages	Displays debug messages for the details of the interaction (message dump) between the CA and the router.
debug crypto pki transactions	Displays debug messages for the trace of interaction (message type) between the CA and the router.
show crypto pki certificates	Displays information about your certificate, the certificate of the CA, and any RA certificates.

crypto ca export pem

Note This command was replaced by the **crypto pki export pem** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To export certificates and Rivest, Shamir, and Adelman (RSA) keys that are associated with a trustpoint in a privacy-enhanced mail (PEM)-formatted file, use the **crypto ca export pem**command in global configuration mode.

crypto ca export *trustpoint* **pem** {**terminal** | **url** *url*} {**3des** | **des**} *passphrase*

Syntax Description

trustpoint	Name of the trustpoint that the associated certificate and RSA key pair will export. The *trustpoint* argument must match the name that was specified via the **crypto pki trustpoint** command.
terminal	Certificate and RSA key pair that will be displayed in PEM format on the console terminal.
url *url*	URL of the file system where your router should export the certificate and RSA key pairs.
3des	Export the trustpoint using the Triple Data Encryption Standard (3DES) encryption algorithm.
des	Export the trustpoint using the DES encryption algorithm.
passphrase	Passphrase that is used to encrypt the PEM file for import. **Note** The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.3(4)T	This command was introduced.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The **crypto ca export pem**command allows you to export certificate and RSA key pairs in PEM-formatted files. The PEM files can then be imported back into the Cisco IOS router (via the **crypto pki import pem** command) or other public key infrastructure (PKI) applications.

Examples

The following example shows how to generate and export the RSA key pair “aaa” and certificates of the router in PEM files that are associated with the trustpoint “mycs”:

```
Router(config)# crypto key generate rsa general-keys label aaa exportable

The name for the keys will be:aaa
Choose the size of the key modulus in the range of 360 to 2048 for your General Purpose
Keys. Choosing a key modulus greater than 512 may take a few minutes.
!
How many bits in the modulus [512]: 2048
% Generating 2048 bit RSA keys ...[OK]
!
Router(config)# crypto pki trustpoint mycs
Router(ca-trustpoint)# enrollment url http://mycs
Router(ca-trustpoint)# rsakeypair aaa
Router(ca-trustpoint)# exit
Router(config)# crypto pki authenticate mycs
Certificate has the following attributes:
Fingerprint:C21514AC 12815946 09F635ED FBB6CF31
% Do you accept this certificate? [yes/no]:y
Trustpoint CA certificate accepted.
!
Router(config)# crypto pki enroll mycs
%
% Start certificate enrollment ..
% Create a challenge password. You will need to verbally provide this password to the CA
Administrator in order to revoke your certificate.
For security reasons your password will not be saved in the configuration.
Please make a note of it.

Password:
Re-enter password:

% The fully-qualified domain name in the certificate will be:Router
% The subject name in the certificate will be:bizarro.cisco.com
% Include the router serial number in the subject name? [yes/no]:n
% Include an IP address in the subject name? [no]:n
Request certificate from CA? [yes/no]:y
% Certificate request sent to Certificate Authority
% The certificate request fingerprint will be displayed.
% The 'show crypto ca certificate' command will also show the fingerprint.

Router(config)# Fingerprint: 8DA777BC 08477073 A5BE2403 812DD157

00:29:11:%CRYPTO-6-CERTRET:Certificate received from Certificate Authority

Router(config)# crypto ca export aaa pem terminal 3des cisco123
% CA certificate:
-----BEGIN CERTIFICATE-----
MIICAzCCAa2gAwIBAgIBATANBgkqhkiG9w0BAQUFADBOMQswCQYDVQQGEwJVUzES
<snip>
waDeNOSI3WlDa0AWq5DkVBkxwgn0TqIJXJOCttjHnWHK1LMcMVGn
-----END CERTIFICATE-----

% Key name:aaa
    Usage:General Purpose Key
-----BEGIN RSA PRIVATE KEY-----
```

```
Proc-Type:4,ENCRYPTED
DEK-Info:DES-EDE3-CBC,ED6B210B626BC81A

Urguv0jnjwOgowWVUQ2XR5nbzzYHI2vGLunpH/IxIsJuNjRVjbAAUpGk7VnPCT87
<snip>
kLCOtxzEv7JHc72gMku9uUlrLSnFH5slzAtoC0czfU4=
-----END RSA PRIVATE KEY-----

% Certificate:
-----BEGIN CERTIFICATE-----
MIICTjCCAfigAwIBAgICIQUwDQYJKoZIhvcNAQEFBQAwTjELMAkGA1UEBhMCVVMx
<snip>
6xlBaIsuMxnHmr89KkKkYlU6
-----END CERTIFICATE-----
```

Related Commands

Command	Description
crypto pki import pem	Imports certificates and RSA keys to a trustpoint from PEM-formatted files.
crypto pki trustpoint	Declares the CA that your router should use.
enrollment	Specifies the enrollment parameters of a CA.

crypto ca export pkcs12

Note This command was replaced by the **crypto pki export pkcs12** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To export Rivest, Shamir, and Adelman (RSA) keys within a PKCS12 file at a specified location, use the **crypto ca export pkcs12** command in global configuration mode.

crypto ca export *trustpointname* **pkcs12** *destination url passphrase*

Syntax Description

trustpointname	Name of the trustpoint who issues the certificate that a user is going to export. When you export the PKCS12 file, the trustpoint name is the RSA key name.
destination url	Location of the PKCS12 file to which a user wants to import the RSA key pair.
passphrase	Passphrase that is used to encrypt the PKCS12 file for export.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

The **crypto ca export pkcs12**command creates a PKCS 12 file that contains an RSA key pair. The PKCS12 file, along with a certificate authority (CA), is exported to the location that you specify with the destination URL. If you decide not to import the file to another router, you must delete the file.

Security Measures

Keep the PKCS12 file stored in a secure place with restricted access.

An RSA keypair is more secure than a passphrase because the private key in the key pair is not known by multiple parties. When you export an RSA key pair to a PKCS#12 file, the RSA key pair now is only as secure as the passphrase.

To create a good passphrase, be sure to include numbers, as well as both lowercase and uppercase letters. Avoid publicizing the passphrase by mentioning it in e-mail or cell phone communications because the information could be accessed by an unauthorized user.

Examples

The following example exports an RSA key pair with a trustpoint name “mytp” to a Flash file:

```
Router(config)# crypto ca export mytp pkcs12 flash:myexport mycompany
```

Related Commands

Command	Description
crypto pki import pkcs12	Imports RSA keys.

crypto ca identity

The **crypto ca identity** command is replaced by the crypto ca trustpoint command. See the crypto ca trustpointcommand for more information.

crypto ca import

Note This command was replaced by the **crypto pki import** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXD.

To import a certificate manually via TFTP or as a cut-and-paste at the terminal, use the **crypto ca import**command in global configuration mode.

crypto ca import *name* **certificate**

Syntax Description

name **certificate**	Name of the certification authority (CA). This name is the same name used when the CA was declared with the **crypto pki trustpoint** command.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

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

Usage Guidelines

You must enter the **crypto ca import** command twice if usage keys (signature and encryption keys) are used. The first time the command is entered, one of the certificates is pasted into the router; the second time the command is entered, the other certificate is pasted into the router. (It does not matter which certificate is pasted first.)

Examples

The following example shows how to import a certificate via cut-and-paste. In this example, the CA trustpoint is “MS.”

```
crypto pki trustpoint MS
 enroll terminal
 crypto pki authenticate MS
!
crypto pki enroll MS
crypto ca import MS certificate
```

Related Commands

Command	Description
crypto pki trustpoint	Declares the CA that your router should use.
enrollment	Specifies the enrollment parameters of your CA.
enrollment terminal	Specifies manual cut-and-paste certificate enrollment.

crypto ca import pem

Note This command was replaced by the **crypto pki import pem** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To import certificates and Rivest, Shamir, and Adelman (RSA) keys to a trustpoint from privacy-enhanced mail (PEM)-formatted files, use the **crypto ca import pem** command in global configuration mode.

crypto ca import *trustpoint* **pem** [**usage-keys**] {**terminal** | **url** *url*} [**exportable**] *passphrase*

Syntax Description

trustpoint	Name of the trustpoint that is associated with the imported certificates and RSA key pairs. The *trustpoint* argument must match the name that was specified via the **crypto pki trustpoint** command.
usage-keys	(Optional) Specifies that two RSA special usage key pairs will be imported (that is, one encryption pair and one signature pair), instead of one general-purpose key pair.
terminal	Certificates and RSA key pairs will be manually imported from the console terminal.
url *url*	URL of the file system where your router should import the certificates and RSA key pairs.
exportable	(Optional) Specifies that the imported RSA key pair can be exported again to another Cisco device such as a router.
passphrase	Passphrase that is used to encrypt the PEM file for import. **Note** The passphrase can be any phrase that is at least eight characters in length; it can include spaces and punctuation, excluding the question mark (?), which has special meaning to the Cisco IOS parser.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.3(4)T	This command was introduced.

Usage Guidelines

The **crypto ca import pem**command allows you import certificates and RSA key pairs in PEM-formatted files.The files can be previously exported from another router or generated from other public key infrastructure (PKI) applications.

Examples

The following example shows how to import PEM files to trustpoint “ggg” via TFTP:

```
Router(config)# crypto ca import ggg pem url tftp://10.1.1.2/johndoe/msca cisco1234
```

```
% Importing CA certificate...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.ca]?
Reading file from tftp://10.1.1.2/johndoe/msca.ca
Loading johndoe/msca.ca from 10.1.1.2 (via Ethernet0):!
[OK - 1082 bytes]
% Importing private key PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.prv]?
Reading file from tftp://10.1.1.2/johndoe/msca.prv
Loading johndoe/msca.prv from 10.1.1.2 (via Ethernet0):!
[OK - 573 bytes]
% Importing  certificate PEM file...
Address or name of remote host [10.1.1.2]?
Destination filename [johndoe/msca.crt]?
Reading file from tftp://10.1.1.2/johndoe/msca.crt
Loading johndoe/msca.crt from 10.1.1.2 (via Ethernet0):!
[OK - 1289 bytes]
% PEM files import succeeded.
Router(config)#
```

Related Commands

Command	Description
crypto pki export pem	Exports certificates and RSA keys that are associated with a trustpoint in a PEM-formatted file.
crypto pki trustpoint	Declares the CA that your router should use.
enrollment	Specifies the enrollment parameters of a CA.

crypto ca import pkcs12

Note This command was replaced by the **crypto pki import pkcs12** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To import Rivest, Shamir, and Adelman (RSA) keys, use the **crypto ca import pkcs12** command in global configuration mode.

crypto ca import *trustpointname* **pkcs12** *source url passphrase*

Syntax Description

trustpointname	Name of the trustpoint who issues the certificate that a user is going to export or import. When importing, the trustpoint name will become the RSA key name.
source url	The location of the PKCS12 file to which a user wants to export the RSA key pair.
passphrase	Passphrase that must be entered to undo encryption when the RSA keys are imported.

Command Default

No default behavior or values

Command Modes

Global configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.

Usage Guidelines

When you enter the **cyrpto ca import pkcs12** command, a ke pair and a trustpoint are generated. If you then decide you want to remove the key pair and trustpoint that were generated, enter the **crypto key zeroize rsa** command to zeroize the key pair and enter the **no crypto ca trustpoint** command to remove the trustpoint.

Note After you import RSA keys to a target router, you cannot export those keys from the target router to another router.

Examples

In the following example, an RSA key pair that has been associated with the trustpoint “forward” is to be imported:

```
Router(config)# crypto ca import forward pkcs12 flash:myexport mycompany
```

Related Commands

Command	Description
crypto pki export pkcs12	Exports RSA keys.
crypto pki trustpoint	Declares the CA that your router should use.

Command	Description
crypto key zeroize rsa	Deletes all RSA keys from your router.

crypto ca profile enrollment

Note This command was replaced with the **crypto pki profile enrollment** command effective with Cisco IOS Release 12.3(7)T and 12.2(18)SXE.

To define an enrollment profile, use the **crypto ca profile enrollment**command in global configuration mode. To delete all information associated with this enrollment profile, use the **no** form of this command.

crypto ca profile enrollment *label*
no crypto ca profile enrollment *label*

Syntax Description

label	Name for the enrollment profile; the enrollment profile name must match the name specified in the **enrollment profile** command.

Command Default

An enrollment profile does not exist.

Command Modes

Global configuration

Command History

Release	Modification
12.2(13)ZH	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.

Usage Guidelines

Before entering this command, you must specify a named enrollment profile using the **enrollment profile** in ca-trustpoint configuration mode.

After entering the **crypto ca profile enrollment** command, you can use any of the following commands to define the profile parameters:

- **authentication command** --Specifies the HTTP command that is sent to the certification authority (CA) for authentication.
- **authentication terminal** --Specifies manual cut-and-paste certificate authentication requests.
- **authentication url** --Specifies the URL of the CA server to which to send authentication requests.
- **enrollment command** --Specifies the HTTP command that is sent to the CA for enrollment.
- **enrollment terminal** --Specifies manual cut-and-paste certificate enrollment.
- **enrollment url** --Specifies the URL of the CA server to which to send enrollment requests.
- **parameter** --Specifies parameters for an enrollment profile. This command can be used only if the **authentication command** or the **enrollment command** is used.

Note The **authentication url**, **enrollment url**, **authentication terminal**, and **enrollment terminal** commands allow you to specify different methods for certificate authentication and enrollment, such as TFTP authentication and manual enrollment.

Examples

The following example shows how to define the enrollment profile named "E" and associated profile parameters:

```
crypto ca trustpoint Entrust
  enrollment profile E
  serial
crypto ca profile enrollment E
 authentication url  http://entrust:81
 authentication command  GET /certs/cacert.der
 enrollment url  http://entrust:81/cda-cgi/clientcgi.exe
 enrollment command  POST reference_number=$P2&authcode=$P1
&retrievedAs=rawDER&action=getServerCert&pkcs10Request=$REQ
 parameter 1 value aaaa-bbbb-cccc
 parameter 2 value 5001
```

Related Commands

Command	Description
crypto ca trustpoint	Declares the CA that your router should use.
enrollment profile	Specifies that an enrollment profile can be used for certificate authentication and enrollment.

crypto ca trusted-root

The **crypto ca trusted-root** command is replaced by the crypto ca trustpoint command. See the **crypto ca trustpoint**command for more information.

crypto ca trustpoint

Note Effective with Cisco IOS Release 12.3(8)T, 12.2(18)SXD, and 12.2(18)SXE, the **crypto ca trustpoint** command is replaced with the **crypto pki trustpoint** command. See the **crypto pki trustpoint** command for more information.

To declare the certification authority (CA) that your router should use, use the **crypto ca trustpoint** command in global configuration mode. To delete all identity information and certificates associated with the CA, use the **no** form of this command.

crypto ca trustpoint *name*
no crypto ca trustpoint *name*

Syntax Description

name	Creates a name for the CA. (If you previously declared the CA and just want to update its characteristics, specify the name you previously created.)

Command Default

Your router does not recognize any CAs until you declare a CA using this command.

Command Modes

Global configuration

Command History

Release	Modification
12.2(8)T	This command was introduced.
12.2(15)T	The **match certificate** subcommand was introduced.
12.3(7)T	This command was replaced by the **crypto pki trustpoint** command. You can still enter the **crypto ca trusted-root**or **crypto ca trustpoint** command, but the command will be written in the configuration as "crypto pki trustpoint."

Usage Guidelines

Use the **crypto ca trustpoint** command to declare a CA, which can be a self-signed root CA or a subordinate CA. Issuing the **crypto ca trustpoint** command puts you in ca-trustpoint configuration mode.

You can specify characteristics for the trustpoint CA using the following subcommands:

- **crl** --Queries the certificate revocation list (CRL) to ensure that the certificate of the peer has not been revoked.
- **default (ca-trustpoint)** --Resets the value of ca-trustpoint configuration mode subcommands to their defaults.
- **enrollment** --Specifies enrollment parameters (optional).
- **enrollment http-proxy** --Accesses the CA by HTTP through the proxy server.
- **match certificate** --Associates a certificate-based access control list (ACL) defined with the **crypto ca certificate map**command.

- **primary** --Assigns a specified trustpoint as the primary trustpoint of the router.
- **root** --Defines the Trivial File Transfer Protocol (TFTP) to get the CA certificate and specifies both a name for the server and a name for the file that will store the CA certificate.

Note Beginning with Cisco IOS Release 12.2(8)T, the **crypto ca trustpoint** command unified the functionality of the **crypto ca identity** and **crypto ca trusted-root** commands, thereby replacing these commands. Although you can still enter the **crypto ca identity** and **crypto ca trusted-root** commands, theconfiguration mode and command will be written in the configuration as **"crypto ca trustpoint**."

Examples

The following example shows how to declare the CA named "ka" and specify enrollment and CRL parameters:

```
crypto ca trustpoint ka
 enrollment url http://kahului:80
```

The following example shows a certificate-based access control list (ACL) with the label "Group" defined in a **crypto ca certificate map** command and included in the **match certificate** subcommand of the **crypto ca | pki trustpoint** command:

```
crypto ca certificate map Group 10
 subject-name co ou=WAN
 subject-name co o=Cisco
!
crypto ca trustpoint pki
 match certificate Group
```

Related Commands

Command	Description
crl	Queries the CRL to ensure that the certificate of the peer has not been revoked.
default (ca-trustpoint)	Resets the value of a ca-trustpoint configuration subcommand to its default.
enrollment	Specifies the enrollment parameters of your CA.
enrollment http-proxy	Accesses the CA by HTTP through the proxy server.
primary	Assigns a specified trustpoint as the primary trustpoint of the router.
root	Obtains the CA certificate via TFTP.

crypto call admission limit

To specify the maximum number of Internet Key Exchange (IKE) security associations (SAs) that the device can establish before IKE begins rejecting new SA requests, use the **crypto call admission limit** command in global configuration mode. To disable this feature, use the **no** form of this command.

crypto call admission limit ike {**in-negotiation-sa** *number* | **sa** *number*}
no crypto call admission limit ike {**in-negotiation-sa** *number* | **sa** *number*}

Syntax Description

ike	Configures the crypto Call Admission Control active IKE SA limit.
in-negotiation-sa *number*	Specifies the maximum number of in-negotiation IKE SAs allowed. Range is from 10 to 99999.
sa *number*	Specifies the number of active IKE SAs allowed on the device. Range is from 0 to 99999.

Command Default

The maximum number of IKE SAs is not specified.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.3(8)T	This command was introduced.
12.2(18)SXD1	This command was integrated into Cisco IOS Release 12.2(18)SXD1 on the Cisco 6500 and Cisco 7600.
12.4(6)T	This command was modified. The **in-negotiation-sa** *number* keyword-argument pair was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA on the Cisco 7600. The **in-negotiation-sa** *number* keyword-argument pair was not supported.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH. The **in-negotiation-sa** *number* keyword-argument pair was not supported.

Usage Guidelines

Use this command to limit the number of IKE SAs permitted to or from a device. By limiting the number of IKE SAs that can be created on the device, you can prevent the device from being impacted due to sudden inflow of IKE SA requests. The ideal limit depends on the particular platform, the network topology, the application, and traffic patterns. When the specified limit is reached, IKE rejects all new SA requests. If you specify an IKE SA limit that is less than the current number of active IKE SAs, a warning is displayed, but SAs are not terminated. New SA requests are rejected until the active SA count is below the configured limit.

Examples

The following example shows how to configure a maximum of 50 IKE SAs before IKE begins rejecting new SA requests.

```
Device(config)# crypto call admission limit ike sa 50
```

The following example shows how to configure a maximum of 100 in-negotiation IKE SAs before IKE begins rejecting new SA requests.

```
Device(config)# crypto call admission limit ike in-negotiation-sa 100
```

Related Commands

Command	Description
show crypto call admission statistics	Monitors Crypto CAC statistics.

crypto connect vlan

To create an interface VLAN for an IPSec VPN SPA and enter crypto-connect mode, use the **crypto connect vlan** command in interface configuration mode. To remove the interface VLAN status from the VLAN, use the **no** form of this command.

crypto connect vlan *vlan-id*
no crypto connect [**vlan** *vlan-id*]

Syntax Description

vlan-id	VLAN ID number.

Command Default

No default behavior or values.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(18)SXE2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

You can enter the **crypto connect vlan** command only from the following:

- The associated port VLAN interface when the EtherChannel interface (port-channel interface) and participating interfaces are switch ports.
- The EtherChannel interface when the EtherChannel interface (port-channel interface) and participant interfaces are routed ports.

The **crypto engine subslot**command is only available for VLANs prior to the VLANs being made interface VLANs by the **crypto connect vlan** command.

When you enter the **crypto connect vlan** command, a target VLAN is made an interface VLAN if and only if the target VLAN is not currently an interface VLAN, and the target VLAN has been added to an inside trunk port using the **crypto engine subslot** command. If the VLAN has been added to more than one inside trunk port, the **crypto connect vlan** command is rejected.

The **no crypto engine subslot** command is allowed only after you enter the **no crypto connect vlan** command, or before you enter the **crypto connect vlan** command.

When you remove an interface VLAN from an inside trunk port and a corresponding crypto engine subslot configuration state exists, then that crypto engine subslot configuration state is not removed. If you remove a VLAN that has a crypto engine subslot configuration state, you need to manually add it back to recover. While in this inconsistent state, any attempt to enter the **no crypto connect vlan** command is rejected.

When you enter the **no crypto connect vlan** command, the interface VLAN status is removed from a VLAN. Any associated crypto engine subslot configuration state is not altered.

Examples

The following example adds port 2/1 to the outside access port VLAN and connects the outside access port VLAN to the inside interface VLAN:

```
Router(config)# interface Vlan101
Router(config-if)# ip address 192.168.101.1 255.255.255.0
Router(config-if)# crypto map cmap
Router(config-if)# crypto engine subslot 3/0
Router(config-if)# interface GigabitEthernet2/1
Router(config-if)# crypto connect vlan 101
```

Related Commands

Command	Description
crypto engine subslot	Assign an interface VLAN that requires encryption to the IPSec VPN SPA.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
show crypto vlan	Displays the VPN running state for an IPSec VPN SPA.

crypto ctcp

To configure Cisco Tunneling Control Protocol (cTCP) encapsulation for Easy VPN, use the **crypto ctcp**command in global configuration mode. To remove the cTCP encapsulation, use the **no** form of this command.

crypto ctcp [{**keepalive** *number-of-seconds* | **port** *port-number*}]
no crypto ctcp [{**keepalive** *number-of-seconds* | **port** *port-number*}]

Syntax Description

keepalive	(Optional) Sets the interval of cTCP keepalives that are sent by the remote device. **Note** This command is configured on the remote device.
number-of-seconds	(Optional) Number of seconds between the keepalives. Value = 5 through 3600. If the **keepalive** keyword is not configured, the default is 5.
port	(Optional) Port number that cTCP will listen to. Up to 10 numbers can be configured. **Note** This keyword is configured only on the server.
port-number	(Optional) Actual port number. Value = 1 through 65535. If the **port** keyword is not configured, the default port number is 10000.

Command Default

cTCP encapsulation is not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(9)T	The **crypto ctcp** command was introduced.
12.4(20)T	The **keepalive** keyword and *number-of-seconds* argument were added.

Usage Guidelines

If cTCP is enabled on a port, any application that uses that port will not function.

When cTCP encapsulation is enabled on the router, only packets less than or equal to 1407 in size can pass through the IPsec tunnel with the Don’t Fragment (DF) bit set. If an attempt is made to send a larger size packet, the following syslog message is generated:

```
CRYPTO_ENGINE: locally-sourced pkt w/DF bit set is too big,ip->tl=1450, mtu=1407
```

Note If a Cisco IOS device is acting as a remote device, it has to send keepalives periodically to keep Network Address Translation (NAT) or firewall sessions from timing out.

Examples

The following example shows that cTCP encapsulation has been configured on port 120:

```
Router (config)# crypto ctcp port 120
```

The following example shows that the cTCP keepalive interval has been set at 30 seconds:

```
Router (config)# crypto ctcp keepalive 30
```

Related Commands

Command	Description
clear crypto ctcp	Clears cTCP encapsulation.
ctcp port	Sets the port number for cTCP encapsulation for Easy VPN.
debug crypto ctcp	Displays information about a cTCP session.
show crypto ctcp	Displays information about a cTCP session.

crypto dynamic-map

To create a dynamic crypto map entry and enter crypto map configuration command mode, use the **crypto dynamic-map** command in global configuration mode. To delete a dynamic crypto map set or entry, use the **no** form of this command.

crypto dynamic-map *dynamic-map-name dynamic-seq-num*
no crypto dynamic-map *dynamic-map-name* [*dynamic-seq-num*]

Syntax Description

dynamic-map-name	Specifies the name of the dynamic crypto map set.
dynamic-seq-num	Specifies the number of the dynamic crypto map entry.

Command Default

No dynamic crypto maps exist.

Command Modes

Global configuration

Command History

Release	Modification
11.3 T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.4(15)T	This command was modified. All changes to PFS settings in the dynamic crypto map template are immediately passed on to the instantiated crypto map PFS settings.

Usage Guidelines

Use dynamic crypto maps to create policy templates that can be used when processing negotiation requests for new security associations from a remote IP security peer, even if you do not know all of the crypto map parameters required to communicate with the remote peer (such as the peer’s IP address). For example, if you do not know about all the IPsec remote peers in your network, a dynamic crypto map allows you to accept requests for new security associations from previously unknown peers. (However, these requests are not processed until the Internet Key Exchange authentication has completed successfully.)

When a router receives a negotiation request via IKE from another IPsec peer, the request is examined to see if it matches a crypto map entry. If the negotiation does not match any explicit crypto map entry, it will be rejected unless the crypto map set includes a reference to a dynamic crypto map.

The dynamic crypto map is a policy template; it will accept “wildcard” parameters for any parameters not explicitly stated in the dynamic crypto map entry. This allows you to set up IPsec security associations with a previously unknown IPsec peer. (The peer still must specify matching values for the nonwildcard IPsec security association negotiation parameters.)

If the router accepts the peer’s request, at the point that it installs the new IPsec security associations it also installs a temporary crypto map entry. This entry is filled in with the results of the negotiation. At this point, the router performs normal processing, using this temporary crypto map entry as a normal entry, even requesting new security associations if the current ones are expiring (based upon the policy specified in the temporary crypto map entry). Once the flow expires (that is, all of the corresponding security associations expire), the temporary crypto map entry is removed.

If changes are made to the Perfect Forward Secrecy (PFS) settings in the dynamic crypto map template,the changes are passed on to the PFS settings in the instantiated crypto map. During the next rekey process the new settings are used to negotiate with the remote peer.

Dynamic crypto map sets are not used for initiating IPsec security associations. However, they are used for determining whether or not traffic should be protected.

The only configuration required in a dynamic crypto map is the **set transform-set** command. All other configuration is optional.

Dynamic crypto map entries, like regular static crypto map entries, are grouped into sets. After you define a dynamic crypto map set (which commonly contains only one map entry) using this command, you include the dynamic crypto map set in an entry of the “parent” crypto map set using the **crypto map** (IPsec global configuration) command. The parent crypto map set is then applied to an interface.

You should make crypto map entries referencing dynamic maps the lowest priority map entries, so that negotiations for security associations will try to match the static crypto map entries first. Only after the negotiation request does not match any of the static map entries do you want it to be evaluated against the dynamic map.

To make a dynamic crypto map the lowest priority map entry, give the map entry referencing the dynamic crypto map the highest *seq-num* of all the map entries in a crypto map set.

For both static and dynamic crypto maps, if unprotected inbound traffic matches a **permit** statement in an access list, and the corresponding crypto map entry is tagged as “IPsec,” then the traffic is dropped because it is not IPsec protected. (This is because the security policy as specified by the crypto map entry states that this traffic must be IPsec protected.)

For static crypto map entries, if outbound traffic matches a **permit** statement in an access list and the corresponding security association (SA) is not yet established, the router will initiate new SAs with the remote peer. In the case of dynamic crypto map entries, if no SA existed, the traffic would simply be dropped (because dynamic crypto maps are not used for initiating new SAs).

Note Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If it is possible for the traffic covered by such a **permit** entry to include multicast or broadcast traffic, the access list should include **deny** entries for the appropriate address range. Access lists should also include **deny** entries for network and subnet broadcast traffic, and for any other traffic that should not be IPsec protected.

Examples

The following example shows how to configure an IPsec crypto map set.

Crypto map entry “mymap 30” references the dynamic crypto map set “mydynamicmap,” which can be used to process inbound security association negotiation requests that do not match “mymap” entries 10 or 20. In this case, if the peer specifies a transform set that matches one of the transform sets specified in “mydynamicmap,” for a flow “permitted” by the access list 103, IPsec will accept the request and set up security associations with the remote peer without previously knowing about the remote peer. If accepted, the resulting security associations (and temporary crypto map entry) are established according to the settings specified by the remote peer.

The access list associated with “mydynamicmap 10” is also used as a filter. Inbound packets that match a **permit** statement in this list are dropped for not being IPsec protected. (The same is true for access lists associated with static crypto maps entries.) Outbound packets that match a **permit** statement without an existing corresponding IPsec SA are also dropped.

```
crypto map mymap 10 ipsec-isakmp
 match address 101
 set transform-set my_t_set1
 set peer 10.0.0.1
 set peer 10.0.0.2
crypto map mymap 20 ipsec-isakmp
 match address 102
 set transform-set my_t_set1 my_t_set2
 set peer 10.0.0.3
crypto map mymap 30 ipsec-isakmp dynamic mydynamicmap
!
crypto dynamic-map mydynamicmap 10
 match address 103
 set transform-set my_t_set1 my_t_set2 my_t_set3
```

Related Commands

Command	Description
crypto map (global IPsec)	Creates or modifies a crypto map entry and enters the crypto map configuration mode.
crypto map (interface IPsec)	Applies a previously defined crypto map set to an interface.
crypto map local-address	Specifies and names an identifying interface to be used by the crypto map for IPsec traffic.
match address (IPsec)	Specifies an extended access list for a crypto map entry.
set peer (IPsec)	Specifies an IPsec peer in a crypto map entry.
set pfs	Specifies that IPsec should ask for PFS when requesting new security associations for this crypto map entry, or that IPsec requires PFS when receiving requests for new security associations.
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPsec security associations.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto engine accelerator logs	Displays a dynamic crypto map set.
show crypto map (IPsec)	Displays the crypto map configuration.

crypto-engine

To enter the QoS policy map configuration mode for the IPsec VPN module, use the **crypto-engine** command in interface configuration mode.

crypto-engine

Syntax Description

This command has no arguments or keywords.

Command Default

This command has no default settings.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.2(33)SXI	Support for this command was introduced.

Usage Guidelines

Once you enter the **crypto-engine** command, the prompt changes to the following:

Router(config-crypto-engine)#

The following crypto engine configuration commands are available when you enter the **crypto-engine** command:

- **default** --Sets a command to its defaults.
- **exit** --Exit service-flow submode.
- **no** --Negates a command or set its defaults.
- **service-policy output** *policy-map-name* --Configures the service policy by assigning a policy map to the output of an interface.

Examples

The following example shows how to apply the policy map to tunnel egress traffic:

```
Router(config)# interface tunnel1

Router(config-if)# crypto-engine
Router(config-crypto-engine)# service-policy output crypto1
```

Related Commands

Command	Description
show policy-map interface	Displays the statistics and configurations of the QoS policies attached to the tunnel interface.

crypto engine accelerator

Note Effective with Cisco IOS Release 12.3(11)T, this command is replaced by the **crypto engine aim**, **crypto engine em**, **crypto engine nm**, **crypto engine onboard**, and **crypto engine slot** commands. See these commands for more information.

To enable the onboard hardware accelerator of the router for IP security (IPsec) encryption, use the **crypto engine accelerator** command in global configuration mode. To disable the use of the onboard hardware IPsec accelerator, and thereby perform IPsec encryption or decryption in software, use the **no** form of this command.

crypto engine accelerator
no crypto engine accelerator

Syntax Description

This command has no arguments or keywords.

Command Default

The hardware accelerator for IPsec encryption is enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(3)T	This command was introduced for the Cisco 1700 series router and other Cisco routers that support hardware accelerators for IPsec encryption.
12.1(3)XL	Support was added for the Cisco uBR905 cable access router.
12.2(2)XA	Support was added for the Cisco uBR925 cable access router.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T and implemented for the AIM-VPN/EPII and AIM-VPN/HPII on the following platforms: Cisco 2691, Cisco 3660, Cisco 3725, and Cisco 3745.
12.2(15)ZJ	This command was implemented for the AIM-VPN/BPII on the following platforms: Cisco 2610XM, Cisco 2611XM, Cisco 2620XM, Cisco 2621XM, Cisco 2650XM, and Cisco 2651XM.
12.3(4)T	The AIM-VPN/BPII was integrated into Cisco IOS Release 12.3(4)T on the following platforms: Cisco 2610XM, Cisco 2611XM, Cisco 2620XM, Cisco 2621XM, Cisco 2650XM, and Cisco 2651XM.
12.3(11)T	This command was replaced by the **crypto engine aim**, **crypto engine em**, **crypto engine nm**, **crypto engine onboard**, and **crypto engine slot** commands.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

This command is not normally needed for typical operations because the onboard hardware accelerator of the router is enabled for IPsec encryption by default. The hardware accelerator should not be disabled except on instruction from Cisco Technical Assistance Center (TAC) personnel.

Examples

The following example shows how to disable the onboard hardware accelerator of the router for IPsec encryption. This disabling is normally needed only after the accelerator has been disabled for testing or debugging purposes.

```
Router(config)# no crypto engine accelerator
Warning! all current connections will be torn down.
Do you want to continue? [yes/no]:
```

Related Commands

Command	Description
clear crypto engine accelerator counter	Resets the statistical and error counters for the hardware accelerator to zero.
crypto ca	Defines the parameters for the certification authority used for a session.
crypto cisco	Defines the encryption algorithms and other parameters for a session.
crypto dynamic-map	Creates a dynamic map crypto configuration for a session.
crypto ipsec	Defines the IPSec security associations and transformation sets.
crypto isakmp	Enables and defines the IKE protocol and its parameters.
crypto key	Generates and exchanges keys for a cryptographic session.
crypto map	Creates and modifies a crypto map for a session.
debug crypto engine accelerator control	Displays each control command as it is given to the crypto engine.
debug crypto engine accelerator packet	Displays information about each packet sent for encryption and decryption.
show crypto engine accelerator ring	Displays the contents of command and transmits rings for the crypto engine.
show crypto engine accelerator sa database	Displays the active (in-use) entries in the crypto engine SA database.
show crypto engine accelerator statistic	Displays the current run-time statistics and error counters for the crypto engine.
show crypto engine brief	Displays a summary of the configuration information for the crypto engine.
show crypto engine configuration	Displays the version and configuration information for the crypto engine.

Command	Description
show crypto engine connections	Displays a list of the current connections maintained by the crypto engine.

crypto engine aim

To reenable an advanced integration module (AIM), use the **crypto engine aim**command in global configuration mode. To disable an AIM encryption module, use the **no** form of this command.

crypto engine aim *aim-slot-number*
no crypto engine aim *aim-slot-number*

Syntax Description

aim-slot-number	Slot number to which an AIM is to be reenabled or disabled.

Command Default

An AIM is neither reenabled nor disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced. This command replaces the **crypto engine accelerator** command.

Usage Guidelines

The **crypto engine accelerator** command will still be usable for a while, but if it is used, only the **crypto engine aim** command will be saved to the running and startup (nonvolatile memory) configuration.

Examples

The following example shows that the AIM in slot 0 is to be reenabled:

```
crypto engine aim 0
```

The following example shows that the AIM in slot 0 is to be disabled:

```
no crypto engine aim 0
```

crypto engine compliance shield disable

To effectively allow weak cryptographic algorithms such as Message Direct 5 (MD5), Data Encryption Standard (DES), or weak RSA keys to be enabled by various features, perform the **crypto engine compliance shield disable** command.

To prevent the weak crypto algorithms from being used by features, use the **no** form of this command.

crypto engine compliance sheild disable
no crypto engine compliance sheild disable

Syntax Description

This command has no arguments or keywords.

Command Default

Weak crypto algorithm check is enabled by default.

Command Modes

Global Configuration

Command History

Release	Modification
Cisco IOS XE Cupertino 17.7.1	This command was introduced.

Usage Guidelines

Weak cryptographic algorithms are insecure and do not provide adequate protection from modern threats when used with features such as IPsec, SSH, etc.

Cisco does not recommend using this command to bypass a weak crypto algorithm check and should only be used as last resort.

For more information on which cryptographic algorithms are considered weak, please refer to the sections on algorithms with a status of “Avoid” or “Legacy” in the Next Generation Cryptography document.

Changing the compliance shield status will require a device reboot to take effect.

Examples

The following example shows when the weak crypto algorithm check is disabled:

```
device(config)# crypto engine compliance shield disable
Disable compliance shield mode will take effect after reboot!
```

crypto engine em

To enable the hardware accelerator of an expansion slot for IP security (IPsec) encryption, use the **crypto engine em**command in global configuration mode. To disable the hardware accelerator of the expansion slot, use the **no** form of this command.

crypto engine em *slot-number*
no crypto engine em *slot-number*

Syntax Description

slot-number	Slot number to which the hardware accelerator of the expansion slot is to be enabled or disabled (applies to slots 0 through 3).

Command Default

The hardware accelerator is neither enabled nor disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced. This command replaces the **crypto engine accelerator** command.

Usage Guidelines

The **crypto engine accelerator** command will still be usable for a while, but if it is used, only the **crypto engine em** command will saved to the running and startup (nonvolatile memory) configuration.

Examples

The following example shows that the hardware accelerator of expansion slot 1 is to be enabled:

```
crypto engine em 1
```

The following example shows that the hardware accelerator of expansion slot 1 is to be disabled:

```
no crypto engine em 1
```

crypto engine mode vrf

To enable VRF-Aware mode for the IPSec VPN SPA, use the **crypto engine mode vrf**command in global configuration mode. To disable VRF-aware mode, use the **no** form of this command.

crypto engine mode *vrf*
no crypto engine mode *vrf*

Command Default

No default behavior or values.

Command Modes

Global configuration

Command History

Release	Modification
12.2(18)SXE2	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

The VRF-Aware IPSec feature introduces IPSec tunnel mapping to Multiprotocol Label Switching (MPLS) VPNs.

Using the VRF-Aware IPSec feature, you can map IPSec tunnels to VPN routing and forwarding instances (VRFs) using a single public-facing address.

Unlike other IPSec VPN SPA feature configurations, when configuring VRF-Aware features, you do not use the **crypto connect vlan** command.

Examples

The following example shows a VRF-Aware IPSec implementation:

```
ip vrf pepsi
 rd 1000:1
 route-target export 1000:1
 route-target import 1000:1
!
ip vrf coke
 rd 2000:1
 route-target export 2000:1
 route-target import 2000:1
crypto engine mode vrf
interface vlan 100
 ip vrf forwarding pepsi
 ip address 10.2.1.1 255.255.255.0
 crypto engine subslot 3/0
 crypto map map100
interface vlan 200
 ip vrf forwarding coke
 ip address 10.2.1.1 255.255.255.0
 crypto engine subslot 3/0
 crypto map map200
interface gi1/1 (hidden VLAN 1000)
 ip address 171.1.1.1
 crypto engine subslot 3/0
! BASIC MPLS CONFIGURATION
```

```
mpls label protocol ldp
tag-switching tdp router-id Loopback0
mls ip multicast flow-stat-timer 9
no mls flow ip
no mls flow ipv6
!
! CONFIGURE THE INTERFACE CONNECTED TO THE MPLS BACKBONE WITH LABEL/TAG SWITCHING
interface GigabitEthernet2/12
 ip address 20.1.0.34 255.255.255.252
 logging event link-status
 speed nonegotiate
 mpls label protocol ldp
tag-switching ip
```

Related Commands

Command	Description
crypto engine subslot	Assigns an interface VLAN that requires encryption to the IPSec VPN SPA.
ip vrf	Configures a VRF routing table and enters VRF configuration mode.
ip vrf forwarding	Associates a VRF with an interface or subinterface.
vrf	Defines the VRF to which the IPSec tunnel will be mapped.

crypto engine nm

To enable the onboard hardware accelerator of a network module for IP security (IPsec) encryption, use the **crypto engine nm**command in global configuration mode. To disable the accelerator of the network module, use the **no** form of this command.

crypto engine nm *slot-number*
no crypto engine nm *slot-number*

Syntax Description

slot-number	Slot number to which the hardware accelerator of a network module is to be enabled or disabled (applies to slots 0 through 5).

Command Default

The hardware accelerator is neither enabled nor disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced. This command replaces the **crypto engine accelerator** command.

Usage Guidelines

The **crypto engine accelerator** command will still be usable for a while, but if it is used, only the **crypto engine nm** command will saved to the running and startup (nonvolatile memory) configuration.

Examples

The following example shows that the hardware accelerator of the network module in slot 0 is to be enabled:

```
crypto engine nm 0
```

The following example shows that the hardware accelerator of the network module in slot 0 is to be disabled:

```
no crypto engine nm 0
```

crypto engine onboard

To enable the hardware accelerator of an onboard module for IP security (IPsec) encryption, use the **crypto engine onboard**command in global configuration mode. To disable the hardware accelerator of the onboard module, use the **no** form of this command.

crypto engine onboard *slot-number*
no crypto engine onboard *slot-number*

Syntax Description

slot-number	Slot number to which the hardware accelerator of the onboard module is to be enabled or disabled (applies to slots 0 and 1).

Command Default

The hardware accelerator is neither enabled nor disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced. This command replaces the **crypto engine accelerator** command.

Usage Guidelines

The **crypto engine accelerator** command will still be usable for a while, but if it is used, only the **crypto engine onboard** command will saved to the running and startup (nonvolatile memory) configuration.

Examples

The following example shows that the hardware accelerator of the onboard module in slot 1 is to be enabled:

```
crypto engine onboard 1
```

The following example shows that the hardware accelerator of the onboard module in slot 1 is to be disabled:

```
no crypto engine onboard 1
```

crypto engine slot

To enable a hardware accelerator, such ISM-VPN (supported by ISR G2 routers) in a service adapter, use the **crypto engine slot** command in global configuration mode. To disable the hardware accelerator in the service adapter, use the **no** form of this command.

crypto engine slot *slot-number*
no crypto engine slot *slot-number*

Syntax Description

slot-number	Slot number to which the hardware accelerator in a service adapter is to be reenabled or disabled (applies to slots 0 through 6).

Command Default

The hardware accelerator is neither enabled nor disabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(11)T	This command was introduced. This command replaces the **crypto engine accelerator** command.

Usage Guidelines

The **crypto engine accelerator** command will still be usable for a while, but if it is used, only the **crypto engine slot** command will be saved to the running and startup (nonvolatile memory) configuration.

Examples

The following example shows how to enable the hardware accelerator of the service adaptor in slot 2:

```
crypto engine slot 2
```

The following example shows how to disable the hardware accelerator of the service adaptor in slot 2:

```
no crypto engine slot 2
```

The following example shows how to enable ISM VPN in slot 0:

```
crypto engine slot 0
```

crypto engine slot (interface)

To assign an interface VLAN, Virtual Routing and Forwarding (VRF) tunnel interface, or Front-door VRF (FVRF) interface that requires encryption to the IPSec VPN Shared Port Adapter (SPA), use the **crypto engine slot** command in interface configuration mode. The command usage and syntax varies based on whether you are in crypto-connect mode or VRF mode. In crypto-connect mode, the command is applied to interface VLANs and only the *slot*/*subslot* arguments are specified; in VRF-mode, the command is applied to interface VLANs, tunnel interfaces, or FVRF interfaces and either the **inside** or **outside** keyword must also be specified. To remove the interface, use the corresponding **no** form of this command.

Crypto-Connect Mode Syntax
crypto engine slot *slot*
no crypto engine slot *slot*

VRF Mode Syntax
crypto engine slot *slot* {**inside** | **outside**}
no crypto engine slot *slot* {**inside** | **outside**}

Syntax Description

slot	Chassis slot number where the Cisco 7600 SSC-400 card is located. Refer to the appropriate hardware manual for slot information. For SIPs and SSCs, refer to the platform-specific SPA hardware installation guide or the corresponding “Identifying Slots and Subslots for SIPs and SPAs” topic in the platform-specific SPA software configuration guide.
inside	(VRF Mode Only) Identifies the interface as an interface VLAN or tunnel interface.
outside	(VRF Mode Only) Identifies the interface as an FVRF interface.

Command Default

No interface is assigned.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.2(33)SRA	This command was introduced into Cisco IOS Release 12.2(33)SRA to support the IPSec VPN SPA on Cisco 7600 series routers and Catalyst 6500 series switches.
12.2(33)SRE	This command was modified. The *subslot* argument was removed.

Usage Guidelines

Usage guidelines vary based on whether you are in crypto-connect mode or VRF mode:

Crypto-Connect Mode Usage Guidelines

With this command, you do not need to explicitly add interface VLANs to the IPSec VPN SPA inside trunk port.

It is strongly recommended that you use the **crypto engine slot** command instead of manually adding and removing VLANs from the inside trunk port.

When you add an interface VLAN to an inside trunk port and that interface VLAN is not already added to another inside trunk port, the crypto engine slot configuration state on the interface VLAN is combined. If the interface VLAN is already added to another inside trunk port, the command is rejected.

You should not try to add all VLANs at one time (If you attempt this, you can recover by manually removing the VLANs from the inside trunk port.)

In crypto-connect mode, the **crypto engine slot** command is used in conjunction with the **crypto connect vlan** command.

In crypto-connect mode, the **crypto engine slot** command is only available for VLANs prior to the VLANs being made interface VLANs by the **crypto connect vlan** command.

The **crypto engine slot** command is rejected if you enter it on a crypto-connected interface VLAN whose current crypto engine slot configuration is different from the subslot specified in the **crypto engine slot** command. To change the crypto engine slot configuration on an interface VLAN, you must ensure that the VLAN is not crypto-connected.

If you change the crypto engine slot configuration on an interface VLAN, any IPSec and IKE SAs that are currently active on that interface VLAN are deleted.

If you enter the **no crypto engine slot** command and the interface VLAN is crypto-connected, the **no crypto engine slot** command is rejected. The **no crypto engine slot** command is allowed only after you enter the **no crypto connect vlan** command, or before you enter the **crypto connect vlan** command.

When you remove an interface VLAN from an inside trunk port and a corresponding crypto engine slot configuration state exists, then that crypto engine slot configuration state is not removed. If you remove a VLAN that has a crypto engine slot configuration state, you need to manually add it back to recover. While in this inconsistent state, any attempt to enter the **no crypto connect vlan** command is rejected.

When you enter the **no crypto connect vlan** command, the interface VLAN status is removed from a VLAN. Any associated crypto engine slot configuration state is not altered.

When you **write** the configuration or **show** the configuration, the crypto engine slot configuration state is expressed in the context of the associated interface VLAN. The interface VLAN is also shown as having been added to the appropriate inside trunk port. This is the case even if the configuration was loaded from a legacy (pre-crypto engine slot) configuration file, or if VLANs were manually added instead of being added through the **crypto engine slot** command.

By editing the **crypto engine slot** commands and inside trunk port VLANs, it is possible to produce an inconsistent configuration file.

VRF Mode Usage Guidelines

When configuring an interface VLAN or tunnel interface in VRF mode, the **crypto-engine slot inside** command must be specified.

When configuring an FVRF interface in VRF mode, the **crypto-engine slot outside** command must be specified.

In VRF mode, the **crypto-connect vlan** command is not used.

In Cisco IOS Release 12.2(33)SRE and later releases the *subslot* argument was removed.

Examples

The following crypto-connect mode example shows how to assign VLAN interface 101 to the IPSec VPN SPA in slot 3, subslot 0:

```
Router(config)# interface Vlan101
Router(config-if)# ip address 192.168.101.1 255.255.255.0
Router(config-if)# crypto map cmap
Router(config-if)# crypto engine slot 3/0
Router(config)# interface GigabitEthernet2/1
Router(config-if)# crypto connect Vlan101
```

The following VRF mode example shows how to assign VLAN interface 101 to the IPSec VPN SPA in slot 3, subslot 0:

```
Router(config)# interface Vlan101
Router(config-if)# ip vrf forwarding abc
Router(config-if)# ip address 10.2.1.1 255.255.255.0
Router(config-if)# crypto engine slot 3/0 inside
Router(config-if)# crypto map map100
```

The following VRF mode example shows how to assign Tunnel interface 1 to the IPSec VPN SPA in slot 4, subslot 0:

```
Router(config)# interface Tunnel1
Router(config)# ip vrf forwarding abc
Router(config-if)# ip address 10.1.1.254 255.255.255.0
Router(config-if)# tunnel source 172.1.1.1
Router(config-if)# tunnel destination 100.1.1.1
Router(config-if)# tunnel mode ipsec profile tp
Router(config-if)# crypto engine slot 4/0 inside
```

The following VRF mode example assigns the WAN-side interface GigabitEthernet1/1 to the IPSec VPN SPA in slot 3, subslot 0:

```
Router(config)# interface GigabitEthernet1/1
Router(config-if)# ip address 171.1.1.1 255.255.255.0
Router(config-if)# crypto engine slot 3/0 outside
```

Related Commands

Command	Description
crypto connect vlan	Creates an interface VLAN for an IPSec VPN SPA and enters crypto-connect mode.
crypto map (interface IPSec)	Applies a previously defined crypto map set to an interface.
ip vrf forwarding	Associates a VRF with an interface.
show crypto vlan	Displays the VPN running state for an IPSec VPN SPA.
tunnel vrf	Associates a VRF instance with a specific tunnel interface.

crypto gdoi ks

To trigger a rekey of group members in a GET VPN network, use the **crypto gdoi ks** command in privileged EXEC mode.

crypto gdoi ks [**group** *group-name*] **rekey** [**replace-now**]

Syntax Description

group *group-name*	(Optional) Name of the group.
rekey	Sends a rekey message based on the latest security policy in the running configuration.
replace-now	(Optional) Removes the old TEKs and KEK from group members (GMs) immediately and installs the new TEKs and KEK.

Command Default

No rekey is triggered.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.2(1)T	This command was introduced.
Cisco IOS XE Release 3.8S	This command was integrated into Cisco IOS XE Release 3.8S.

Usage Guidelines

When you change the policy (for example, from DES to AES) on the key server (KS) and exit from global configuration mode, a syslog message appears on the primary KS indicating that the policy has changed and a rekey is needed. You can enter the **crypto gdoi ks** command to send a rekey based on the latest security policy in the running configuration.

When each GM receives this triggered rekey, it installs the new SAs (for example, for AES) and shortens the lifetimes of the old SAs (for example, for DES). The GM continues to encrypt and decrypt traffic using the old SAs until their shortened lifetimes expire.

For GMs that are running older versions that do not yet support the **crypto gdoi ks** command, the primary KS uses the software versioning feature to detect those versions and only triggers a rekey without sending instruction for policy replacement. Therefore, when a GM receives the rekey, it installs the new SAs but does not shorten the lifetimes of the old SAs. (This behavior is the same as the prior rekey method and ensures backward compatibility for devices that cannot support policy replacement.)

If the **replace-now** keyword is used, the GM that receives the rekey will immediately remove the old TEKs and KEK and install the new TEKs and KEK. Therefore, the new policy takes effect immediately without waiting for existing policy SAs to expire.

You must use this command on the KS or primary KS. If you try to use this command on the secondary KS, it rejects the command as shown below:

```
Device# crypto gdoi ks rekey

ERROR for group GET: This command must be executed on Pri-KS
```

Note

The **replace-now** keyword could cause a temporary traffic discontinuity, because all GMs may not receive the rekey message at the same time.

Examples

The following example shows how to trigger a rekey on all GMs:

```
Device# crypto gdoi ks rekey
Device#
*Jan 28 09:17:44.363: %GDOI-5-KS_SEND_UNICAST_REKEY: Sending Unicast Rekey with
policy-replace for group GET from address 10.0.8.1 with seq # 2
```

The following example shows how to remove the old TEKs and KEK from GMs immediately and install the new TEKs and KEK:

```
Device# crypto gdoi ks rekey replace-now
```

Related Commands

Command	Description
show crypto gdoi feature	Displays the version of the GET VPN software running on each KS and GM in the GET VPN network and displays whether each device is running a version that supports GM removal, rekey triggering with policy replacement, or the GDOI MIB.

crypto gdoi gm

For group members to change the IP security (IPsec) security association (SA) status, use the **crypto gdoi gm**command in privileged EXEC mode.

crypto gdoi gm group *group-name* {**ipsec direction inbound optional** | **ipsec direction inbound only** | **ipsec direction both**}

Syntax Description

group *group-name*	Name of the group.
ipsec direction inbound optional	Allows a group member to change the IPsec SA status to inbound optional. IPsec SA will accept cipher or plain text or both and will encrypt the packet before forwarding it.
ipsec direction inbound only	Allows a group member to change the IPsec SA status to inbound only. IPsec SA will accept cipher or plain text or both and will forward the packet in clear text.
ipsec direction both	Allows a group member to change the IPsec SA status to both inbound and outbound. IPsec SA will accept only cipher text and will encrypt the packet before forwarding it.

Command Default

If the **sa receive-only**command is specified on the key server, the group member remains in receive-only mode.

Command Modes

Privileged EXEC

Command History

Release	Modification
12.4(11)T	This command was introduced.

Usage Guidelines

This command is executed on group members. This command and its various keywords aid in testing individual group members and verifies that the group members are encrypting or decrypting traffic. This command and its keywords can be used only after the **sa receive-only** command has been configured on the key server.

The **ipsec direction inbound optional**keyword is used for situations in which all group members have been instructed to install the IPsec SAs as inbound only but for which a group member wants to install the IPsec SAs as inbound optional.

The **ipsec direction inbound only** keyword is used when a group member wants to change a previously set IPsec SA status to inbound only.

The **ipsec direction both**keyword is used when a group member has to change a previously set IPsec SA status to both inbound and outbound. In this setting, the group member accepts only cipher text.

Examples

The following example shows how to determine whether a group member can accept cipher text.

On Group Member 1, configure the following:

```
crypto gdoi gm group groupexample ipsec direction inbound only
```

On Group Member 2, configure the following:

```
crypto gdoi gm group groupexample ipsec direction inbound optional
```

Then Ping Group Member 1.

Group Member 2 will have encrypted the packet and will send an encrypted packet to Group Member 1, which then decrypts that packet. If the traffic is from Group Member 1 to Group Member 2, Group Member 1 will forward the packet in clear text, and Group Member will accept it.

Related Commands

Command	Description
sa receive-only	Specifies that an IPsec SA is to be installed by a group member as "inbound only."

crypto gdoi group

To create a Group Domain of Interpretation (GDOI) group and enter GDOI group configuration mode, use the **crypto gdoi group** command in global configuration mode. To disable a GDOI group, use the **no** form of this command.

crypto gdoi group [ipv6]*group-name*
no crypto gdoi group [ipv6] *group-name*

Syntax Description

group-name	Name of the group. You can use up to 80 characters.
ipv6	Creates an IPv6 group.

Command Default

A GDOI group is not defined.

Command Modes

Global configuration

Command History

Release	Modification
12.4(6)T	This command was introduced.
15.2(3)T	This command was modified. The **ipv6** keyword was added.

Usage Guidelines

There are more options for configuring a group on a key server than there are for configuring a group member. The group is identified by an identity and by the server. If the GDOI group is a group member, the address of the server is specified. If the GDOI group is a key server, “server local” is specified, which indicates that this is the key server.

Examples

The following example shows how to configure an IPv4 GDOI group for a key server:

```
crypto gdoi group mygroup
 identity number 4444
 server local
```

The following example shows how to configure an IPv6 GDOI group for a key server:

```
crypto gdoi group ipv6 mygroup2
 identity number 4444
 server local
```

The following example shows how to configure an IPv4 GDOI group for a group member:

```
crypto gdoi group mygroup3
 identity number 3333
 server address ipv4 10.0.5.2
```

crypto identity

To configure the identity of the router with a given list of distinguished names (DNs) in the certificate of the router, use the **crypto identity** command in global configuration mode. To delete all identity information associated with a list of DNs, use the **no** form of this command.

crypto identity *name*
no crypto identity *name*

Syntax Description

name	Identity of the router, which is associated with the given list of DNs.

Command Default

If this command is not enabled, the IP address is associated with the identity of the router.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(4)T	This command was introduced.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(20)T	Support for IPv6 was added.

Usage Guidelines

The crypto identity command allows you to configure the identity of a router with a given list of DNs. Thus, when used with the dn and fqdn commands, you can set restrictions in the router configuration that prevent peers with specific certificates, especially certificates with particular DNs, from having access to selected encrypted interfaces.

Note The identity of the peer must be the same as the identity in the exchanged certificate.

Examples

The following example shows how to configure a DN-based crypto map:

```
! The following is an IPSec crypto map (part of IPSec configuration). It can be used only
! by peers that have been authenticated by DN and if the certificate belongs to BigBiz.
crypto map map-to-bigbiz 10 ipsec-isakmp
 set peer 172.21.114.196
 set transform-set my-transformset
 match address 124
 identity to-bigbiz
!
crypto identity to-bigbiz
 dn ou=BigBiz
!
!
! This crypto map can be used only by peers that have been authenticated by hostname
! and if the certificate belongs to little.com.
crypto map map-to-little-com 10 ipsec-isakmp
 set peer 172.21.115.119
 set transform-set my-transformset
 match address 125
 identity to-little-com
!
crypto identity to-little-com
 fqdn little.com
!
```

Related Commands

Command	Description
crypto mib ipsec flowmib history failure size	Associates the identity of the router with the DN in the certificate of the router.
fqdn	Associates the identity of the router with the hostname that the peer used to authenticate itself.

crypto ikev2 authorization policy

To configure an IKEv2 authorization policy, use the **crypto ikev2 authorization policy** command in global configuration mode. To remove this command and all associated subcommands from your configuration, use the **no** form of this command. To return the authorization policy to its default value, use the **default** form of this command.

crypto ikev2 authorization policy *policy-name*
no crypto ikev2 authorization policy *policy-name*
default crypto ikev2 authorization policy

Syntax Description

policy-name	Group definition that identifies which policy is enforced for users.

Command Default

The default IKEv2 authorization policy is used.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(3)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

Use the **crypto ikev2 authorization policy** command to specify the group for which a policy profile must be defined and the group policy information that needs to be defined or changed. You may wish to change the group policy on your router if you decide to connect to the client using a group ID that does not match the *policy-name* argument. The authorization policy is referred from the IKEv2 profile using the **aaa authorization group** command where the group name can be directly specified or derived from the remote identities using a name mangler.

If AAA authorization is configured as local, AAA derives the authorization attributes from IKEv2 client configuration group through the callback to crypto component.

After enabling this command, which puts the networking device in IKEv2 group authorization policy mode, you can specify the characteristics for the authorization policy using the following commands:

- **dhcp--** Configures an IP address on the remote access client for the Dynamic Host Configuration Protocol (DHCP) to use**.**
- **dns** --Specifies the primary and secondary Domain Name Service (DNS) servers for the group.
- **netmask** --Subnet mask to be used by the client for local connectivity.
- **pool** --Refers to the IP local pool address used to allocate internal IP addresses to clients.
- **subnet-acl** --Configures split tunneling.
- **wins** --Specifies the primary and secondary Windows Internet Naming Service (WINS) servers for the group.

You can modify the default authorization policy using the **crypto ikev2 authorization policy default** command. You can either modify the entire authorization policy or modify one of the above commands.

You can disable the default authorization policy using the **no crypto ikev2 authorization policy default** command. When disabled, the values in the default authorization policy are copied and the default proposal remains inactive.

Examples

The following example shows how the client configuration group is referred from IKEv2 profile using the **aaa authorization group** command where the group name is specified directly. In this example, the policy is enforced for users that matches the group name “abc.”

```
aaa new-model
aaa authorization network aaa-group-list default local
!
crypto ikev2 authorization policy
abc
  pool pool1
  dns 198.51.100.1 198.51.100.100
  wins 203.0.113.1 203.0.113.115
  dhcp server 3.3.3.3
  dhcp giaddr 192.0.2.1
  dhcp timeout 10
  netmask 255.255.255.0
  subnet-acl acl-123
!
crypto ikev2 profile profile1
  authentication remote eap
aaa authorization group aaa-group-list abc
!
ip access-list extended acl-123
permit ip 209.165.200.225 0.0.0.31 any
permit ip 209.165.201.1 255.255.255.224 any
```

Related Commands

Command	Description
aaa authorization group	Sets parameters that restrict user access to a network.
dhcp	Configures an IP address for the DHCP to use.
dns	Specifies the primary and secondary DNS servers for the group.
netmask	Specifies the netmask of the subnet address that is assigned to the client.
pool	Defines a local pool address for assigning IP addresses.
subnet-acl	Defines ACL for split tunneling.
wins	Specifies the internal WINS server addresses.

crypto ikev2 certificate-cache

To set the cache size to store certificates, use the **crypto ikev2 certificate-cache** command in global configuration mode. To delete the cache size, use the **no** form of this command.

crypto ikev2 certificate-cache *number-of-certificates*
no crypto ikev2 certificate-cache

Syntax Description

number-of-certificates	The maximum number of certificates that can be stored in the cache.

Command Default

The cache size is not set.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to set the cache to store the maximum number of certificates fetched from the HTTP URLs.

Examples

The following example sets the cache size to store 500 certificates:

```
Router(config)# crypto ikev2 certificate-cache 500
```

Related Commands

Command	Description
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 cluster

To configure an Internet Key Exchange Version 2 (IKEv2) cluster policy in a Hot Standby Router Protocol (HSRP) cluster, use the **crypto ikev2 cluster** command in global configuration mode. To remove this command and all associated commands from your configuration, use the **no** form of this command.

crypto ikev2 cluster
no crypto ikev2 cluster

Syntax Description

This command has no keywords or arguments.

Command Default

An IKEv2 cluster policy is not configured.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.2(4)M	This command was introduced.

Usage Guidelines

Use the **crypto ikev2 cluster** command to define an IKEv2 cluster policy and to enter IKEv2 cluster configuration mode.

After enabling this command, you can specify the characteristics for the cluster policy by using the following commands:

- **holdtime**
- **master**
- **port**
- **slave**
- **standby-group**

To view the cluster configuration, use the **show crypto ikev2 cluster** command.

Examples

The following example shows how to configure an IKEv2 cluster policy:

```
Device(config)# crypto ikev2 cluster
Device(config-ikev2-cluster)# master crypto-load 10
Device(config-ikev2-cluster)# slave priority 90
```

Related Commands

Command	Description
holdtime	Specifies the time interval to receive messages.
master (IKEv2)	Defines settings for the primary gateway in the HSRP cluster.
port (IKEv2)	Specifies port settings for the HSRP cluster.

Command	Description
show crypto ikev2 cluster	Displays the cluster policy configuration.
slave (IKEv2)	Defines settings for the secondary gateways in the HSRP cluster.
standby-group	Defines HSRP cluster settings.

crypto ikev2 cookie-challenge

To enable a cookie challenge for Internet Key Exchange Version 2 (IKEv2), use the **crypto ikev2 cookie-challenge** command in global configuration mode. To disable the cookie challenge, use the **no** form of this command.

crypto ikev2 cookie-challenge *number*
no crypto ikev2 cookie-challenge

Syntax Description

number	Enables the IKEv2 cookie challenge when the number of half-open security associations (SAs) crosses the configured number. The range is 1 to 1000.

Command Default

The cookie challenge is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to enable the IKEv2 cookie challenge. A cookie challenge mitigates the effect of a DoS attack when an IKEv2 responder is flooded with session initiation requests from forged IP addresses.

Examples

The following example sets the cookie challenge to 450:

```
Router(config)# crypto ikev2 cookie-challenge 450
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 cts

To enable IPsec inline tagging globally, use the **crypto ikev2 cts** command in global configuration mode. To disable the SGT inline tagging, use the **no** form of this command.

crypto ikev2 cts sgt
no crypto ikev2 cts sgt

Syntax Description

sgt	Enables Security Group Tag (SGT) IPsec inline tagging.

Command Default

IPsec inline tagging is not enabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.2(2)T	This command was introduced.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

This command applies to all sessions on the router. If IPsec inline tagging is disabled, the new negotiated sessions do not negotiate the vendor ID (VID). However, the current SA and the subsequent SA rekey are enabled with the feature until the lifetime of the SA. This applies to the new IPsec SA and the rekey of the IPsec SA established using the parent or rekeyed IKE SA.

Examples

The following example shows how to enable IPsec inline tagging on an sVTI initiator and dVTI responder.

```
crypto ikev2 proposal p1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy policy1
 proposal p1
!
crypto ikev2 keyring key
 peer peer
  address ::/0
  pre-shared-key cisco
 !
 peer v4
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco
 !
```

```
!
!
crypto ikev2 profile prof3
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring key
!
crypto ikev2 cts sgt
!
crypto ipsec transform-set trans1 esp-aes esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 10.1.1.2
 set transform-set trans
 set ikev2-profile prof3
 match address ipv4acl
!
!
interface Loopback1
 ip address 209.165.201.1 255.255.255.224
 ipv6 address 2001::4:1/112
!
interface Loopback2
 ip address 209.165.200.1 255.255.255.224
 ipv6 address 2001::40:1/112
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 192.168.210.74 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/1
 ip address 172.16.0.1 255.240.0.0
 duplex auto
 speed auto
 ipv6 address 2001::5:1/112
 ipv6 enable
 crypto map cmap
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 172.16.0.2
ip route 10.12.255.200 255.0.0.0 172.31.255.254
!
ip access-list extended ipv4acl
 permit ip host 209.165.201.1host 192.168.12.125
 permit ip host 209.165.200.1 host 172.18.0.1
 permit ip host 172.28.0.1 host 10.10.10.1
 permit ip host 10.12.255.200 host 192.168.14.1
!
logging esm config
ipv6 route ::/0 2001::5:2
!
!
!
!
```

```
!!
control-plane
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
exception data-corruption buffer truncate
scheduler allocate 20000 1000


crypto ikev2 proposal p1
 encryption aes-cbc-192
 integrity sha1
 group 15
!
crypto ikev2 policy policy1
 proposal p1
!
crypto ikev2 keyring key
 peer peer
  address 172.160.1.1 255.240.0.0
  pre-shared-key cisco
 !
 peer v4_p2
  address 172.31.255.1 255.240.0.0
  pre-shared-key cisco
 !
crypto ikev2 profile prof
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring key
 virtual-template 25
!
crypto ikev2 cts sgt
!
crypto ipsec transform-set trans esp-null esp-sha-hmac
!
crypto ipsec profile prof_ipv4
 set transform-set trans
 set ikev2-profile prof1_ipv4
!
!
interface Loopback0
 ip address 192.168.12.1 255.255.0.0
!
interface Loopback1
 no ip address
!
interface Loopback2
 ip address 172.18.0.1 255.240.0.0
!
```

```
interface Loopback10
 no ip address
 ipv6 address 2001::8:1/112
!
interface Loopback11
 no ip address
 ipv6 address 2001::80:1/112
!
interface Embedded-Service-Engine0/0
 no ip address
 shutdown
!
interface GigabitEthernet0/0
 ip address 10.1.1.2 255.0.0.0
 duplex auto
 speed auto
 ipv6 address 2001::7:1/112
 ipv6 enable
!
interface GigabitEthernet0/1
 ip address 10.10.10.2 255.255.255.0
 duplex auto
 speed auto
!
interface GigabitEthernet0/2
 ip address 192.168.210.144 255.255.255.0
 duplex auto
 speed auto
!
interface FastEthernet0/0/0
 no ip address
 shutdown
!
interface FastEthernet0/0/1
 no ip address
!
interface FastEthernet0/0/2
 no ip address
!
interface FastEthernet0/0/3
 no ip address
!
!
interface Virtual-Template25 type tunnel
 ip unnumbered GigabitEthernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile prof_ipv4
!
interface Vlan1
 no ip address
!
!
ip forward-protocol nd
!
no ip http server
no ip http secure-server
!
ip route 0.0.0.0 0.0.0.0 10.1.1.1
ip route 172.17.0.0 255.240.0.0 10.10.10.1
!
logging esm config
ipv6 route ::/0 2001::7:2
!
control-plane
```

```
!
!
!
line con 0
 exec-timeout 0 0
line aux 0
line 2
 no activation-character
 no exec
 transport preferred none
 transport input all
 transport output lat pad telnet rlogin lapb-ta mop udptn v120 ssh
 stopbits 1
line vty 0 4
 login
 transport input all
!
exception data-corruption buffer truncate
scheduler allocate 20000 1000
end
```

crypto ikev2 diagnose

To enable Internet Key Exchange Version 2 (IKEv2) error diagnostics, use the **crypto ikev2 diagnose** command in global configuration mode. To disable the error diagnostics, use the **no** form of this command.

crypto ikev2 diagnose error *number*
no crypto ikev2 diagnose error

Syntax Description

error	Enables the IKEv2 error path tracing.
number	Specifies the maximum number of errors allowed in the exit path entry. The range is 1 to 1000.

Command Default

IKEv2 error diagnostics is not enabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to enable IKEv2 error path tracing and to specify the number of entries in the exit path database. When the number exceeds the specified number, new entries replace the old entries.

Examples

The following example sets the maximum number of entries that can be logged:

```
Router(config)# crypto ikev2 diagnose error 500
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 dpd

To configure Dead Peer Detection (DPD) for Internet Key Exchange Version 2 (IKEv2), use the **crypto ikev2 dpd**command in global configuration mode. To delete DPD, use the **no** form of this command.

crypto ikev2 dpd *interval retry-interval* {**on-demand** | **periodic**}
no crypto ikev2 dpd

Syntax Description

interval	Specifies the keepalive interval in seconds.
retry-interval	Specifies the retry interval in seconds when there is no reply from the peer.
on-demand	Specifies the on-demand mode to send keepalive only in the absence of any incoming data traffic, to check the liveness of the peer before sending any data.
periodic	Specifies the periodic mode to send keepalives regularly at a specified interval.

Command Default

DPD is disabled by default.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to configure DPD globally for all peers. The DPD configuration in a Internet Key Exchange Version 2 (IKEv2) profile overrides the global DPD configuration.

Examples

The following example shows how to configure the periodic mode for DPD. In the example in this step, the first DPD is sent after 30 seconds when there is no incoming ESP traffic. After waiting for 6 seconds (which is the specified retry interval), DPD retries are sent agressively 5 times in intervals of 6 seconds each. So, a total of 66 seconds (30 + 6 + 6 * 5 = 66) elapses before a crypto session is torn down because of DPD.

```
Router(config)# crypto ikev2 dpd 30 6 on-demand
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.

Command	Description
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 fragmentation

To configure Internet Key Exchange Version 2 (IKEv2) fragmentation, use the **crypto ikev2 fragmentation** command in global configuration mode. To disable the fragmentation, use the **no** form of this command.

crypto ikev2 fragmentation mtu *mtu-size*
no crypto ikev2 fragmentation

Syntax Description

mtu *mtu-size*	Specifies the maximum transmission unit in bytes. The range is from 68 to 1500 bytes.

Command Default

IKEv2 fragmentation is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(3)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

Use this command to fragment the IKEv2 packets at IKEv2 layer and to avoid fragmentation after encryption. The MTU size refers to the IP or UDP encapsulated IKEv2 packets. The formula for fragmenting a packet is calculated as follows:

Specified MTU - UDP header - IP header = fragment packet size.

Using the above formula, if the MTU size is 100, specified in the command as **crypto ikev2 fragmentation mtu 100**, an IKE packet is fragmented if the packet size is greater than 72 bytes.

100 (specified MTU) - 8 (UDP header) - 20 (IP header) = 72 bytes.

Examples

The following example shows how to configure IKEv2 fragmentation:

```
Router# enable
Router(config)# crypto ikev2 fragmentation mtu 200
```

crypto ikev2 http-url

To enable lookup based on HTTP URL, use the **crypto ikev2 http-url**command in global configuration mode. To disable the lookup based on HTTP URL, use the **no** form of this command.

crypto ikev2 http-url cert
no crypto ikev2 http-url cert

Syntax Description

cert	Enable certificate lookup based on the HTTP URL.

Command Default

HTTP CERT is enabled by default.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1.(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to enable certificate lookup based on the HTTP URL. HTTP CERT indicates that the node is capable of looking up certificates based on the URL. This avoids the fragmentation that results when transferring large certificates.

Examples

The following example shows how to configure HTTP CERT:

```
Router(config)# crypto ikev2 http-url cert
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 keyring

To configure an Internet Key Exchange version 2 (IKEv2) key ring, use the **crypto ikev2 keyring** command in the global configuration mode. To delete an IKEv2 keyring, use the **no** form of this command.

crypto ikev2 keyring *keyring-name*
no crypto ikev2 keyring *keyring-name*

Syntax Description

keyring-name	Name of the keyring.

Command Default

There is no default key ring.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

IKEv2 keyrings are independent of IKEv1 keyrings. The key differences are as follows:

- IKEv2 keyrings support symmetric and asymmetric preshared keys.
- IKEv2 keyrings do not support Rivest, Shamir and Adleman (RSA) public keys.
- IKEv2 keyrings are specified in the IKEv2 profile and are not looked up, unlike IKEv1 where keys are looked up on receipt of MM1 to negotiate the preshared key authentication method. The authentication method is not negotiated in IKEv2.
- IKEv2 keyrings are not associated with VPN routing and forwarding (VRF) during configuration. The VRF of an IKEv2 keyring is the VRF of the IKEv2 profile that refers the keyring.
- A single keyring can be specified in an IKEv2 profile, unlike an IKEv1 profile, which can specify multiple keyrings.
- A single keyring can be specified in more than one IKEv2 profile, if the same keys are shared across peers matching different profiles.
- An IKEv2 keyring is structured as one or more peer subblocks.

On an IKEv2 initiator, IKEv2 keyring key lookup is performed using the peer’s hostname or the address, in that order. Use the hostname (ikev2 keyring) and address (ikev2 keyring) commands to configure the hostname and address in the IKEv2 keyring peer configuration mode.

On an IKEv2 responder, the key lookup is performed using the peer’s IKEv2 identity or the address, in that order. Use the address (ikev2 keyring) and **identity**(ikev2 keyring) command to configure the address and identity in IKEv2 keyring peer configuration mode.

Note You cannot configure the same identity in more than one peer.

The best match is only performed for address configurations and a key lookup is performed for the remaining peer identification, including identity address.

Examples

The following example shows how to configure a keyring:

```
Router(config)# crypto ikev2 keyring keyring-1
Router(config-ikev2-keyring)# peer peer1
Router(config-ikev2-keyring-peer)# description example.com
Router(config-ikev2-keyring-peer)# address 0.0.0.0 0.0.0.0

Router(config-ikev2-keyring-peer)# pre-shared-key xyz-key
```

The following example shows how a keyring match is performed. In the example, the key lookup for peer 10.0.0.1 would first match the wildcard key abc-key, then the prefix key abc-key and finally the host key host1-abc-key and the best match host1-abc-key is used.

```
Router(config)# crypto ikev2 keyring keyring-1
Router(config-ikev2-keyring)# peer peer1
Router(config-ikev2-keyring-peer)# description example.com
Router(config-ikev2-keyring-peer)# address 0.0.0.0 0.0.0.0

Router(config-ikev2-keyring-peer)# pre-shared-key xyz-key
Router(config-ikev2-keyring)# peer peer1
Router(config-ikev2-keyring-peer)# description abc.example.com
Router(config-ikev2-keyring-peer)# address 10.0.0.0 255.255.0.0

Router(config-ikev2-keyring-peer)# pre-shared-key abc-key
Router(config-ikev2-keyring)# peer host1
Router(config-ikev2-keyring-peer)# description host1@abc.example.com
Router(config-ikev2-keyring-peer)# address 10.0.0.1
Router(config-ikev2-keyring-peer)# pre-shared-key host1-abc-key
```

In the following example, the key lookup for peer 10.0.0.1 would first match the host key host1-abc-key. Because, this is a specific match, no further lookup is performed.

```
Router(config)# crypto ikev2 keyring keyring-2
Router(config-ikev2-keyring)# peer host1
Router(config-ikev2-keyring-peer)# description host1 in abc.example.com sub-domain
Router(config-ikev2-keyring-peer)# address 10.0.0.1
Router(config-ikev2-keyring-peer)# pre-shared-key host1-abc-key
Router(config-ikev2-keyring)# peer host2
Router(config-ikev2-keyring-peer)# description example domain
Router(config-ikev2-keyring-peer)# address 0.0.0.0 0.0.0.0
Router(config-ikev2-keyring-peer)# pre-shared-key xyz-key
```

Related Commands

Command	Description
address (ikev2 keyring)	Specifies the IPv4 address or the range of the peers in IKEv2 keyring.
description (ikev2 keyring)	Describes an IKEv2 peer or a peer group for the IKEv2 keyring.
hostname (ikev2 keyring)	Specifies the hostname for the peer in the IKEv2 keyring.

Command	Description
identity (ikev2 keyring)	Identifies the peer with IKEv2 types of identity.
peer	Defines a peer or a peer group for the keyring.
pre-shared-key **(ikev2 keyring)**	Defines a preshared key for the IKEv2 peer.

crypto ikev2 limit

To enable call admission control in Internet Key Exchange Version 2 (IKEv2), use the **crypto ikev2 limit** command in the global configuration mode. To disable call admission control, use the **no** form of this command.

crypto ikev2 limit {**max-in-negotiation-sa** *limit* [{**incoming** | **outgoing**}] | **max-sa** *limit* | **queue sa-init** *limit*}
no crypto ikev2 limit {**max-in-negotiation-sa** *limit* [{**incoming** | **outgoing**}] | **max-sa** *limit* | **queue sa-init**}

Syntax Description

max-in-negotiation-sa *limit*	Limits the total number of in-negotiation IKEv2 security associations (SAs) on the node.
incoming	(Optional) Limits the total number of in-negotiation IKEv2 SAs on the incoming node.
outgoing	(Optional) Limits the total number of in-negotiation IKEv2 SAs on the outgoing node.
max-sa *limit*	Limits the total number of IKEv2 SAs on the node.
queue sa-init *limit*	Limits the incoming SA_INIT requests size.

Command Default

By default, there is no configured limit on the number of IKEv2 SAs.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
Cisco IOS XE Release 3.12S	This command was modified. The **incoming** and **outgoing** keywords were added.
Cisco IOS XE Everest 16.5.1	This command was modified. The **queue sa-init** *limit* keyword-argument pair was added.
Cisco IOS XE Everest 16.6.1	The default **queue sa-init** *limit* of 5000 from being nvgen was stopped.

Usage Guidelines

Call admission control limits the in-negotiation and total number of IKEv2 SA on a node.

Note In IKEv2, rekey is not a new security association (SA) unlike in IKEv1. Hence, the rekey SA is not counted.

The **queue sa-init** *limit* keyword-argument pair limits the queue size to improve performance if you encounter packet drops from the initiating client due to response timeout. The packets are dropped when a source device sends IKEv2 INIT packets to a destination device to establish a tunnel, and the destination device is unable to process IKEv2 INIT packets faster due to a large queue of packets for processing on the responder device.

Examples

The following example shows how to enable call admission control:

```
Device(config)# crypto ikev2 max-in-negotiation-sa limit 5000
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 name mangler

To configure the Internet Key Exchange version 2 (IKEv2) name mangler, use the **crypto ikev2 name mangler** command in global configuration mode. To delete the name mangler, use the **no** form of this command.

crypto ikev2 name mangler *mangler-name*
no crypto ikev2 name mangler *mangler-name*

Syntax Description

mangler-name	IKEv2 mangler name.

Command Default

IKEv2 name mangler is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(3)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.

Usage Guidelines

The IKEv2 name mangler is used to derive a name for the AAA group or user authorization requests. The name mangler contains multiple statements--one for each identity type. The name mangler is derived from the specified portions of different forms of remote IKE identities or EAP identity. The name mangler is referred in the IKEv2 profile using the **aaa authorization** command.

After enabling this command, which puts the networking device in IKEv2 **name mangler configuration mode, you can specify the characteristics for the name mangler using the following commands:**

- **dn--** Derives the name from the remote identity of type distinguished name (DN)**.**
- **eap** --Derives the name from remote identities of type Extensible Authentication Protocol (EAP).
- **email** --Derives the name from the remote identity of type e-mail.
- **fqdn** --Derives the name from the remote identity of type Fully Qualified Domain Name (FQDN).

Examples

The following example shows how to define name manglers based on identity of type FQDN:

```
crypto ikev2 name-mangler mangler1
 fqdn domain
crypto ikev2 name-mangler mangler2
 fqdn hostname
crypto ikev2 name-mangler mangler3
 fqdn all
```

The following example shows how to define name manglers based on identity of type e-mail:

```
crypto ikev2 name-mangler mangler1
 email domain
crypto ikev2 name-mangler mangler2
```

```
 email username
crypto ikev2 name-mangler mangler3
 email all
```

The following example shows how to define name manglers based on identity of type DN:

```
crypto ikev2 name-mangler mangler2
  DN country
crypto ikev2 name-mangler mangler3
  DN state
crypto ikev2 name-mangler mangler4
  DN organization
crypto ikev2 name-mangler mangler5
  DN organization-unit
```

The following example shows how to define name manglers based on identity of type EAP:

```
crypto ikev2 name-mangler mangler1
  eap all
crypto ikev2 name-mangler mangler2
  eap prefix user123 delimiter @
crypto ikev2 name-mangler mangler3
  eap suffix cisco delimiter
crypto ikev2 name-mangler mangler4
 eap DN common-name
```

Related Commands

Command	Description
dn (IKEv2)	Derives the name from identity of type DN.
eap (IKEv2)	Derives the name from identity of type EAP.
email	Derives the name from identity of type e-mail.
fqdn	Derives the name from identity of type FQDN.

crypto ikev2 nat

To configure Network Address Translation (NAT) keepalive for Internet Key Exchange Version 2 (IKEv2), use the **crypto ikev2 nat**command in global configuration mode. To delete NAT keepalive configuration, use the **no** form of this command.

crypto ikev2 nat keepalive *interval*
no crypto ikev2 nat keepalive *interval*

Syntax Description

keepalive *interval*	Specifies the NAT keepalive interval in seconds.

Command Default

NAT keepalive is disabled by default.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Use this command to configure NAT keepalive globally for all peers. The NAT keepalive configuration specified in the IKEv2 profile overrides the global configuration. NAT keepalive prevents the deletion of NAT translation entries in the absence of any traffic, when NAT is between IKEv2 peers.

Examples

The following example shows how to specify a NAT keepalive interval of 500 seconds:

```
Router(config)# crypto ikev2 nat keepalive 500
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 window	Specifies the IKEv2 window size.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ikev2 policy

To configure an Internet Key Exchange Version 2 (IKEv2) policy, use the **crypto ikev2 policy** command in global configuration mode. To delete a policy, use the **no** form of this command. To return the policy to its default value, use the **default** form of this command.

crypto ikev2 policy *name*
no crypto ikev2 policy *name*
default crypto ikev2 policy

Syntax Description

name	Name of the IKEv2 policy.

Command Default

A default IKEv2 policy is used only in the absence of any user-defined IKEv2 policy. The default IKEv2 policy will have the default IKEv2 proposal and will match all local addresses in a global VPN Routing and Forwarding (VRF).

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

An IKEv2 policy contains the proposals that are used to negotiate the encryption, integrity, Psuedo-Random Function (PRF) algorithms and Diffie-Hellman (DH) group in SA_INIT exchange. IKEv2 policy can have match statements, which are used as selection criteria to select a policy for negotiation.

An IKEv2 policy must contain at least one proposal to be considered as complete, and can have more proposals and match statements.

A policy can have similar or different match statements. Match statements that are similar are logically ORed and match statements that are different are logically ANDed. There is no precedence between match statements of different types. Policy check will happen in a sequential order. To avoid unexpected or unpredictable behavior during IKEv2 policy selection, overlapping match statements must not be configured.

A policy is matched as follows:

- If no IKEv2 policy is configured, the default policy is used for negotiating a SA that uses any local address in a global VRF.
- If IKEv2 policies are configured, the policy with the best match is selected.
- If none of the configured policies matches, the SA_INIT exchange does not start.

You can modify the default policy using the **crypto ikev2 policy default** command. You can modify the entire policy or one of the statements in the policy.

You can disable the default policy using the **no crypto ikev2 policy default** command. When disabled, the values in the default policy are copied and the default policy remains inactive.

Examples

The following examples show how to configure a policy and how a policy match is performed:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal pro1
Router(config-ikev2-policy)# match fvrf green
Router(config-ikev2-policy)# match address local 10.0.0.1
```

The policy policy1 is selected and proposal pro1 is used for negotiating IKEv2 SA with the local address as 10.0.0.1 and the FVRF as green:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal pro1
Router(config-ikev2-policy)# match address local 10.0.0.1
```

The policy policy1 is selected and proposal pro1 is used for negotiation of the IKEv2 SA that is negotiatied with the local address as 10.0.0.1 and the FVRF as global:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal pro1
Router(config-ikev2-policy)# match fvrf green
```

The policy policy1 is selected and proposal pro1 is used for negotiation of the IKEv2 SA that is negotiatied with any local address and the FVRF as green.

How a Policy Match Is Performed

The following example shows how a policy is chosen out of two policies:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal1
Router(config-ikev2-policy)# match fvrf green
Router(config)# crypto ikev2 policy policy2
Router(config-ikev2-policy)# proposal1
Router(config-ikev2-policy)# match fvrf green
Router(config-ikev2-policy)# match local address 10.0.0.1
```

To negotiate the SA for local address 10.0.0.1 and FVRF as green, policy 2 is selected because policy 2 is the best match:

```
Router(config)# crypto ikev2 policy policy1
Router(config-ikev2-policy)# proposal2
Router(config-ikev2-policy)# match local address 10.0.0.1
Router(config-ikev2-policy)# match fvrf green
Router(config)# crypto ikev2 policy policy2
Router(config-ikev2-policy)# proposal1
Router(config-ikev2-policy)# match fvrf green
Router(config-ikev2-policy)# match local address 10.0.0.1
```

In this case, even though both the policies are the best match, policy1 is selected, because it was configured first.

Related Commands

Command	Description
match (ikev2 policy)	Matches an IKEv2 policy based on the parameters.
proposal	Specifies the proposals that must be used in the IKEv2 policy.

Command	Description
show crypto ikev2 policy	Displays the default or user-defined IKEv2 policy.

crypto ikev2 profile

To configure an Internet Key Exchange Version 2 (IKEv2) profile, use the **crypto ikev2 profile** command in global configuration mode. To delete the profile, use the **no** form of this command.

crypto ikev2 profile *profile-name*
no crypto ikev2 profile *profile-name* **dynamic**

Syntax Description

profile-name	The name of the IKEv2 profile.

Command Default

There is no default IKEv2 profile. However, there are default values for some commands under the profile, such as lifetime.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.
17.2.1	The dynamic keyword was introduced.

Usage Guidelines

Use this command to define an IKEv2 profile. An IKEv2 profile is a repository of the nonnegotiable parameters of the IKE security associations (SAs) (such as local/remote identities and authentication methods) and the services that will be available to the authenticated peers that match the profile. The following are the characteristics of an IKEv2 profile:

- It must be attached to either a crypto map or an IPsec profile on the IKEv2 initiator and responder.
- It must contain a match identity or match certificate statement; otherwise the profile is considered incomplete and is unused.
- The statements match VRF, local or remote authentication methods are optional.

The table below describes the differences between IKEv1 and IKEv2 profiles.

Table 1: Differences between IKEv1 and IKEv2 Profiles

IKEv1	IKEv2
The authentication method is a negotiable parameter and must be specified in the ISAKMP policy.	The authentication method is not a negotiable parameter, can be asymmetric, and must be specified in the profile.
Multiple keyrings can be specified in the profile.	A single keyring can be specified in the profile and is optional also.

The IKEv2 profile applied on the crypto interface must be the same as IKEv2 profile that matches the peer identity received in the IKE_AUTH exchange.

Examples

The following examples show an IKEv2 profile matched on a remote identity and an IKEv2 profile catering to two peers using different authentication method.

IKEv2 Profile Matched on Remote Identity

The following profile caters to peers that identify using fqdn example.com and authenticate with rsa-signature using trustpoint-remote. The local node authenticates with pre-share using keyring-1.

```
Router(config)# crypto ikev2 profile profile2
Router(config-ikev2-profile)# match identity remote fqdn example.com
Router(config-ikev2-profile)# identity local email router2@example.com
Router(config-ikev2-profile)# authentication local pre-share
Router(config-ikev2-profile)# authentication remote rsa-sig
Router(config-ikev2-profile)# keyring keyring-1
Router(config-ikev2-profile)# pki trustpoint trustpoint-remote verify
Router(config-ikev2-profile)# lifetime 300
Router(config-ikev2-profile)# dpd 5 10 on-demand
Router(config-ikev2-profile)# virtual-template 1
```

IKEv2 Profile Catering to Two Peers Using Different Authentication Method

The following profile caters to two peers: user1@example.com that authenticate with pre-share using keyring-1, and user2@example.com authenticates with rsa-signature using trustpoint-remote. However, the local peer authenticates the remote peers with rsa-signature using trustpoint-local.

```
Router(config)# crypto ikev2 profile profile2
Router(config-ikev2-profile)# match identity remote email user1@example.com
Router(config-ikev2-profile)# match identity remote email user2@example.com
Router(config-ikev2-profile)# identity local email router2@abc.com
Router(config-ikev2-profile)# authentication local rsa-sig
Router(config-ikev2-profile)# authentication remote pre-share
Router(config-ikev2-profile)# authentication remote rsa-sig
Router(config-ikev2-profile)# keyring keyring-1
Router(config-ikev2-profile)# pki trustpoint trustpoint-local sign
Router(config-ikev2-profile)# pki trustpoint trustpoint-remote verify
Router(config-ikev2-profile)# lifetime 300
Router(config-ikev2-profile)# dpd 5 10 on-demand
Router(config-ikev2-profile)# virtual-template 1
```

EAP Authentication with External EAP Server

The following example shows how to configure the remote access server using the remote EAP authentication method with an external EAP server:

```
aaa new-model
aaa authentication login aaa-eap-list default group radius
!
crypto ikev2 profile profile2
 authentication remote eap
 aaa authentication eap aaa-eap-list
```

EAP Authentication with Local and External EAP

The following example shows how to configure the remote access server with local and external EAP server using the remote EAP authentication method:

```
aaa new-model
aaa authentication login aaa-eap-list default group radius
aaa authentication login aaa-eap-local-list default group tacacs
!
crypto ikev2 profile profile2
 authentication remote eap
 authentication remote eap-local
 aaa authentication eap aaa-eap-list
 aaa authentication eap-local aaa-eap-local-list
```

Configuring the Local Policy

This example shows how to configure the AAA authorization for a local group policy:

```
aaa new-model
aaa authorization network aaa-group-list default local
!
 crypto ikev2 client configuration group cisco
  pool addr-pool1
  dns 198.51.100.1 198.51.100.100
  wins 203.0.113.1 203.0.113.115
 !
 crypto ikev2 profile profile1
  authentication remote eap
  aaa authorization group aaa-group-list abc
```

The aaa-group-list specifies that the group authorization is local and that the AAA username is abc. The authorization list name corresponds to the group policy defined in the **crypto ikev2 client configuration group** command.

External AAA-based Group Policy

This example shows how to configure an external AAA-based group policy. The aaa-group-list specifies that the group authorization is RADIUS based. The name mangler derives the group name from the domain part of ID-FQDN, which is abc.

```
 aaa new-model
 aaa authorization network aaa-group-list default group radius
 !
 crypto ikev2 name-mangler mangler1
  fqdn domain
  !
 crypto ikev2 profile profile1
  identity remote fqdn host1.abc
  authentication remote eap
  aaa authorization group aaa-group-list name-mangler mangler1
```

External AAA-based User Policy

This example shows how to configure an external AAA-based group policy. The aaa-user-list specifies that the user authorization is RADIUS based. The name mangler derives the username from the hostname part of ID-FQDN, which is host1.

```
aaa new-model
aaa authorization network aaa-user-list default group radius
!
crypto ikev2 name-mangler mangler2
 fqdn hostname
!
crypto ikev2 profile profile1
 match identity remote fqdn host1.abc
  authentication remote eap
 aaa authorization user aaa-user-list name-mangler mangler2
```

Related Commands

Command	Description
aaa authentication (IKEv2 profile)	Defines the AAA authentication list for EAP authentication.
aaa authorization (IKEv2 profile)	Defines the AAA authorization for a local or group policy.
authentication (IKEv2 profile)	Defines the local and remote authentication methods.
dynamic (IKEv2 profile)	Configures the IKEv2 profile settings to be dynamic.
crypto ikev2 keyring	Defines an IKEv2 keyring.
show crypto ikev2 profile	Displays the IKEv2 profile.

crypto ikev2 proposal

To configure an Internet Key Exchange Version 2 (IKEv2) proposal, use the **crypto ikev2 proposal** command in global configuration mode. To delete an IKEv2 proposal, use the **no** form of this command. To return the proposal to its default value, use the **default** form of this command.

crypto ikev2 proposal *name*
no crypto ikev2 proposal *name*
default crypto ikev2 proposal

Syntax Description

name	Name of the proposal. The proposals are attached to IKEv2 policies using the **proposal** command.

Command Default

The default IKEv2 proposal is used.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

An IKEv2 proposal is a set of transforms used in the negotiation of IKEv2 SA as part of the IKE_SA_INIT exchange. An IKEv2 proposal is regarded as complete only when it has at least an encryption algorithm, an integrity algorithm and a Diffie-Hellman (DH) group configured. If no proposal is configured and attached to an IKEv2 policy, then the default proposal is used in negotiation.

You can modify the default proposal using the **crypto ikev2 proposal default** command. You can modify the entire proposal or one of the transforms namely, the encryption algorithm, the integrity algorithm and the DH group.

You can disable the default proposal using the **no crypto ikev2 proposal default** command. When disabled, the values in the default proposal are copied and the default proposal remains inactive.

Although this command is similar to the **crypto isakmp policy** command, the IKEv2 proposal differs as follows:

- An IKEv2 proposal allows configuration of one or more transforms for each transform type.
- An IKEv2 proposal does not have any associated priority.

Note The IKEv2 proposals must be attached to the IKEv2 policies for using the proposals in negotiation. If a proposal is not configured, then the default IKEv2 proposal is used with the default IKEv2 policy.

When multiple transforms are configured for a transform type, the order of priority is from left to right.

A proposal with multiple transforms for each transform type translates to all possible combinations of transforms. If only a subset of these combinations is required, then they must be configured as individual proposals.

```
Router(config)# crypto ikev2 proposal proposal-1
Router(config-ikev2-proposal)# encryption aes-cbc-128, aes-cbc-192
Router(config-ikev2-proposal)# integrity sha, sha256
Router(config-ikev2-proposal)# group 14
```

For example, the commands shown translates to the following transform combinations:

```
aes-cbc-128, sha, 14
aes-cbc-192, sha, 14
aes-cbc-128, sha256, 14
aes-cbc-192, sha256, 14
```

To configure the first and last transform combinations, the commands are as follows:

```
Router(config)# crypto ikev2 proposal proposal-1
Router(config-ikev2-proposal)# encryption aes-cbc-128
Router(config-ikev2-proposal)# integrity sha
Router(config-ikev2-proposal)# group 14
Router(config)# crypto ikev2 proposal proposal-2
Router(config-ikev2-proposal)# encryption aes-cbc-192
Router(config-ikev2-proposal)# integrity sha256
Router(config-ikev2-proposal)# group 14
```

Examples

The following examples show how to configure a proposal:

IKEv2 Proposal with One Transform for Each Transform Type

```
Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128
Device(config-ikev2-proposal)# integrity sha1
Device(config-ikev2-proposal)# group 14
```

IKEv2 Proposal with Multiple Transforms for Each Transform Type

```
Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-192
Device(config-ikev2-proposal)# integrity sha2 sha256
Device(config-ikev2-proposal)# group 14 15
```

The IKEv2 proposal **proposal-2** shown translates to the following prioritized list of transform combinations:

- aes-cbc-128, sha1, 14
- aes-cbc-128, sha1, 15
- aes-cbc-128, sha256, 14
- aes-cbc-128, sha256, 15
- aes-cbc-192, sha1, 14
- aes-cbc-192, sha1, 15
- aes-cbc-192, sha256, 14
- aes-cbc-192, sha256, 15

IKEv2 Proposals on the Initiator and Responder

The proposal of the initiator is as follows:

```
Device(config)# crypto ikev2 proposal proposal-1
Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-196
Device(config-ikev2-proposal)# integrity sha1 sha256
Device(config-ikev2-proposal)# group 14 16
```

The proposal of the responder is as follows:

```
Device(config)# crypto ikev2 proposal proposal-2
Device(config-ikev2-proposal)# encryption aes-cbc-196 aes-cbc-128
Device(config-ikev2-proposal)# integrity sha256 sha1
Device(config-ikev2-proposal)# group 16 14
```

In the scenario shown, the initiator choice of algorithms is preferred and the selected algorithms are as follows:

```
encryption aes-cbc-128
integrity sha1
group 14
```

Related Commands

Command	Description
encryption (ikev2 proposal)	Specifies the encryption algorithm in an IKEv2 proposal.
group (ikev2 proposal)	Specifies the Diffie-Hellman group identifier in an IKEv2 proposal.
integrity (ikev2 proposal)	Specifies the integrity algorithm in an IKEv2 proposal.
show crypto ikev2 proposal	Displays the parameters for each IKEv2 proposal.

crypto ikev2 redirect

To configure an Internet Key Exchange Version 2 (IKEv2) redirect mechanism on a gateway and a client, use the **crypto ikev2 redirect** command in global configuration mode. To remove the redirect mechanism, use the **no** form of this command.

crypto ikev2 redirect {**client** [{**max-redirects** *number*}] | **gateway** {**auth** | **init**}}
no crypto ikev2 redirect {**client** | **gateway**}

Syntax Description

client	Enables the redirect mechanism on a FlexVPN client.
max-redirects *number*	(Optional) Specifies the maximum number of redirects that can be configured per session on a FlexVPN client for redirect loop detection. The range is from 1 to 255. The default is 5.
gateway	Enables the redirect mechanism on a gateway.
auth	Enables the redirects mechanism on a gateway when a security association (SA) is authenticated.
init	Enables the redirect mechanism on a gateway when an SA is initiated.

Command Default

The redirects mechanism is disabled (on a gateway and a client).

Command Modes

Global configuration (config)

Command History

Release	Modification
15.2(4)M	This command was introduced.

Usage Guidelines

A thorough security analysis shows that redirect during IKE_AUTH is neither more nor less secure than redirect during IKE_INIT. However, for performance and scalability reasons, we recommend redirect during IKE_INIT.

Examples

The following example shows how to enable the redirects mechanism on the client and the gateway during initiation:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 redirect client
Device(config)# crypto ikev2 redirect gateway init
```

Related Commands

Command	Description
crypto ikev2 cluster	Defines an IKEv2 cluster policy in an HSRP cluster.

crypto ikev2 window

To configure the Internet Key Exchange Version 2 (IKEv2) window size, use the **crypto ikev2 window**command in global configuration mode. To delete IKEv2 window configuration, use the **no** form of this command.

crypto ikev2 window *window-size*
no crypto ikev2 window

Syntax Description

window-size	Size of the window that can range from 1 to 20.

Command Default

The default window size is 5.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(1)T	This command was introduced.
Cisco IOS XE Release 3.3S	This command was integrated into Cisco IOS XE Release 3.3S.
15.2(4)S	This command was integrated into Cisco IOS Release 15.2(4)S.

Usage Guidelines

Window size allows multiple IKEv2 request-response pairs in transit. Use this command to specify the IKEv2 window size to have multiple IKEv2 request-response pairs in transit.

Examples

The following example shows how to configure a window size of 10:

```
Router(config)# crypto ikev2 window size 10
```

Related Commands

Command	Description
crypto ikev2 certificate-cache	Specifies the cache size to store certificates fetched from HTTP URLs.
crypto ikev2 cookie-challenge	Enables IKEv2 cookie challenge.
crypto ikev2 diagnose error	Enables IKEv2 error diagnosis.
crypto ikev2 dpd	Defines DPD globally for all peers.
crypto ikev2 http-url cert	Enables HTTP CERT support.
crypto ikev2 limit	Defines call admission control for all peers.
crypto ikev2 nat	Defines NAT keepalive globally for all peers.
crypto logging ikev2	Enables IKEv2 syslog messages on a server.

crypto ipsec client ezvpn (global)

To create a Cisco Easy VPN remote configuration and enter the Cisco Easy VPN remote configuration mode, use the **crypto ipsec client ezvpn** command in global configuration mode. To delete the Cisco Easy VPN remote configuration, use the **no** form of this command.

crypto ipsec client ezvpn *name*
no crypto ipsec client ezvpn *name*

Note A separate **crypto ipsec client ezvpn** command in interface configuration mode assigns a Cisco Easy VPN remote configuration to the interface.

Note For network extension mode, the dynamic NAT rule is not inserted by EZVPN client when a duplicate split tunnel (ACE has same source address but different destination address) entry is pushed from EZVPN server for network extension mode.

Syntax Description

name	Identifies the Cisco Easy VPN remote configuration with a unique, arbitrary name.

Command Default

Newly created Cisco Easy VPN remote configurations default to **client** mode.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(4)YA	This command was introduced for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(8)YJ	This command was enhanced to enable you to manually establish and terminate an IPsec VPN tunnel on demand for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(15)T	This command was integrated into Cisco IOS Release 12.2(15)T.
12.3(4)T	The **username** command was added, and the **pee**command was changed so that the command may now be input multiple times.
12.3(7)XR	The **acl** and **backup** commands were added.
12.2(18)SXD	This command was integrated into Cisco IOS Release 12.2(18)SXD.
12.3(11)T	The **acl** command was integrated into Cisco IOS Release 12.3(11)T. However, the **backup** command was not integrated into Cisco IOS Release 12.3(11)T.

Release	Modification
12.4(2)T	The **virtual-interface** command was added.
12.4(4)T	The **default** keyword was added to the **peer** command, and the **flow allow acl** and **idle-time** commands were added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.4(11)T	The **nat acl** and **nat allow** commands were added.

Usage Guidelines

The **crypto ipsec client ezvpn** command creates a Cisco Easy VPN remote configuration and then enters the Cisco Easy VPN remote configuration mode, at which point you can enter the following commands:

- **acl** {*acl-name* | *acl-number*}--Specifies multiple subnets in a Virtual Private Network (VPN) tunnel. Up to 50 subnets may be configured.
    - The *acl-name* argument is the name of the access control list (ACL).
    - The *acl-number* argument is the number of the ACL.

Note Use the **acl** command in the Network Extension Mode (NEM) to expand the networks that are being extended. The **permit** statements in the ACL allow you to add additional networks to the list of extended networks. Without an ACL, the VPN only provides connectivity with the directly connected network of the inside interface.

- **backup** {*ezvpn-config-name*} **track** {*tracked-object-number*}--Specifies the Easy VPN configuration that will be activated when the backup is triggered.
    - **backup** {*ezvpn-config-name*}--Specifies the Easy VPN configuration that will be activated when the backup is triggered.
    - **track** {*tracked-object-number*}--Specifies the link to the tracking system so that the Easy VPN state machine can get the notification to trigger the backup.
- **connect** [**auto** | **manual | acl**]--Manually establishes and terminates an IP Security (IPsec) tunnel on demand.
    - The **auto** keyword is the default setting, because it was the initial Cisco Easy VPN remote functionality. The IPsec VPN tunnel is automatically connected when the Cisco Easy VPN Remote feature is configured on an interface.
    - The **manual** keyword specifies the manual setting to direct the Cisco Easy VPN remote to wait for a command or application programming interface (API) call before attempting to establish the Cisco Easy VPN remote connection. When the tunnel times out or fails, subsequent connections have to wait for the command to reset to manual or to an API call.
    - The **acl** keyword specifies the ACL-triggered setting, which is used for transactional-based applications and dial backup. Using this option, you can define the “interesting” traffic that triggers the tunnel to be established.
- **default**--Sets the following command to its default values.
- **exit**--Exits the Cisco Easy VPN configuration mode and returns to global configuration mode.

- **flow allow acl** [*name* | *number*]--Restricts the client from sending traffic in clear text when the tunnel is down. The *name* argument is the access list name. The *number* argument is the access list number, which can be 100 through 199.
- **flow restrict**—Restricts the traffic coming from Cisco Easy VPN inside interface to goout in clear text when a VPN tunnel is down.
- **group** *group-name* **key** *group-key*--Specifies the group name and key value for the VPN connection.
- **idletime**--(Optional) Sets the idle time after which an Easy VPN tunnel is brought down.
- **local-address** *interface-name*--Informs the Cisco Easy VPN remote which interface is used to determine the public IP address, which is used to source the tunnel. This command applies only to the Cisco uBR905 and Cisco uBR925 cable access routers.
  - The value of the *interface-name* argument specifies the interface used for tunnel traffic.

After specifying the local address used to source tunnel traffic, the IP address can be obtained in two ways:

- - The **local-address** command can be used with the **cable-modem dhcp-proxy** {**interface loopback** *number} command to obtain a public IP address and automatically assign it to the loopback interface.*
  - The IP address can be manually assigned to the loopback interface.
- **mode** {**client** | **network-extension** | **network extension plus**}--Specifies the VPN mode of operation of the router:
  - The **client** keyword (default) automatically configures the router for Cisco Easy VPN client mode operation, which uses Network Address Translation (NAT) or Peer Address Translation (PAT) address translations. When the Cisco Easy VPN remote configuration is assigned to an interface, the router automatically creates the NAT or PAT and access list configuration needed for the VPN connection.
  - The **network-extension** keyword specifies that the router should become a remote extension of the enterprise network at the other end of the VPN connection. The PCs that are connected to the router typically are assigned an IP address in the address space of the enterprise network.
  - The **network extension plus** keyword is identical to network extension mode with the additional capability of being able to request an IP address via mode configuration and automatically assign it to an available loopback interface. The IPsec security associations (SAs) for this IP address are automatically created by Easy VPN Remote. The IP address is typically used for troubleshooting (using ping, Telnet, and Secure Shell).
- nat acl {*acl-name* | *acl-number*}--Enables split-tunneling for the traffic specified by the ACL name or the ACL number.
  - The *acl-name* argument is the name of the ACL.
  - The *acl-number* argument is the number of the ACL.
- nat allow--Allows NAT to be integrated with Cisco Easy VPN.
- **no**--Removes the command or sets it to its default values.
- **peer** {*ipaddress* | *hostname* } [**default**]--Sets the peer IP address or hostname for the VPN connection. A hostname can be specified only when the router has a Domain Name System (DNS) server available for hostname resolution.

The **peer** command may be input multiple times.

The **default** keyword defines the given peer as the primary peer. When Phase 1 SA negotiations fail and Easy VPN fails over from the primary peer to the next peer on its backup list and the primary peer is again available, the current connection is torn down and the primary peer is reconnected.

- **username** *name* **password** {**0** | **6**} {*password*}--Allows you to save your extended authentication (Xauth) password locally on the PC. On subsequent authentications, you may activate the save-password tick box on the software client or add the username and password to the Cisco IOS hardware client profile. The setting remains until the save-password attribute is removed from the server group profile.
    - **0** specifies that an unencrypted password will follow.
    - **6** specifies that an encrypted password will follow.
    - *password specifies an unencrypted (cleartext) user password.*

The save-password option is useful only if the user password is static, that is, it is not a one-time password (OTP), such as a password generated by a token.

- **virtual-interface** [*virtual-template-number*]--Specifies a virtual interface for an Easy VPN remote device. If a virtual template number is specified, the virtual interface is derived from the virtual template that is configured. If a virtual template number is not specified, a generic virtual-access interface of the type tunnel is created. If the creation is successful, Easy VPN makes the virtual-access interface its outside interface (that is, the crypto map and NAT are applied on the virtual-access interface). If the creation is a failure, Easy VPN prints an error message and remains in the IDLE state.

After configuring the Cisco Easy VPN remote configuration, use the **exit** command to exit the Cisco Easy VPN remote configuration mode and return to global configuration mode.

Note You cannot use the **no crypto ipsec client ezvpn** command to delete a Cisco Easy VPN remote configuration that is assigned to an interface. You must remove that Cisco Easy VPN remote configuration from the interface before you can delete the configuration.

Examples

The following example shows a Cisco Easy VPN remote configuration named “telecommuter-client” being created on a Cisco uBR905 or Cisco uBR925 cable access router and being assigned to cable interface 0:

```
Router# configure terminal

Router(config)# crypto ipsec client ezvpn telecommuter-client

Router(config-crypto-ezvpn)# group telecommute-group
 key secret-telecommute-key

Router(config-crypto-ezvpn)# peer telecommuter-server

Router(config-crypto-ezvpn)# mode client

Router(config-crypto-ezvpn)# exit

Router(config)# interface c0

Router(config-if)# crypto ezvpn telecommuter-client

Router(config-if)# exit
```

Note Specifying the **mode client** option as shown above is optional because this is a default configuration for these options.

The following example shows the Cisco Easy VPN remote configuration named “telecommuter-client” being removed from the interface and then deleted:

```
Router# configure terminal

Router(config)# interface e1

Router(config-if)# no crypto ipsec client ezvpn telecommuter-client

Router(config-if)# exit

Router(config)# no crypto ipsec client ezvpn telecommuter-client
```

The following example shows that a virtual IPsec interface has been configured for the Easy VPN remote device:

```
crypto ipsec client ezvpn EasyVPN1
 virtual-interface 3
```

Related Commands

Command	Description
crypto ipsec client ezvpn (interface)	Assigns a Cisco Easy VPN remote configuration to an interface.

crypto ipsec client ezvpn (interface)

To assign a Cisco Easy Virtual Private Network (VPN) remote configuration to an interface other than a virtual interface, to specify whether the interface is outside or inside, and to configure multiple outside and inside interfaces, use the **crypto ipsec client ezvpn** command in interface configuration mode. To remove the Cisco Easy VPN remote configuration from the interface, use the **no** form of this command.

crypto ipsec client ezvpn *name* [{**outside** | **inside**}]
no crypto ipsec client ezvpn *name* [{**outside** | **inside**}]

Syntax Description

name	Specifies the Cisco Easy VPN remote configuration to be assigned to the interface. **Note** The interface specified cannot be a virtual interface.
outside	(Optional) Specifies the outside interface of the IP Security (IPsec) client router. You can add up to four outside tunnels for all platforms, one tunnel per outside interface.
inside	(Optional) Specifies the inside interface of the IPsec client router. The Cisco 1700 series has no default inside interface, and any inside interface must be configured. The Cisco 800 series routers and Cisco uBR905 and Cisco uBR925 cable access routers have default inside interfaces. However, you can configure any inside interface and add up to three inside interfaces for all platforms.

Command Default

The default inside interface is the Ethernet interface on Cisco 800 series routers and Cisco uBR905 and Cisco uBR925 cable access routers.

Command Modes

Interface configuration (config-if)

Command History

Release	Modification
12.2(4)YA	This command was introduced on Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(8)YJ	This command was enhanced to enable you to configure multiple outside and inside interfaces for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
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.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelines

The **crypto ipsec client ezvpn** command assigns a Cisco Easy VPN remote configuration to an interface, enabling the creation of a VPN connection over that interface to the specified VPN peer. If the Cisco Easy VPN remote configuration is configured for the client mode of operation, the router is also automatically configured for network address translation (NAT) or port address translation (PAT) and for an associated access list.

Note The **crypto ipsec client ezvpn** command is not supported on virtual interfaces.

Release 12.2(8)YJ

The **crypto ipsec client ezvpn** command was enhanced to allow you to configure multiple outside and inside interfaces. To configure multiple outside and inside interfaces, you must use the **interface** *interface-name* command to first define the type of interface on the IPsec client router.

- In client mode for the Cisco Easy VPN client, a single security association (SA) connection is used for encrypting and decrypting the traffic coming from all the inside interfaces. In network extension mode, one SA connection is established for each inside interface.
- When a new inside interface is added or an existing one is removed, all established SA connections are deleted and new ones are initiated.
- Configuration information for the default inside interface is shown with the **crypto ipsec client ezvpn name inside** command. All inside interfaces, whether they belong to a tunnel, are listed in interface configuration mode as an inside interface, along with the tunnel name.

Release 12.2(4)YA

The following restrictions apply to the **crypto ipsec client ezvpn** command:

- The Cisco Easy VPN remote feature supports only one tunnel, so the **crypto ipsec client ezvpn** command can be assigned to only one interface. If you attempt to assign it to more than one interface, an error message is displayed. You must use the **no** form of this command to remove the configuration from the first interface before assigning it to the second interface.
- The **crypto ipsec client ezvpn** command should be assigned to the outside interface of the NAT or PAT. This command cannot be used on the inside NAT or PAT interface. On some platforms, the inside and outside interfaces are fixed.

For example, on Cisco uBR905 and Cisco uBR925 cable access routers, the outside interface is always the cable interface. On Cisco 1700 series routers, the FastEthernet interface defaults to being the inside interface, so attempting to use the **crypto ipsec client ezvpn** command on the FastEthernet interface displays an error message.

Note A separate **crypto ipsec client ezvpn** command exists in global configuration mode that creates a Cisco Easy VPN remote configuration. You must first use the global configuration version of the **crypto ipsec client ezvpn** command to create a Cisco Easy VPN remote configuration before assigning it to an interface.

Examples

The following example shows a Cisco Easy VPN remote configuration named “ telecommuter-client” being assigned to the cable interface on a Cisco uBR905 or a Cisco uBR925 cable access router:

```
Router# configure terminal
Router(config)# interface c0
Router(config-if)# crypto ipsec client ezvpn telecommuter-client

Router(config-if)# exit
```

The fo llowing example first shows an attempt to delete the Cisco Easy VPN remote configuration named “telecommuter-client, ” but the configuration cannot be deleted because it is still assigned to an interface. The configuration is then removed from the interface and deleted.

```
Router# configure terminal

Router(config)# no crypto ipsec client ezvpn telecommuter-client
Error: crypto map in use by interface; cannot delete
Router(config)# interface e1

Router(config-if)# no crypto ipsec client ezvpn telecommuter-client

Router(config-if)# exit

Router(config)# no crypto ipsec client ezvpn telecommuter-client
```

Related Commands

Command	Description
crypto ipsec client ezvpn (global)	Creates and modifies a Cisco Easy VPN remote configuration.
interface	Configures an interface type.

crypto ipsec client ezvpn connect

To connect to a specified IPSec Virtual Private Network (VPN) tunnel in a manual configuration, use the **crypto ipsec client ezvpn connect** command in privileged EXEC mode. To disable the connection, use the **no** form of this command.

crypto ipsec client ezvpn connect *name*
no crypto ipsec client ezvpn connect *name*

Syntax Description

name	Identifies the IPSec VPN tunnel with a unique, arbitrary name.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(8)YJ	This command was introduced on Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
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.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelines

This command is used with the **connect** [**auto** | **manual** | **acl**] subcommand. After the manual setting is designated, the Cisco Easy VPN remote waits for a command or application programming interface (API) call before attempting to establish the Cisco Easy VPN remote connection.

If the configuration is manual, the tunnel is connected only after the **crypto ipsec client ezvpn connect** name command is entered in privileged EXEC mode, and after the **connect** [**auto**] | **manual** subcommand is entered.

Examples

The following example shows how to connect an IPSec VPN tunnel named ISP-tunnel on a Cisco uBR905/uBR925 cable access router:

```
Router# crypto ipsec client ezvpn connect
 ISP-tunnel
```

Related Commands

Command	Description
connect	Manually establishes and terminates an IPSec VPN tunnel on demand.
crypto ipsec client ezvpn (global)	Creates and modifies a Cisco Easy VPN remote configuration.

crypto ipsec client ezvpn xauth

To respond to a pending Virtual Private Network (VPN) authorization request, use the **crypto ipsec client ezvpn xauth** command in privileged EXEC mode.

crypto ipsec client ezvpn xauth *name*

Syntax Description

name	Identifies the IP Security (IPSec) VPN tunnel with a unique, arbitrary name. This name is required.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.2(4)YA	This command was introduced on Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(8)YJ	This command was enhanced to specify an IPSec VPN tunnel for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
12.2(8)YJ	This command was enhanced to specify an IPSec VPN tunnel for Cisco 806, Cisco 826, Cisco 827, and Cisco 828 routers; Cisco 1700 series routers; and Cisco uBR905 and Cisco uBR925 cable access routers.
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.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelines

If the tunnel name is not specified, the authorization request is made on the active tunnel. If there is more than one active tunnel, the command fails with an error requesting that you specify the tunnel name.

When making a VPN connection, individual users might also be required to provide authorization information, such as a username or password. When the remote end requires this information, the router displays a message on the console of the router instructing the user to enter the **crypto ipsec client ezvpn xauth**command. The user then uses command-line interface (CLI) to enter this command and to provide the information requested by the prompts that follow after the command has been entered.

Note If the user does not respond to the authentication notification, the message is repeated every 10 seconds.

Examples

The following example shows an example of the user being prompted to enter the **crypto ipsec client ezvpn xauth** command. The user then enters the requested information and continues.

```
Router#
```

```
20:27:39: EZVPN: Pending XAuth Request, Please enter the following command:
20:27:39: EZVPN: crypto ipsec client ezvpn xauth
Router> crypto ipsec client ezvpn xauth
Enter Username and Password: userid
Password: ************
```

Related Commands

Command	Description
crypto ipsec client ezvpn (interface)	Assigns a Cisco Easy VPN Remote configuration to an interface.

crypto ipsec transform-set default

To enable default IP Security (IPsec) transform sets, use the **crypto ipsec transform-set default** command in global configuration mode. To disable the default IPsec transform sets, use the **no** form of this command.

crypto ipsec transform-setdefault
no crypto ipsec transform-setdefault

Syntax Description

This command has no arguments or keywords.

Command Default

The default IPsec transform sets are enabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.4(20)T	This command was introduced.
Cisco IOS XE Release 2.4	This command was implemented on the Cisco ASR 1000 series routers.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

A default transform set will be used by any crypto map or ipsec profile where no other transform set has been configured if the following is true:

- The default transform sets have not been disabled with the **no crypto ipsec default transform-set** command.
- The crypto engine in use supports the encryption algorithm.

Each default transform set defines both an Encapsulation Security Protocol (ESP) encryption transform type and an ESP authentication transform type as shown in the table below.

Table 2: Default Transform Sets and Parameters

Default Transform Name	ESP Encryption Transform and Description	ESP Authentication Transform and Description
#$!default_transform_set_0	esp-3des (ESP with the 168-bit Triple Data Encryption Standard [3DES or Triple DES] encryption algorithm)	esp-sha-hmac (ESP with the Secure Hash Algorithm [SHA-1, HMAC variant] authentication algorithm)

Default Transform Name	ESP Encryption Transform and Description	ESP Authentication Transform and Description
#$!default_transform_set_1	esp-aes (ESP with the 128-bit Advanced Encryption Standard [AES] encryption algorithm)	esp-sha-hmac

Examples

```
The following example displays output from the show crypto ipsec transform-set default
 command when the default transform sets are enabled, the default setting.
Router# show crypto ipsec transform-set default

Transform set #$!default_transform_set_1: { esp-aes esp-sha-hmac  }
   will negotiate = { Transport,  },

Transform set #$!default_transform_set_0: { esp-3des esp-sha-hmac  }
   will negotiate = { Transport,  },
The following example displays output from the show crypto ipsec transform-set default
 command when the default transform sets have been disabled with the no crypto ipsec default
 transform-set
command.
Router(config)# no crypto ipsec default transform-set
Router(config)# exit
Router#
Router# show crypto ipsec transform-set default
! There is no output.
Router#
```

The following is example system log message that is generated whenever IPsec security associations (SAs) have negotiated with a default transform set.

```
%CRYPTO-5-IPSEC_DEFAULT_TRANSFORM: Using Default IPSec transform-set
```

Related Commands

Command	Description
show crypto isakmp default policy	Displays the default IKE policies currently in use.

crypto ipsec df-bit (global)

To set the DF bit for the encapsulating header in tunnel mode to all interfaces, use the **crypto ipsec df-bit** command in global configuration mode.

crypto ipsec df-bit [{**clear** | **set** | **copy**}]

Syntax Description

clear	Outer IP header will have the DF bit cleared, and the router may fragment the packet to add the IP Security (IPSec) encapsulation.
set	Outer IP header will have the DF bit set; however, the router may fragment the packet if the original packet had the DF bit cleared.
copy	The router will look in the original packet for the outer DF bit setting. The **copy** keyword is the default setting.

Command Default

The default is **copy**

Command Modes

Global configuration

Command History

Release	Modification
12.2(2)T	This command was introduced.

Usage Guidelines

Use the **crypto ipsec df-bit** command in global configuration mode to configure your router to specify the DF bit in an encapsulated header.

You may want use the **clear** setting for the DF bit when encapsulating tunnel mode IPSec traffic so you can send packets larger than the available maximum transmission unit (MTU) size or if you do not know what the available MTU size is.

Note In Cisco IOS Release 15.2(1)T and later releases, either wait till the next rekey/SA installation or enter the **clear crypto session** command for the **crypto ipsec df-bit clear** command to take effect.

If this command is enabled without a specified setting, the router will use the **copy** setting as the default.

Examples

The following example shows how to clear the DF bit on all interfaces:

```
crypto ipsec df-bit clear
```

crypto ipsec df-bit (interface)

To set the DF bit for the encapsulating header in tunnel mode to a specific interface, use the **crypto ipsec df-bit** command in interface configuration mode.

crypto ipsec df-bit [{**clear** | **set** | **copy**}]

Syntax Description

clear	Outer IP header has the DF bit cleared, and the router may fragment the packet to add the IP Security (IPSec) encapsulation.
set	Outer IP header has the DF bit set; however, the router may fragment the packet if the original packet had the DF bit cleared.
copy	The router looks in the original packet for the outer DF bit setting.

Command Default

The default setting is the same as the **crypto ipsec df-bit** command setting in global configuration mode.

Command Modes

Interface configuration

Command History

Release	Modification
12.2(2)T	This command was introduced.

Usage Guidelines

Use the **crypto ipsec df-bit** command in interface configuration mode to configure your router to specify the DF bit in an encapsulated header. This command overrides any existing DF bit global settings.

You may want use the **clear** setting for the DF bit when encapsulating tunnel mode IPSec traffic so you can send packets larger than the available maximum transmission unit (MTU) size or if you do not know what the available MTU size is.

Note In Cisco IOS Release 15.2(1)T and later releases, either wait till the next rekey/SA installation or enter the **clear crypto session** command for the **crypto ipsec df-bit clear** command to take effect.

If this command is enabled without a specified setting, the router uses the **crypto ipsec df-bit** command setting in global configuration mode.

Examples

In following example, the router is configured to globally clear the setting for the DF bit and copy the DF bit on the interface named Ethernet0. Thus, all interfaces except Ethernet0 allows the router to send packets larger than the available MTU size; Ethernet0 allows the router to fragment the packet.

```
crypto isakmp policy 1
 encr aes
 hash sha
 authentication pre-share
 group 14
crypto isakmp key Delaware address 192.168.10.66
crypto isakmp key Key-What-Key address 192.168.11.19
!
!
crypto ipsec transform-set BearMama ah-sha-hmac esp-aes
crypto ipsec df-bit clear
!
!
crypto map armadillo 1 ipsec-isakmp
set peer 192.168.10.66
set transform-set BearMama
match address 101
!
crypto map basilisk 1 ipsec-isakmp
set peer 192.168.11.19
set transform-set BearMama
match address 102
!
!
interface Ethernet0
 ip address 192.168.10.38 255.255.255.0
 ip broadcast-address 0.0.0.0
 media-type 10BaseT
 crypto map armadillo
 crypto ipsec df-bit copy
!
interface Ethernet1
 ip address 192.168.11.75 255.255.255.0
 ip broadcast-address 0.0.0.0
 media-type 10BaseT
 crypto map basilisk
!
interface Serial0
 no ip address
 ip broadcast-address 0.0.0.0
 no ip route-cache
 no ip mroute-cache
```

crypto ipsec fragmentation (global)

To enable prefragmentation for IP Security (IPSec) Virtual Private Networks (VPNs) on a global basis, use the **crypto ipsec fragmentation**command in global configuration mode. To disable a manually configured command, use the **no** form of this command.

crypto ipsec fragmentation {**before-encryption** | **after-encryption**}
no crypto ipsec fragmentation {**before-encryption** | **after-encryption**}

Syntax Description

before-encryption	Enables prefragmentation for IPSec VPNs. The default is that prefragmentation is enabled.
after-encryption	Disables prefragmentation for IPSec VPNs.

Command Default

If you do not enter this command, prefragmentation is enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.1(11b)E	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.

Usage Guidelines

Use the **before-encryption**keywordto enable prefragmentation for IPSec VPNs; use the **after-encryption** keyword to disable prefragmentation for IPSec VPNs. This command allows an encrypting router to predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPSec security association (SA). If it is predetermined that the packet will exceed the maximum transmission unit (MTU) of the output interface, the packet is fragmented before encryption.

Note This command does not show up in the a running configuration if the default global command is enabled. It shows in the running configuration only when you explicitly enable the command on an interface.

Examples

The following example shows how to globally enable prefragmentation for IPSec VPNs:

```
crypto ipsec fragmentation before-encryption
```

crypto ipsec fragmentation (interface)

To enable prefragmentation for IP Security (IPSec) Virtual Private Networks (VPNs) on an interface, use the **crypto ipsec fragmentation**command in interface configuration mode. To disable a manually configured command, use the **no** form of this command.

crypto ipsec fragmentation {**before-encryption** | **after-encryption**}
no crypto ipsec fragmentation {**before-encryption** | **after-encryption**}

Syntax Description

before-encryption	Enables prefragmentation for IPSec VPNs.
after-encryption	Disables prefragmentation for IPSec VPNs.

Command Default

If no other prefragmentation for IPSec VPNs commands are in the configuration, the router will revert to the default global configuration.

Command Modes

Interface configuration

Command History

Release	Modification
12.1(11b)E	This command was introduced.
12.2(13)T	This command was integrated into Cisco IOS Release 12.2(13)T.
12.2(14)S	This command was integrated into Cisco IOS Release 12.2(14)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use the **before-encryption** keyword to enable prefragmentation for IPSec VPNs per interface; use the **after-encryption** keyword to disable prefragmentation for IPSec VPNs. This command allows an encrypting router to predetermine the encapsulated packet size from information available in transform sets, which are configured as part of the IPSec security association (SA). If it is predetermined that the packet will exceed the maximum transmission unit (MTU) of output interface, the packet is fragmented before encryption.

Examples

The following example shows how to enable prefragmentation for IPSec VPNs on an interface and then how to display the output of the show running configuration command:

Note This command shows in the running configuration only when you explicitly enable it on the interface.

```
Router(config-if)# crypto ipsec fragmentation before-encryption
Router(config-if)# exit
Router# show running-config
crypto isakmp policy 10
```

```
 encryption aes
 authentication pre-share
 group 14
crypto isakmp key abcd123 address 209.165.202.130
!
crypto ipsec transform-set fooprime esp-aes esp-sha-hmac
!
crypto map bar 10 ipsec-isakmp
 set peer 209.165.202.130
 set transform-set fooprime
 match address 102
```

crypto ipsec ike sa-strength-enforcement

To ensure that the strength of the IKE encryption cipher is higher than the strength of its child IPsec SA's encryption ciphers, use the **crypto ipsec ike sa-strength-enforcement** command. To disable this feature, use the **no** form of this command.

crypto ipsec ike sa-strength-enforcement
no crypto ipsec ike sa-strength-enforcement

Command Default

Enforcement is disabled by default.

Command Modes

Global configuration (config)

Command History

Release	Modification
Cisco IOS XE Release 17.13	This command was introduced.

Usage Guidelines

It is a good security practice to configure IPSec such that the strength of the IKE SA encryption cipher is greater than or equal to the strength of its child IPsec SA encryption cipher. The strength enforcement only affects the encryption cipher. It does not alter the integrity or key exchange algorithms. The encryption cipher strength comparison is done during session negotiation or establishment. It is not enforced during the configuration of IKE or IPsec. The number of bits in the encryption key determines the strength of the encryption cipher.

When this command is enabled, the IKEv1 and IKEv2 sessions compare the relative strength of each child SA's selected encryption cipher. If the child SA's encryption algorithm is stronger than the IKEv1 or IKEv2 encryption algorithms, the child SA negotiation will be aborted, and a new high-severity syslog and debug message will be issued to identify the cause of the failed negotiation.

The following table lists the supported encryption ciphers in order of strength (from highest to lowest). The encryption ciphers on the same line have equivalent strength for purposes of this check.

Table 3: Supported Encryption Ciphers

ISAKMP/IKEv1	IKEv2	IPSec
AES-256	AES-CBC-256 (default), AES-GCM-256	ESP-AES-256
AES-192	AES-CBC-192	ESP-AES-192
		ESP-SEAL-160
AES-128 (default)	AES-CBC-128, AES-GCM-128	ESP-AES-128 (default), ESP-GCM-128

Examples

The following example shows how to configure Security Association Strength Enforcement.

```
Router(config) #crypto ipsec ike sa-strength-enforcement
% Warning: Please make sure IKE SA encryption keysize configured, is greater than or equal
```

```
 to IPSec SA encryption keysize.
Please run "clear crypto session" to enforce stronger IKE SA encryption immediately.
```

Related Commands

Command	Description
show crypto session detail	Display the status of the crypto session.
show running-config ipsec	Displays the IPSec configuration details.

crypto ipsec ipv4-deny

To configure deny address ranges at the global (IPSec VPN SPA) level, use the **crypto ipsec ipv4 deny-policy** command in global configuration mode.

crypto ipsec ipv4-deny {**jump** | **clear** | **drop**}

Syntax Description

jump	Causes the search to jump to the beginning of the ACL associated with the next sequence in the crypto map and continues the search when a deny address is hit.
clear	Allows traffic to pass through in the clear (unencrypted) state when a deny address is hit.
drop	Causes traffic to be dropped when a deny address is hit.

Command Modes

The default behavior is **jump**.

Command Modes

Global configuration

Command History

Release	Modification
12.2(18)SXE2	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 prevent repeated address ranges from being programmed in the hardware, resulting in more efficient TCAM space utilization.

Specifying a deny address range in an ACL results in “jump” behavior. When a denied address range is hit, it forces the search to “jump” to the beginning of the ACL associated with the next sequence in a crypto map and continue the search.

The **clear** keyword allows a deny address range to be programmed in hardware. The deny addresses are then filtered out for encryption and decryption. If the voice private network (VPN) mode is crypto-connect, when a deny address is hit, the search is stopped and traffic is allowed to pass in the clear (unencrypted) state.

If the VPN mode is VRF mode, the deny address matching traffic is dropped.

If you want to pass clear traffic on an address, you must insert a deny address range for each sequence in a crypto map.

Each permit list of addresses inherits all the deny address ranges specified in the ACL. A deny address range causes the software to do a subtraction of the deny address range from a permit list, and creates multiple permit address ranges that need to be programmed in hardware. This behavior can cause repeated address ranges to be programmed in the hardware for a single deny address range, resulting in multiple permit address ranges in a single ACL.

If you apply the specified keyword **(jump**, **clear**, or **drop**) when crypto maps are already configured on the IPSec VPN SPA, all existing IPSec sessions are temporarily removed and restarted, which impacts traffic on your network.

The number of deny entries that can be specified in an ACL are dependent on the keyword specified:

- **jump** --Supports up to 8 deny entries in an ACL.
- **clear** --Supports up to 1000 deny entries in an ACL.
- **drop** --Supports up to 1000 deny entries in an ACL.

Examples

The following example shows a configuration using the deny-policy **clear** option. In this example, when a deny address is hit, the search will stop and traffic will be allowed to pass in the clear (unencrypted) state:

```
Router(config)# crypto ipsec ipv4-deny clear
```

Related Commands

Command	Description
access-list	Defines a standard or extended IP access list.

crypto ipsec nat-transparency

To enable security parameter index (SPI) matching or User Datagram Protocol (UDP) encapsulation between two Virtual Private Network (VPN) devices, use the **crypto ipsec nat-transparency**command on both devices in global configuration mode. To disable both SPI matching and UDP encapsulation, use the **no** form of this command with each keyword.

crypto ipsec nat-transparency {**spi-matching** | **udp-encaps**}
no crypto ipsec nat-transparency {**spi-matching** | **udp-encaps**}

Syntax Description

spi-matching	Enables SPI matching on both endpoints.
udp-encaps	Enables UDP encapsulation on both endpoints.

Command Default

When this command is entered, UDP encapsulation is enabled by default.

Command Modes

Global configuration

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(15)T	The command syntax was modified to add the **spi-matching**keyword**.**

Usage Guidelines

You can use this command to resolve issues that arise when Network Address Translation (NAT) is configured in an IP Security (IPsec)-aware network. This command has two mutually exclusive options:

- The default option is UDP encapsulation of the IPsec protocols.
- The alternative is to match the inbound SPI to the outbound SPI.

When you enter the **crypto ipsec nat-transparency** command, UDP encapsulation is configured unless you either specifically disable it or configure SPI matching. You can disable both options, but doing so might cause problems if the device you are configuring uses NAT and is part of a VPN.

To disable SPI matching, configure UDP encapsulation or use the **no** form of this command with the keyword **spi-matching**. To disable UDP encapsulation, configure SPI matching or use the **no** form of this command with the keyword **udp-encaps**. To disable both SPI matching and UDP encapsulation, first disable UDP encapsulation, and then disable SPI matching. If you disable both options, the **show running-config** command displays: **no crypto ipsec nat-transparency udp-encaps.**

Examples

The following example enables SPI matching on the endpoint routers:

```
crypto ipsec nat-transparency spi-matching
```

Related Commands

Command	Description
clear ip nat translation	Clears dynamic NAT translations from the translation table.

Command	Description
ip nat	Designates that traffic originating from or destined for the interface is subject to NAT.
ip nat inside destination	Enables NAT of the inside destination address.
ip nat inside source	Enables NAT of the inside source address.
ip nat outside source	Enables NAT of the outside source address.
show ip nat statistics	Displays NAT statistics.
show ip nat translations	Displays active NAT translations.
show crypto isakmp sa detail nat	Displays NAT translations of source and destination addresses.

crypto ipsec optional

To enable IP Security (IPSec) passive mode, use the **crypto ipsec optional** command in global configuration mode. To disable IPSec passive mode, use the **no** form of this command.

crypto ipsec optional
no crypto ipsec optional

Syntax Description

This command has no arguments or keywords.

Command Default

IPSec passive mode is not enabled.

Command Modes

Global configuration

Command History

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

Usage Guidelines

Use the **crypto ipsec optional**command to implement an intermediate mode (IPSec passive mode) that allows a router to accept unencrypted and encrypted data. IPSec passive mode is valuable for users who wish to migrate existing networks to IPSec because all routers will continue to interact with routers that encrypt data (that is, that have been upgraded with IPSec) and also with routers that have yet to be upgraded.

After this feature is disabled, all active connections that are sending unencrypted packets are cleared, and a message that reminds the user to enter the **write memory** command is sent.

Note Because a router in IPSec passive mode is insecure, ensure that no routers are accidentally left in this mode after upgrading a network.

Examples

The following example shows how to enable IPSec passive mode:

```
crypto map xauthmap 10 ipsec-isakmp
 set peer 209.165.202.145
 set transform-set xauthtransform
 match address 192
!
crypto ipsec optional
!
interface Ethernet1/0
 ip address 209.165.202.147 255.255.255.224
 crypto map xauthmap
!
access-list 192 permit ip host 209.165.202.147 host 209.165.202.145
```

crypto ipsec optional retry

To adjust the amount of time that a packet can be routed in the clear (unencrypted), use the **crypto ipsec optional retry**command in global configuration mode. To return to the default setting (5 minutes), use the **no** form of this command.

crypto ipsec optional retry *seconds*
no crypto ipsec optional retry *seconds*

Syntax Description

seconds	Time a connection can exist before another attempt is made to establish an encrypted IP Security (IPSec) session. The default value is 5 minutes.

Command Default

5 minutes

Command Modes

Global configuration

Command History

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

Usage Guidelines

You must enable the **crypto ipsec optional** command, which enables IPSec passive mode, before you can use this command.

Examples

The following example shows how to enable IPSec passive mode:

```
crypto map xauthmap 10 ipsec-isakmp
 set peer 209.165.202.145
 set transform-set xauthtransform
 match address 192
!
crypto ipsec optional
crypto ipsec optional retry 60
!
interface Ethernet1/0
 ip address 209.165.202.147 255.255.255.224
 crypto map xauthmap
!
access-list 192 permit ip host 209.165.202.147 host 209.165.202.145
```

Related Commands

Command	Description
crypto ipsec optional	Enables IPSec passive mode.

crypto ipsec profile

To define the IP Security (IPsec) parameters that are to be used for IPsec encryption between two IPsec routers and to enter IPsec profile configuration mode, use the **crypto ipsec profile** command in global configuration mode. To delete an IPsec profile, use the **no** form of this command. To return the IPsec profile to its default value, use the **default** form of this command.

crypto ipsec profile *name*
no crypto ipsec profile *name*
default crypto ipsec profile

Syntax Description

name	Profile name.

Command Default

The default IPsec profile is used.

Command Modes

Global configuration (config)

Command History

Release	Modification
12.2(13)T	This command was introduced.
12.2(18)SXE	This command was integrated into Cisco IOS Release 12.2(18)SXE.
12.4(4)T	Support for IPv6 was added.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
Cisco IOS XE Release 2.1	This command was introduced on Cisco ASR 1000 Series Routers.

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

An IPsec profile abstracts the IPsec policy settings into a single profile that can be used in other parts of the Cisco IOS configuration.

The IPsec profile shares most of the same commands with the crypto map configuration, but only a subset of the commands are valid in an IPsec profile. Only commands that pertain to an IPsec policy can be issued under an IPsec profile; you cannot specify the IPsec peer address or the access control list (ACL) to match the packets that are to be encrypted.

After this command has been enabled, the following commands can be configured under an IPsec profile:

- **default** —Lists the commands that can be configured under the **crypto ipsec profile** command.
- **description** —Describes the crypto map statement policy.

- **dialer** —Specifies dialer-related commands.
- **redundancy** —Specifies a redundancy group name.
- **set-identity** —Specifies identity restrictions.
- **set isakmp-profile** —Specifies an ISAKMP profile.
- **set pfs** —Specifies perfect forward secrecy (PFS) settings.
- **set security-association** —Defines security association parameters.
- **set-transform-set** —Specifies a list of transform sets in order of priority.

After enabling this command, the only parameter that must be defined under the profile is the transform set via the **set transform-set** command.

You can modify the default IPsec profile using the **crypto ipsec profile default** command. You can disable the default IPsec profile using the **no crypto ipsec profile default** command.

For more information on transform sets, refer to the section “Defining Transform Sets” in the chapter “Configuring IPSec Network Security” in the *Cisco IOS Security Configuration Guide*.

Examples

The following example shows how to configure a crypto map that uses an IPsec profile:

```
crypto ipsec transform-set cat-transforms esp-aes esp-sha-hmac
 mode transport
!
crypto ipsec profile cat-profile
 set transform-set cat-transforms
 set pfs group14
!
interface Tunnel1
 ip address 192.168.1.1 255.255.255.252
 tunnel source FastEthernet2/0
 tunnel destination 10.13.7.67
 tunnel protection ipsec profile cat-profile
```

Related Commands

Command	Description
crypto ipsec transform-set	Defines a transform set.
set pfs	Specifies that IPsec should ask for PFS when requesting new security associations for a crypto map entry.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
tunnel protection	Associates a tunnel interface with an IPsec profile.

crypto ipsec security-association dummy

To enable the generation and transmission of dummy packets in an IPsec traffic flow, use the **crypto ipsec security-association dummy** command in global configuration mode. To disable this generation and transmission, use the **no** form of this command.

crypto ipsec security-association dummy{**pps** *rate* |**seconds** *seconds*}
no crypto ipsec security-association dummy

Syntax Description

pps *rate*	Packets per second rate. The range is 0 to 25.
seconds *seconds*	Delay, in seconds, between packets. The range is 1 to 3600.

Command Default

Generating and transmitting dummy packets is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.2(4)M3	This command was introduced.
Cisco IOS XE Release 3.10S	This command was integrated into Cisco IOS XE Release 3.10S.

Usage Guidelines

RFC 4303 specifies a method to hide packet data in an IPsec traffic flow by adding dummy packets in the traffic flow. Use the **crypto ipsec security-association dummy** command to generate and transmit dummy packets to hide data in the IPsec traffic flow. The dummy packet is designated by setting the next header field in the Encapsulating Security Payload (ESP) packet to a value of 59. When a crypto engine receives such packets, it discards them.

Use the **pps** *rate* keyword/argument pair to specify a rate greater than one packet per second.

Examples

The following example generates dummy packets in the traffic flow every five seconds:

```
Device# configure terminal
Device(config)# crypto ipsec security-association dummy seconds 5
```

Related Commands

Command	Description
set security-association dummy	Enables the generation and transmission of dummy packets for an IPsec traffic flow in a crypto map.

crypto ipsec security-association idle-time

To configure the IP Security (IPSec) security association (SA) idle timer, use the **crypto ipsec security-association idle-time** command in global configuration mode or crypto map configuration mode. To inactivate the IPSec SA idle timer, use the **no** form of this command.

crypto ipsec security-association idle-time *seconds*
no crypto ipsec security-association idle-time

Syntax Description

seconds	Time, in seconds, that the idle timer allows an inactive peer to maintain an SA. The range is 60 to 86400 seconds.

Command Default

IPSec SA idle timers are disabled.

Command Modes

Global configuration
Crypto map configuration

Command History

Release	Modification
12.2(15)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.

Usage Guidelines

Use the **crypto ipsec security-association idle-time** command to configure the IPSec SA idle timer. This timer controls the amount of time that an SA will be maintained for an idle peer.

Use the **crypto ipsec security-association lifetime**command to configure global lifetimes for IPSec SAs. There are two lifetimes: a timed lifetime and a traffic-volume lifetime. A security association expires after the first of these lifetimes is reached.

The IPSec SA idle timers are different from the global lifetimes for IPSec SAs. The expiration of the global lifetimes is independent of peer activity. The IPSec SA idle timer allows SAs associated with inactive peers to be deleted before the global lifetime has expired.

If the IPSec SA idle timers are not configured with the **crypto ipsec security-association idle-time** command, only the global lifetimes for IPSec SAs are applied. SAs are maintained until the global timers expire, regardless of peer activity.

Note If the last IPSec SA to a given peer is deleted due to idle timer expiration, the Internet Key Exchange (IKE) SA to that peer will also be deleted.

Release 12.2(33)SRA or later releases Release 12.2(33)SXH or later releases

In a system using the IPSec VPN SPA with these software releases, the configured value for the *seconds* argument is rounded up to the next multiple of 600 seconds (ten minutes), and the rounded value becomes the polling interval for SA idle detection. Because the SA idle condition must be observed in two successive pollings, the period of inactivity may last up to twice the polling period before the SAs are deleted.

Examples

The following example configures the IPSec SA idle timer to drop SAs for inactive peers after at least 750 seconds:

```
Router# configure terminal
Router(config)# crypto ipsec security-association idle-time 750
```

With Cisco IOS Release 12.2(15)T or later releases, the SA will be deleted after an inactivity period of 750 seconds.

With Cisco IOS Release 12.2(33)SRA or 12.2(33)SXH or later releases, the configured value of 750 seconds will be rounded up to 1200 seconds (the next multiple of 600), which becomes the idle polling interval. The SA will be deleted after two successive idle pollings, resulting in an inactivity period of between 1200 and 2400 seconds before deletion.

Related Commands

Command	Description
clear crypto sa	Deletes IPSec SAs.
crypto ipsec security-association lifetime	Changes global lifetime values used when negotiating IPSec SAs.

crypto ipsec security-association lifetime

To change global lifetime values used when negotiating IPsec security associations, use the **crypto ipsec security-association lifetime**command in global configuration mode. To reset a lifetime to the default value, use the **no** form of this command.

crypto ipsec security-association lifetime {**seconds** *seconds* | **kilobytes** *kilobytes* | **kilobytes disable**}
no crypto ipsec security-association lifetime {**seconds** | **kilobytes** | **kilobytes disable**}

Syntax Description

seconds *seconds*	Specifies the number of seconds a security association will live before expiring. The default is 3600 seconds (one hour).
kilobytes *kilobytes*	Specifies the volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before that security association expires. The default is 4,608,000 kilobytes.
kilobytes disable	Disables the Internet Key Exchange (IKE) rekey based on volume only on the router on which it is configured. • If the **no** form is used with this keyword, lifetime settings switch back to the default settings.

Command Default

3600 seconds (one hour) and 4,608,000 kilobytes (10 megabits per second for one hour).

Command Modes

Global configuration

Command History

Release	Modification
11.3T	This command was introduced.
12.2(13)T	The security association negotiation changed. Prior to Cisco IOS Release 12.2(13)T, the new security association was negotiated either 30 seconds before the **seconds** lifetime expired or when the volume of traffic through the tunnel reached 256 kilobytes less than the **kilobytes** lifetime. Effective with Cisco IOS Release 12.2(13)T, the negotiation is either 30 seconds before the **seconds** lifetime expires or when the volume of traffic through the tunnel reaches 90 percent of the **kilobytes** lifetime.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
12.2(33)SXI	The **disable** keyword was added. **Note** This keyword addition is for only Cisco IOS Release 12.2(33)SXI.
15.0(1)M	The **disable** keyword was added.

Usage Guidelines

IPsec security associations use shared secret keys. These keys and their security associations time out together.

Assuming that the particular crypto map entry does not have lifetime values configured, when the router requests new security associations during security association negotiation, it will specify its global lifetime value in the request to the peer; it will use this value as the lifetime of the new security associations. When the router receives a negotiation request from the peer, it will use the smaller of the lifetime value proposed by the peer or the locally configured lifetime value as the lifetime of the new security associations.

There are two lifetimes: a “timed” lifetime and a “traffic-volume” lifetime. The security association expires after the first of these lifetimes is reached.

If you change a global lifetime, the change is only applied when the crypto map entry does not have a lifetime value specified. The change will not be applied to existing security associations, but will be used in subsequent negotiations to establish new security associations. If you want the new settings to take effect sooner, you can clear all or part of the security association database by using the **clear crypto sa** command. Refer to the **clear crypto sa** command for more details.

To change the global timed lifetime, use the **crypto ipsec security-association lifetime seconds** form of the command. The timed lifetime causes the security association to time out after the specified number of seconds have passed.

To change the global traffic-volume lifetime, use the **crypto ipsec security-association lifetime kilobytes** form of the command. The traffic-volume lifetime causes the security association to time out after the specified amount of traffic (in kilobytes) has been protected by the key of the security association.

Shorter lifetimes can make it harder to mount a successful key recovery attack, since the attacker has less data encrypted under the same key to work with. However, shorter lifetimes require more CPU processing time for establishing new security associations.

The lifetime values are ignored for manually established security associations (security associations installed using an **ipsec-manual** crypto map entry).

How The Lifetimes Work

The security association (and corresponding keys) will expire according to whichever occurs sooner, either after the number of seconds has passed (specified by the **seconds**keyword) or after the amount of traffic in kilobytes has passed (specified by the **kilobytes** keyword).

A new security association is negotiated *before* the lifetime threshold of the existing security association is reached, to ensure that a new security association is ready for use when the old one expires. The **seconds**lifetime and the **kilobytes**lifetime each have a jitter mechanism to avoid security association rekey collisions. The new security association is negotiated either (30 plus a random number of) seconds before the **seconds** lifetime expires or when the traffic volume reaches (90 minus a random number of) percent of the **kilobytes**lifetime (whichever occurs first).

If no traffic has passed through the tunnel during the entire life of the security association, a new security association is not negotiated when the lifetime expires. Instead, a new security association will be negotiated only when IPsec sees another packet that should be protected.

Disabling the Volume Lifetime

The **crypto ipsec security-association lifetime kilobytes disable** form of the command disables the volume lifetime. Using this command form should result in a significant improvement in performance and reliability, and this option can be used to reduce packet loss in high traffic environments. It can be used to prevent frequent rekeys that are triggered by reaching the volume lifetimes.

Note The volume lifetime can also be disabled using the **set security-association lifetime kilobytes disable** command.

Examples

The following example shortens both lifetimes, because the administrator feels there is a higher risk that the keys could be compromised. The timed lifetime is shortened to 2700 seconds (45 minutes), and the traffic-volume lifetime is shortened to 2,304,000 kilobytes (10 megabits per second for one half hour).

```
crypto ipsec security-association lifetime seconds 2700
crypto ipsec security-association lifetime kilobytes 2304000
```

The following example shows that the **kilobytes disable** keyword has been used to disable the volume lifetime.

```
crypto ipsec security-association lifetime kilobytes disable
```

Related Commands

Command	Description
set security-association lifetime	Overrides (for a particular crypto map entry) the global lifetime value, which is used when negotiating IPsec security associations.
show crypto ipsec security-association lifetime	Displays the security-association lifetime value configured for a particular crypto map entry.

crypto ipsec security-association multi-sn

To enable multiple sequence number space per IPSec SA (security association), use the **crypto ipsec security-association multi-sn** command in global configuration mode. To disable multiple sequence number space, use the **no** form of the command.

crypto ipsec security-association multi-sn
no crypto ipsec security-association multi-sn

Syntax Description

This command has no keywords or arguments

Command Default

Multiple sequence number space is not enabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
16.6.1	This command was introduced.

Usage Guidelines

All existing sessions need to be cleared before configuring this feature. Else, traffic from the existing sessions will be dropped.

This feature needs to be configured on both the tunnel routers in an IPSec connection. If this featues is only enabled on one router, the other router will drop packets.

Example

The following example shows how to enable multiple sequence number space on a device:

```
Device(config)# crypto ipsec security-association multi-sn
Warning: Existing sessions if any, might experience traffic drop due to SPI not found
```

Note This command is not supported on Cisco ISR44xx series devices.

crypto ipsec security-association replay disable

To disable anti-replay checking globally, use the **crypto ipsec security-association replay disable** command in global configuration mode. To reset the configuration to enable anti-replay checking, use the **no** form of this command.

crypto ipsec security-association replay disable
no crypto ipsec security-association replay disable

Syntax Description

This command has no arguments or keywords.

Command Default

Anti-replay checking is enabled.

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF6	This command was integrated into Cisco IOS Release 12.2(18)SXF6.

Examples

The following example shows that anti-replay checking has been disabled globally:

```
crypto map mymap 10
exit
crypto ipsec security-association replay disable
```

Related Commands

Command	Description
crypto ipsec security-association replay window-size	Sets the size of the SA anti-replay window.

crypto ipsec security-association replay window-size

To set the size of the security association (SA) anti-replay window globally, use the **crypto ipsec security-association replay window-size**command in global configuration mode. To reset the window size to the default of 64, use the **no** form of this command.

crypto ipsec security-association replay window-size [*N*]
no crypto ipsec security-association replay window-size

Syntax Description

N	(Optional) Size of the window. Values can be 64, 128, 256, 512, or 1024. This value becomes the default value. **Note** The window size is significant only if anti-replay checking is enabled.

Command Default

If a window size is not entered, the default is 64.

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF6	This command was integrated into Cisco IOS Release 12.2(18)SXF6.

Examples

The following example shows that the size of the SA anti-replay window has been set globally to 128:

```
crypto map mymap 20
exit
crypto ipsec security-association replay window-size 128
```

Related Commands

Command	Description
crypto ipsec security-association replay disable	Disables anti-replay checking.

crypto ipsec server send-update

To send auto-update notifications any time after an Easy VPN connection is "up," use the **crypto ipsec server send-update** command in privileged EXEC mode.

crypto ipsec server send-update *group-name*
no crypto ipsec server send-update *group-name*

Syntax Description

group-name	Name of group to which to send auto-update notifications.

Command Default

Auto-update notifications are not sent.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
12.4(2T)	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS 12.2SX family of releases. Support in a specific 12.2SX release is dependent on your feature set, platform, and platform hardware.

Usage Guidelines

This command is configured on a server. By configuring the command, the auto update notification is sent manually after the tunnel is "up."

Examples

The following example shows that automatic update notifications are to be sent to GroupA:

```
crypto ipsec server send-update GroupA
```

crypto ipsec transform-set

To define a transform set—an acceptable combination of security protocols and algorithms—use the **crypto ipsec transform-set** command in global configuration mode. To delete a transform set, use the **no** form of this command. To return the transform-set to its default value, use the **default** form of this command.

crypto ipsec transform-set *transform-set-name transform1* [*transform2*] [*transform3*] [*transform4*]
no crypto ipsec transform-set *transform-set-name*
default crypto ipsec transform-set

Syntax Description

transform-set-name	Name of the transform set to create (or modify).
transform1 transform2 transform3 transform4	Type of transform set. You may specify up to four "transforms": one Authentication Header (AH), one Encapsulating Security Payload (ESP) encryption, one ESP authentication, and one compression. These transforms define the IP Security (IPSec) security protocols and algorithms. Accepted transform values are described in the table below.

Command Default

The default transform-set is used.

Command Modes

Global configuration

This command invokes the crypto transform configuration mode.

Command History

Release	Modification
11.3 T	This command was introduced.
12.2(13)T	The following transform set options were added: **esp-aes**, **esp-aes 192**, and **esp-aes 256**.
12.3(7)T	The **esp-seal** transform set option was added.
12.2(33)SRA	This command was integrated into Cisco IOS release 12.(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.1(2)T	This command was modified in Cisco IOS Release 15.1(2)T. The **esp-gcm** and **esp-gmac transforms were added.**

Usage Guidelines

Note Security threats, as well as the cryptographic technologies to help protect against them, are constantly changing. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

A transform set is an acceptable combination of security protocols, algorithms, and other settings to apply to IPSec-protected traffic. During the IPSec security association (SA) negotiation, the peers agree to use a particular transform set when protecting a particular data flow.

You can configure multiple transform sets, and then specify one or more of these transform sets in a crypto map entry. The transform set defined in the crypto map entry is used in the IPSec SA negotiation to protect the data flows specified by the access list of that crypto map entry. During the negotiation, the peers search for a transform set that is the same at both peers. When such a transform set is found, it is selected and will be applied to the protected traffic as part of the IPSec SAs of both peers.

When Internet Key Exchange (IKE) is not used to establish SAs, a single transform set must be used. The transform set is not negotiated.

Before a transform set can be included in a crypto map entry, it must be defined using this command.

Although this command is similar to the **crypto isakmp policy** command, the IKEv2 proposal differs as follows:

A transform set specifies one or two IPSec security protocols (either AH, ESP, or both) and specifies which algorithms to use with the selected security protocol. The AH and ESP IPSec security protocols are described in the “*Allowed Transform Combinations*” section.

To define a transform set, you specify one to four “transforms”--each transform represents an IPSec security protocol (AH or ESP) plus the algorithm you want to use. When the particular transform set is used during negotiations for IPSec SAs, the entire transform set (the combination of protocols, algorithms, and other settings) must match a transform set at the remote peer.

In a transform set you can specify the AH protocol, the ESP protocol, or both. If you specify an ESP protocol in a transform set, you can specify just an ESP encryption transform set or both an ESP encryption transform set and an ESP authentication transform set.

The table below lists the acceptable transform set combination selections for the AH and ESP protocols.

Table 4: Allowed Transform Combinations

Transform Type	Transform	Description
AH Transform >*Pick only one.*	**ah-md5-hmac**	AH with the MD5 (Message Digest 5) (a Hash-based Message Authentication Code [HMAC] variant) authentication algorithm. (No longer recommended).
	ah-sha-hmac	AH with the SHA (Secure Hash Algorithm) (an HMAC variant) authentication algorithm.

Transform Type	Transform	Description
ESP Encryption Transform (*>Pick only one.*	**esp-aes**	ESP with the 128-bit Advanced Encryption Standard (AES) encryption algorithm.
	esp-gcm **esp-gmac**	The **esp-gcm** and **esp-gmac** transforms are ESPs with either a 128 or 256 bit encryption algorithm. The default for either of these transforms is 128 bits. **Note** Both the **esp-gcm** and **esp-gmac** transforms cannot be configured together with any other ESP transform within the same crypto IPsec transform set using the crypto ipsec transform-set command.
	esp-aes 192	ESP with the 192-bit AES encryption algorithm.
	esp-aes 256	ESP with the 256-bit AES encryption algorithm.
	esp-des	ESP with the 56-bit Data Encryption Standard (DES) encryption algorithm. (No longer recommended).
	esp-3des	ESP with the 168-bit DES encryption algorithm (3DES or Triple DES). (No longer recommended).
	esp-null	Null encryption algorithm.
	esp-seal	ESP with the 160-bit SEAL encryption algorithm. (No longer recommended).
ESP Authentication Transform (*Pick only one.*)	**esp-md5-hmac**	ESP with the MD5 (HMAC variant) authentication algorithm. (No longer recommended).
	esp-sha-hmac	ESP with the SHA (HMAC variant) authentication algorithm.
IP Compression Transform	**comp-lzs**	IP compression with the Lempel-Ziv-Stac (LZS) algorithm. **Note** The IP Compression Transform is not supported on Cisco IOS XE software.

Examples of acceptable transform set combinations are as follows:

- **ah-sha-hmac**
- **esp-gcm 256**
- **esp-aes**
- **esp-aes** and **esp-sha-hmac**
- **ah-sha-hmac** and **esp-aes** and **esp-sha-hmac**
- **comp-lzs** and **esp-sha-hmac** and **esp-aes** (In general, the **comp-lzs** transform set can be included with any other legal combination that does not already include the **comp-lzs** transform.)
- **esp-seal** and **esp-md5-hmac**

The parser will prevent you from entering invalid combinations; for example, after you specify an AH transform set, it will not allow you to specify another AH transform set for the current transform set.

IPSec Protocols: AH and ESP

Both the AH and ESP protocols implement security services for IPSec.

AH provides data authentication and antireplay services.

ESP provides packet encryption and optional data authentication and antireplay services.

ESP encapsulates the protected data--either a full IP datagram (or only the payload)--with an ESP header and an ESP trailer. AH is embedded in the protected data; it inserts an AH header immediately after the outer IP header and before the inner IP datagram or payload. Traffic that originates and terminates at the IPSec peers can be sent in either tunnel or transport mode; all other traffic is sent in tunnel mode. Tunnel mode encapsulates and protects a full IP datagram, while transport mode encapsulates or protects the payload of an IP datagram. For more information about modes, see the **mode**(IPSec) command description.

The esp-seal Transform

There are three limitations on the use of the **esp-seal** transform set:

- The **esp-seal** transform set can be used only if no crypto accelerators are present. This limitation is present because no current crypto accelerators implement the SEAL encryption transform set, and if a crypto accelerator is present, it will handle all IPSec connections that are negotiated with IKE. If a crypto accelerator is present, the Cisco IOS software will allow the transform set to be configured, but it will warn that it will not be used as long as the crypto accelerator is enabled.
- The **esp-seal** transform set can be used only in conjunction with an authentication transform set, namely one of these: **esp-md5-hmac**, (not recommended) **esp-sha-hmac**, **ah-md5-hmac** (not recommended), or **ah-sha-hmac**. This limitation is present because SEAL encryption is especially weak when it comes to protecting against modifications of the encrypted packet. Therefore, to prevent such a weakness, an authentication transform set is required. (Authentication transform sets are designed to foil such attacks.) If you attempt to configure an IPSec transform set using SEAL but without an authentication transform set, an error is generated, and the transform set is rejected.
- The **esp-seal** transform set cannot be used with a manually keyed crypto map. This limitation is present because such a configuration would reuse the same keystream for each reboot, which would compromise security. Because of the security issue, such a configuration is prohibited. If you attempt to configure a manually keyed crypto map with a SEAL-based transform set, an error is generated, and the transform set is rejected.

Selecting Appropriate Transform Sets

The following tips may help you select transform sets that are appropriate for your situation:

- If you want to provide data confidentiality, include an ESP encryption transform set.
- If you want to ensure data authentication for the outer IP header as well as the data, include an AH transform set. (Some consider the benefits of outer IP header data integrity to be debatable.)
- If you use an ESP encryption transform set, also consider including an ESP authentication transform set or an AH transform set to provide authentication services for the transform set.
- If you want data authentication (either using ESP or AH), you can choose from the MD5 or SHA (HMAC keyed hash variants) authentication algorithms. The SHA algorithm is generally considered stronger than MD5 but is slower.
- Note that some transform sets might not be supported by the IPSec peer.

Note If a user enters an IPSec transform set that the hardware does not support, a warning message will be displayed immediately after the **crypto ipsec transform-set** command is entered.

- In cases where you need to specify an encryption transform set but do not actually encrypt packets, you can use the **esp-null** transform.

Suggested transform set combinations follow:

- **esp-aes** and **esp-sha-hmac**
- **esp-aes 256** and **esp-sha-hmac**

The Crypto Transform Configuration Mode

After you issue the **crypto ipsec transform-set** command, you are put into the crypto transform configuration mode. While in this mode, you can change the mode to tunnel or transport. (These are optional changes.) After you have made these changes, type **exit** to return to global configuration mode. For more information about these optional changes, see the **match address** (IPSec) and **mode** (IPSec) command descriptions.

Changing Existing Transform Sets

If one or more transform sets are specified in the **crypto ipsec transform-set** command for an existing transform set, the specified transform sets will replace the existing transform sets for that transform set.

If you change a transform set definition, the change is only applied to crypto map entries that reference the transform set. The change will not be applied to existing SAs but will be used in subsequent negotiations to establish new SAs. If you want the new settings to take effect sooner, you can clear all or part of the SA database by using the **clear crypto sa** command.

Default Transform Set

You can modify the default transform-set using the **crypto ipsec transform-set default** command. You can disable the default transform-set using the **no crypto ipsec transform-set default** command.

If you do not specify a transform-set, the default transform-set is used with the default profile.

Examples

The following example defines two transform sets. The first transform set will be used with an IPSec peer that supports the newer ESP and AH protocols. The second transform set will be used with an IPSec peer that supports only the older transforms.

```
Router (config)# crypto ipsec transform-set newer esp-aes esp-sha-hmac
Router (config)# crypto ipsec transform-set older ah-md5-hmac esp-des
```

The following example is a sample warning message that is displayed when a user enters an IPSec transform set that the hardware does not support:

```
Router (config)# crypto ipsec transform transform-1 esp-aes 256 esp-sha-hmac
WARNING:encryption hardware does not support transform
esp-aes 256 within IPSec transform transform-1
```

The following output example shows that SEAL encryption has been correctly configured with an authentication transform set:

```
Router (config)# crypto ipsec transform-set seal esp-seal esp-sha-hmac
```

The following example is a warning message that is displayed when SEAL encryption has been configured with a crypto accelerator present:

```
Router (config)# show running-config
```

```
crypto ipsec transform-set seal esp-seal esp-sha-hmac
! Disabled because transform not supported by encryption hardware
```

The following example is an error message that is displayed when SEAL encryption has been configured without an authentication transform set:

```
Router (config)# crypto ipsec transform seal esp-seal
ERROR: Transform requires either ESP or AH authentication.
```

The following example is an error message that is displayed when SEAL encryption has been configured within a manually keyed crypto map:

```
Router (config)# crypto map green 10 ipsec-manual
%Note: This new crypto map will remain disabled until a peer
       and a valid access list have been configured.
Router (config-crypto-map)# set transform seal
ERROR: transform seal illegal for a manual crypto map.
```

Related Commands

Command	Description
clear crypto sa	Deletes IPSec security associations.
crypto ipsec transform-set	Defines a transform set--an acceptable combination of security protocols and algorithms.
match address	Specifies an extended access list for a crypto map entry.
mode (IPSec)	Changes the mode for a transform set.
set transform-set	Specifies which transform sets can be used with the crypto map entry.
show crypto ipsec transform-set	Displays the configured transform sets.