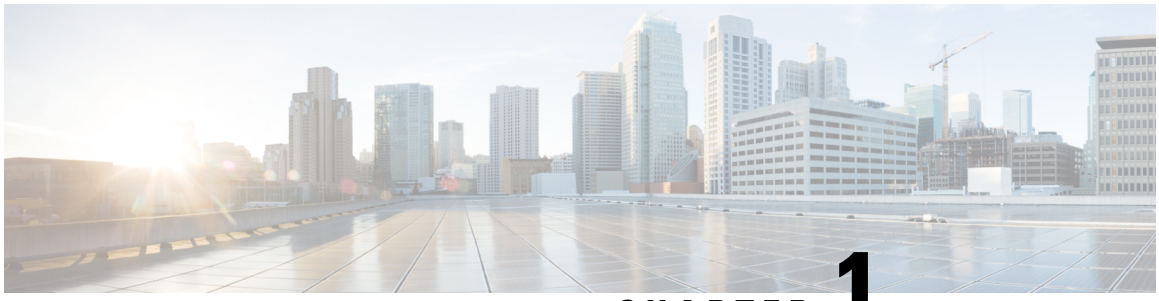
FlexVPN and Internet Key Exchange Version 2 Configuration Guide, Cisco IOS XE Gibraltar 16.11.x

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

CONTENTS



CHAPTER 1

Read Me First

Important Information about Cisco IOS XE 16

Effective Cisco IOS XE Release 3.7.0E for Catalyst Switching and Cisco IOS XE Release 3.17S (for Access and Edge Routing) the two releases evolve (merge) into a single version of converged release—the Cisco IOS XE 16—providing one release covering the extensive range of access and edge products in the Switching and Routing portfolio.

Feature Information

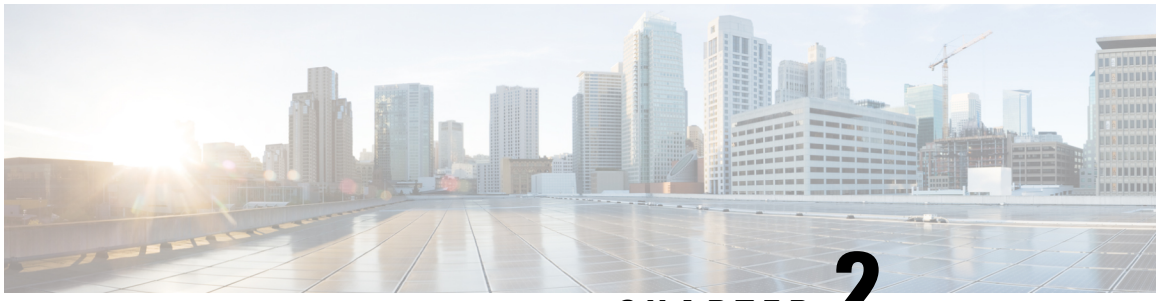
Use Cisco Feature Navigator to find information about feature support, platform support, and Cisco software image support. An account on Cisco.com is not required.

Related References

- Cisco IOS Command References, All Releases

Obtaining Documentation and Submitting a Service Request

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business impact you're looking for with the technologies that matter, visit Cisco Services.
- To submit a service request, visit Cisco Support.
- To discover and browse secure, validated enterprise-class apps, products, solutions and services, visit Cisco Marketplace.
- To obtain general networking, training, and certification titles, visit Cisco Press.
- To find warranty information for a specific product or product family, access Cisco Warranty Finder.



CHAPTER 2

Introduction to FlexVPN

Internet Key Exchange Version 2 (IKEv2), a next-generation key management protocol based on RFC 4306, is an enhancement of the IKE Protocol. IKEv2 is used for performing mutual authentication and establishing and maintaining security associations (SAs).

FlexVPN is Cisco's implementation of the IKEv2 standard featuring a unified paradigm and CLI that combines site to site, remote access, hub and spoke topologies and partial meshes (spoke to spoke direct). FlexVPN offers a simple but modular framework that extensively uses the tunnel interface paradigm while remaining compatible with legacy VPN implementations using crypto maps.

This guide contains the following modules:

Configuring Internet Key Exchange Version 2 (IKEv2) and FlexVPN Remote Access

This module describes IKEv2 CLI and is divided into basic and advanced sections.

The basic section introduces basic IKEv2 commands and describes IKEv2 smart defaults and the mandatory IKEv2 commands required for FlexVPN remote access. This module is a prerequisite for understanding subsequent chapters.

The advanced section describes global IKEv2 commands and how to override the default IKEv2 commands.

Configuring FlexVPN Server

This module describes FlexVPN server features, IKEv2 commands required to configure FlexVPN server, remote access clients and the supported RADIUS attributes.

Configuring FlexVPN Client

This module describes FlexVPN client features and the IKEv2 commands required for FlexVPN client.

Configuring IKEv2 Load Balancer

This module describes the IKEv2 Load Balancer Support feature and the IKEv2 commands required to configure the IKEv2 Load Balancer.

Configuring IKEv2 Fragmentation

The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document.

Configuring IKEv2 Reconnect

The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client.

Configuring IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices.

Configuring IKEv2 Change of Authorization Support

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session.

Configuring Aggregate Authentication

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP

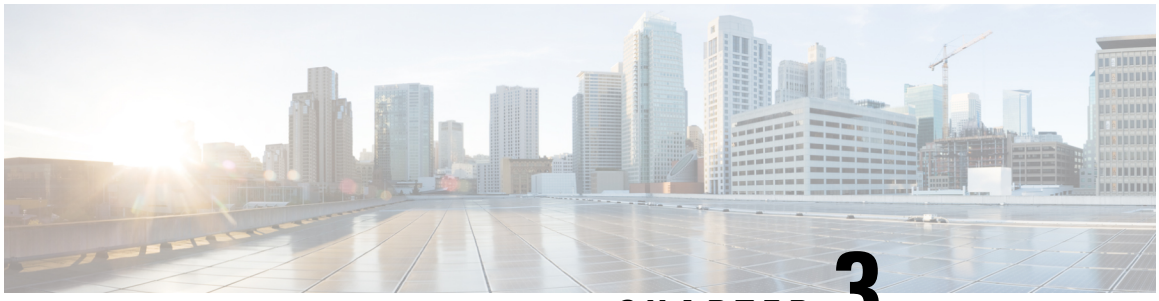
authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server.

Appendix: FlexVPN RADIUS Attributes

This module describes the RADIUS attributes supported by FlexVPN server.

Appendix: IKEv2 and Legacy VPNs

This module contains configuration examples on how to configure legacy VPNs such as crypto maps and DMVPN with Internet Key Exchange Version 2 (IKEv2).



CHAPTER 3

Configuring Internet Key Exchange Version 2

This module contains information about and instructions for configuring basic and advanced Internet Key Exchange Version 2 (IKEv2). The tasks and configuration examples for IKEv2 in this module are divided as follows:

- Basic IKEv2—Provides information about basic IKEv2 commands, IKEv2 smart defaults, basic IKEv2 profile, and IKEv2 key ring.
- Advanced IKEv2—Provides information about global IKEv2 commands and how to override IKEv2 smart defaults.

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.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Configuring Internet Key Exchange Version 2

You should be familiar with the concepts and tasks described in the "Configuring Security for VPNs with IPsec" module.

Restrictions for Configuring Internet Key Exchange Version 2

You cannot configure an option that is not supported on a specific platform. For example, in a security protocol, the capability of the hardware-crypto engine is important, and you cannot specify the Triple Data Encryption Standard (3DES) or the Advanced Encryption Standard (AES) type of encryption transform in a nonexportable image, or specify an encryption algorithm that a crypto engine does not support.

Note IKEv2 is not supported on Integrated Service Routers (ISR) G1.

Information About Internet Key Exchange Version 2

IKEv2 Supported Standards

Cisco implements the IP Security (IPsec) Protocol standard for use in Internet Key Exchange Version 2 (IKEv2).

Note Cisco no longer recommends using DES or MD5 (including HMAC variant); instead, you should use AES and SHA-256. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The component technologies implemented in IKEv2 are as follows:

- AES-CBC—Advanced Encryption Standard-Cipher Block Chaining
- SHA (HMAC variant)—Secure Hash Algorithm
- Diffie-Hellman—A public-key cryptography protocol
- DES—Data Encryption Standard (No longer recommended)
- MD5 (HMAC [Hash-based Message Authentication Code] variant)—Message digest algorithm 5 (No longer recommended)

For more information about supported standards and component technologies, see the "Supported Standards for Use with IKE" section in the "Configuring Internet Key Exchange for IPsec VPNs" module in the *Internet Key Exchange for IPsec VPNs Configuration Guide*.

Benefits of IKEv2

Dead Peer Detection and Network Address Translation-Traversal

Internet Key Exchange Version 2 (IKEv2) provides built-in support for Dead Peer Detection (DPD) and Network Address Translation-Traversal (NAT-T).

Certificate URLs

Certificates can be referenced through a URL and hash, instead of being sent within IKEv2 packets, to avoid fragmentation.

Denial of Service Attack Resilience

IKEv2 does not process a request until it determines the requester, which addresses to some extent the Denial of Service (DoS) problems in IKEv1, which can be spoofed into performing substantial cryptographic (expensive) processing from false locations.

EAP Support

IKEv2 allows the use of Extensible Authentication Protocol (EAP) for authentication.

Multiple Crypto Engines

If your network has both IPv4 and IPv6 traffic and you have multiple crypto engines, choose one of the following configuration options:

- One engine handles IPv4 traffic and the other engine handles IPv6 traffic.
- One engine handles both IPv4 and IPv6 traffic.

Reliability and State Management (Windowing)

IKEv2 uses sequence numbers and acknowledgments to provide reliability, and mandates some error-processing logistics and shared state management.

Internet Key Exchange Version 2 CLI Constructs

IKEv2 Proposal

An Internet Key Exchange Version 2 (IKEv2) proposal is a collection of transforms used in the negotiation of Internet Key Exchange (IKE) security associations (SAs) as part of the IKE_SA_INIT exchange. The transform types used in the negotiation are as follows:

- Encryption algorithm
- Integrity algorithm
- Pseudo-Random Function (PRF) algorithm
- Diffie-Hellman (DH) group

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 proposal. See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to override the default IKEv2 proposal and to define new proposals.

IKEv2 Policy

An IKEv2 policy contains proposals that are used to negotiate the encryption, integrity, PRF algorithms, and DH group in the IKE_SA_INIT exchange. It can have match statements, which are used as selection criteria to select a policy during negotiation.

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 policy. See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to override the default IKEv2 policy and to define new policies.

IKEv2 Profile

An IKEv2 profile is a repository of nonnegotiable parameters of the IKE SA, such as local or remote identities and authentication methods and services that are available to authenticated peers that match the profile. An IKEv2 profile must be attached to either a crypto map or an IPsec profile on the initiator. An IKEv2 profile is not mandatory on the responder.

IKEv2 Key Ring

An IKEv2 key ring is a repository of symmetric and asymmetric preshared keys and is independent of the IKEv1 key ring. The IKEv2 key ring is associated with an IKEv2 profile and hence supports a set of peers that match the IKEv2 profile. The IKEv2 key ring gets its VPN routing and forwarding (VRF) context from the associated IKEv2 profile.

IKEv2 Smart Defaults

The IKEv2 Smart Defaults feature minimizes the FlexVPN configuration by covering most of the use cases. IKEv2 smart defaults can be customized for specific use cases, though this is not recommended.

See the "Configuring Advanced IKEv2 CLI Constructs" section for information about how to modify the default IKEv2 constructs.

The following rules apply to the IKEv2 Smart Defaults feature:

1. A default configuration is displayed in the corresponding **show** command with **default** as a keyword and with no argument. For example, the **show crypto ikev2 proposal default** command displays the default IKEv2 proposal and the **show crypto ikev2 proposal** command displays the default IKEv2 proposal, along with any user-configured proposals.
2. A default configuration is displayed in the **show running-config all** command; it is not displayed in the **show running-config** command.
3. You can modify the default configuration, which is displayed in the **show running-config all** command.
4. A default configuration can be disabled using the **no** form of the command; for example, **no crypto ikev2 proposal default**. A disabled default configuration is not used in negotiation but the configuration is displayed in the **show running-config** command. A disabled default configuration loses any user modification and restores system-configured values.
5. A default configuration can be reenabled using the default form of the command, which restores system-configured values; for example, **default crypto ikev2 proposal**.

6. The default mode for the default transform set is transport; the default mode for all other transform sets is tunnel.

Note Cisco no longer recommends using MD5 (including HMAC variant) and Diffie-Hellman (DH) groups 1, 2 and 5; instead, you should use SHA-256 and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

The following table lists the commands that are enabled with the IKEv2 Smart Defaults feature, along with the default values.

Table 1: IKEv2 Command Defaults

Command Name	Default Values

crypto ikev2 authorization policy	`Device# show crypto ikev2 authorization policy default`
	
	`IKEv2 Authorization policy: default`
	`route set interface`
	`route accept any tag: 1 distance: 2`

crypto ikev2 proposal	`Device# show crypto ikev2 proposal`
	
	`IKEv2 proposal: default`
	`Encryption: AES-CBC-256`
	`Integrity: SHA512 SHA384`
	`PRF: SHA512 SHA384`
	`DH Group: DH_GROUP_256_ECP/Group 19 DH_GROUP_2048_MODP/Group 14 DH_GROUP_521_ECP/Group 21 DH_GROUP_1536_MODP/Group 5`

crypto ikev2 policy	`Device# show crypto ikev2 policy default`
	
	`IKEv2 policy: default`
	`Match fvrf: any`
	`Match address local: any`
	`Proposal: default`

crypto ipsec profile	`Device# show crypto ipsec profile default`
	
	`IPSEC profile default`
	`Security association lifetime: 4608000 kilobytes/3600 seconds`
	`Responder-Only (Y/N): N`
	`PFS (Y/N): N`
	`Transform sets={`
	`default: { esp-aes esp-sha-hmac },`
	`}`

crypto ipsec transform-set	`Device# show crypto ipsec transform-set default`
	
	`Transform set default: { esp-aes esp-sha-hmac }`
	`will negotiate = { Tunnel, },`

Note Before you can use the default IPsec profile, explicitly specify the **crypto ipsec profile** command on a tunnel interface using the **tunnel protection ipsec profile default** command.

IKEv2 Suite-B Support

Suite-B is a set of cryptographic algorithms promulgated by the National Security Agency as part of its Cryptographic Modernization Program. Suite-B for Internet Key Exchange (IKE) and IPsec is defined in RFC 4869. The Suite-B components are as follows:

- Advanced Encryption Standard (AES) 128- and 256-bit keys configured in the IKEv2 proposal. For data traffic, AES should be used in Galois Counter Mode (GCM) that is configured in the IPsec transform set.
- Elliptic Curve Digital Signature Algorithm (ECDSA) configured in the IKEv2 profile.
- Secure Hashing Algorithm 2 (SHA-256 and SHA-384) configured in the IKEv2 proposal and IPsec transform set.

Suite-B requirements comprise four user-interface suites of cryptographic algorithms for use with IKE and IPsec. Each suite consists of an encryption algorithm, a digital-signature algorithm, a key-agreement algorithm, and a hash- or message-digest algorithm. See the "Configuring Security for VPNs with IPsec" feature module for detailed information about Cisco Suite-B support.

AES-GCM Support

An authenticated encryption algorithm provides a combined functionality of encryption and integrity. Such algorithms are called combined mode algorithms. The Support of AES-GCM as an IKEv2 Cipher on IOS feature provides the use of authenticated encryption algorithms for encrypted messages in IKEv2 protocol by adding the Advanced Encryption Standard in Galois/Counter Mode (AES-GCM). AES-GCM supports the key size of 128- and 256-bits—AES-GCM-128 and AES-GCM-256.

Note If AES-GCM is the only encryption algorithm, integrity algorithms cannot be added to the proposal.

Auto Tunnel Mode Support in IKEv2

When configuring a VPN headend in a multiple vendor scenario, you must be aware of the technical details of the peer or responder. For example, some devices may use IPsec tunnels while others may use generic routing encapsulation (GRE) or IPsec tunnel, and sometimes, a tunnel may be IPv4 or IPv6. In the last case, you must configure an Internet Key Exchange (IKE) profile and a virtual template.

The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. This feature is useful on dual stack hubs aggregating multivendor remote access, such as Cisco AnyConnect VPN Client, Microsoft Windows7 Client, and so on.

Note The Tunnel Mode Auto Selection feature eases the configuration for a responder only. The tunnel must be statically configured for an initiator.

The Tunnel Mode Auto Selection feature can be activated using the **auto mode** keywords in the **virtual-template** command in the IKEv2 profile configuration.

How to Configure Internet Key Exchange Version 2

Configuring Basic Internet Key Exchange Version 2 CLI Constructs

To enable IKEv2 on a crypto interface, attach an Internet Key Exchange Version 2 (IKEv2) profile to the crypto map or IPsec profile applied to the interface. This step is optional on the IKEv2 responder.

Note The difference between IKEv1 and IKEv2 is that you need not enable IKEv1 on individual interfaces because IKEv1 is enabled globally on all interfaces on a device.

Perform the following tasks to manually configure basic IKEv2 constructs:

Configuring the IKEv2 Keyring

Perform this task to configure the IKEv2 key ring if the local or remote authentication method is a preshared key.

IKEv2 key ring keys must be configured in the peer configuration submode that defines a peer subblock. An IKEv2 key ring can have multiple peer subblocks. A peer subblock contains a single symmetric or asymmetric key pair for a peer or peer group identified by any combination of the hostname, identity, and IP address.

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

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

On an IKEv2 initiator, the IKEv2 key ring key lookup is performed using the peer's hostname or the address, in that order. On an IKEv2 responder, the key lookup is performed using the peer's IKEv2 identity or the address, in that order.

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 keyring** *keyring-name*
4. **peer** *name*
5. **description** *line-of-description*
6. **hostname** *name*
7. **address** {*ipv4-address* [*mask*] | *ipv6-address prefix*}
8. **identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
9. **pre-shared-key** {**local** | **remote**} [**0** | **6**] *line* **hex** *hexadecimal-string*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 keyring** *keyring-name* **Example:** `Device(config)# crypto ikev2 keyring kyr1`	Defines an IKEv2 key ring and enters IKEv2 key ring configuration mode.
Step 4	**peer** *name* **Example:** `Device(config-ikev2-keyring)# peer peer1`	Defines the peer or peer group and enters IKEv2 key ring peer configuration mode.
Step 5	**description** *line-of-description* **Example:** `Device(config-ikev2-keyring-peer)# description this is the first peer`	(Optional) Describes the peer or peer group.

	Command or Action	Purpose
Step 6	**hostname** *name* **Example:** `Device(config-ikev2-keyring-peer)# hostname host1`	Specifies the peer using a hostname.
Step 7	**address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} **Example:** `Device(config-ikev2-keyring-peer)# address 10.0.0.1 255.255.255.0`	Specifies an IPv4 or IPv6 address or range for the peer. **Note** This IP address is the IKE endpoint address and is independent of the identity address.

Step 8	**identity** {**address** {*ipv4-address* | *ipv6-address*} | **fqdn domain** *domain-name* | **email domain** *domain-name* | **key-id** *key-id*}
	Example:
	`Device(config-ikev2-keyring-peer)# identity address 10.0.0.5`
		Identifies the IKEv2 peer through the following identities:
		• E-mail
		• Fully qualified domain name (FQDN) .
		Note When FQDN is used to identify the peer in the keyring configuration, use the IP address of the peer along with the FQDN
		`crypto ikev2 keyring key1`
		`peer headend-1`
		`address 1.1.1.1 >>>>>>>>>`
		`identity fqdn NFVIS-headend-1.cisco.com`
		`pre-shared-key Cisco123`
		• IPv4 or IPv6 address
		• Key ID
		Note The identity is available for key lookup on the IKEv2 responder only.

Step 9	**pre-shared-key** {**local** | **remote**} [**0** | **6**] *line* **hex** *hexadecimal-string* **Example:** `Device(config-ikev2-keyring-peer)# pre-shared-key local key1`	Specifies the preshared key for the peer.
Step 10	**end** **Example:** `Device(config-ikev2-keyring-peer)# end`	Exits IKEv2 key ring peer configuration mode and returns to privileged EXEC mode.

What to Do Next

After configuring the IKEv2 key ring, configure the IKEv2 profile. For more information, see the "Configuring IKEv2 Profile (Basic)" section.

Configuring an IKEv2 Profile (Basic)

Perform this task to configure the mandatory commands for an IKEv2 profile.

An IKEv2 profile is a repository of nonnegotiable parameters of the IKE security association (SA) (such as local or remote identities and authentication methods) and services available to authenticated peers that match the profile. An IKEv2 profile must be configured and associated with either a crypto map or an IPsec profile on the IKEv2 initiator. Use the **set ikev2-profile** *profile-name* command to associate a profile with a crypto map or an IPsec profile. To disassociate the profile, use the **no** form of the command.

The following rules apply to match statements:

- An IKEv2 profile must contain a match identity or a match certificate statement; otherwise, the profile is considered incomplete and is not used. An IKEv2 profile can have more than one match identity or match certificate statements.
- An IKEv2 profile must have a single match Front Door VPN routing and forwarding (FVRF) statement.
- When a profile is selected, multiple match statements of the same type are logically ORed, and multiple match statements of different types are logically ANDed.
- The match identity and match certificate statements are considered to be the same type of statements and are ORed.
- Configuration of overlapping profiles is considered a misconfiguration. In the case of multiple profile matches, no profile is selected.

Use the **show crypto ikev2 profile** *profile-name* command to display the IKEv2 profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **description** *line-of-description*
5. **aaa accounting** {**psk** | **cert** | **eap**} *list-name*
6. **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}
7. **dpd** *interval retry-interval* {**on-demand** | **periodic**}
8. **dynamic**
9. **identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*}
10. **initial-contact force**
11. **ivrf** *name*
12. **keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] }
13. **lifetime** *seconds*
14. **match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*}
15. **nat keepalive** *seconds*
16. **pki trustpoint** *trustpoint-label* [**sign** | **verify**]
17. **virtual-template** *number* **mode auto**
18. **shutdown**

19. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables the privileged EXEC mode. Enter your password, if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters the global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters the IKEv2 profile configuration mode.
Step 4	**description** *line-of-description* **Example:** `Device(config-ikev2-profile)# description This is an IKEv2 profile`	(Optional) Describes the profile.
Step 5	**aaa accounting** {**psk** | **cert** | **eap**} *list-name* **Example:** `Device(config-ikev2-profile)# aaa accounting eap list1`	(Optional) Enables authentication, authorization, and accounting (AAA) accounting method lists for IPsec sessions. **Note** If the **psk**, **cert**, or **eap** keyword is not specified, the AAA accounting method list is used irrespective of the peer authentication method.
Step 6	**authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}} **Example:** `Device(config-ikev2-profile)# authentication local ecdsa-sig`	Specifies the local or remote authentication method. • **rsa-sig**—Specifies RSA-sig as the authentication method. • **pre-share**—Specifies the preshared key as the authentication method. • **ecdsa-sig**—Specifies ECDSA-sig as the authentication method. • **eap**—Specifies EAP as the remote authentication method. • **query-identity**—Queries the EAP identity from the peer. • **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response.

	Command or Action	Purpose
		Note You can specify only one local authentication method but multiple remote authentication methods.
Step 7	**dpd** *interval retry-interval* {**on-demand** | **periodic**} **Example:** `Device(config-ikev2-profile)# dpd 30 6 on-demand`	This step is optional. Configures Dead Peer Detection (DPD) globally for peers matching the profile. By default, the Dead Peer Detection (DPD) is disabled. **Note** 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.
Step 8	**dynamic** **Example:** `Device(config-ikev2-profile)# dynamic`	Configures a dynamic IKEv2 profile. This keyword has been introduced in the Cisco IOS XE 17.2.1 release. **Note** When you configure a dynamic profile, you cannot configure local or remote authentication and identity using the command line interface.
Step 9	**identity local** {**address** {*ipv4-address* | *ipv6-address*} | **dn** | **email** *email-string* | **fqdn** *fqdn-string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# identity local email abc@example.com`	This is an optional step. Specifies the local IKEv2 identity type. **Note** If the local authentication method is a preshared key, the default local identity is the IP address. If the local authentication method is a Rivest, Shamir, and Adleman (RSA) signature, the default local identity is a Distinguished Name.
Step 10	**initial-contact force** **Example:** `Device(config-ikev2-profile)# initial-contact force`	Enforces initial contact processing if the initial contact notification is not received in the IKE_AUTH exchange.
Step 11	**ivrf** *name* **Example:** `Device(config-ikev2-profile)# ivrf vrf1`	This is an optional step. Specifies a user-defined VPN routing and forwarding (VRF) or global VRF if the IKEv2 profile is attached to a crypto map. • If you use the IKEv2 profile for tunnel protection, you must configure the Inside VRF (IVRF) for the tunnel interface on the tunnel interface. **Note** IVRF specifies the VRF for cleartext packets. The default value for IVRF is FVRF.
Step 12	**keyring** {**local** *keyring-name* | **aaa** *list-name* [**name-mangler** *mangler-name* | **password** *password*] }	Specifies the local or AAA-based key ring that must be used with the local and remote preshared key authentication method.

	Command or Action	Purpose
	Example: `Device(config-ikev2-profile)# keyring aaa keyring1 name-mangler mangler1`	**Note** You can specify only one key ring. Local AAA is not supported for AAA-based preshared keys. **Note** Depending on your release, the **local** keyword and the **name-mangler** *mangler-name* keyword-argument pair should be used. **Note** When using AAA, the default password for a Radius access request is "cisco". You can use the **password** keyword within the **keyring** command to change the password.
Step 13	**lifetime** *seconds* **Example:** `Device(config-ikev2-profile)# lifetime 1000`	Specifies the lifetime, in seconds, for the IKEv2 SA.
Step 14	**match** {**address local** {*ipv4-address* | *ipv6-address* | **interface** *name*} | **certificate** *certificate-map* | **fvrf** {*fvrf-name* | **any**} | **identity remote address** {*ipv4-address* [*mask*] | *ipv6-address prefix*} | {**email** [*domain string*] | **fqdn** [*domain string*]} *string* | **key-id** *opaque-string*} **Example:** `Device(config-ikev2-profile)# match address local interface Ethernet 2/0`	Uses match statements to select an IKEv2 profile for a peer.
Step 15	**nat keepalive** *seconds* **Example:** `Device(config-ikev2-profile)# nat keepalive 500`	(Optional) Enables NAT keepalive and specifies the duration in seconds. • By default, NAT is disabled.
Step 16	**pki trustpoint** *trustpoint-label* [**sign** | **verify**] **Example:** `Device(config-ikev2-profile)# pki trustpoint tsp1 sign`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method. **Note** If the **sign** or **verify** keyword is not specified, the trustpoint is used for signing and verification. **Note** In contrast to IKEv1, a trustpoint must be configured in an IKEv2 profile for certificate-based authentication to succeed. There is no fallback for globally configured trustpoints if this command is not present in the configuration. The trustpoint configuration applies to the IKEv2 initiator and responder.
Step 17	**virtual-template** *number* **mode auto** **Example:** `Device(config-ikev2-profile)# virtual-template 1 mode auto`	This is an optional step. Specifies the virtual template for cloning a virtual access interface (VAI). • **mode auto** - Enables the tunnel mode auto selection feature.

	Command or Action	Purpose
		Note For the IPsec Dynamic Virtual Tunnel Interface (DVTI), a virtual template must be specified in an IKEv2 profile, without which an IKEv2 session is not initiated.
Step 18	**shutdown** **Example:** `Device(config-ikev2-profile)# shutdown`	(Optional) Shuts down the IKEv2 profile.
Step 19	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits the IKEv2 profile configuration mode and returns to the privileged EXEC mode.

Configuring Advanced Internet Key Exchange Version 2 CLI Constructs

This section describes the global IKEv2 CLI constructs and how to override the IKEv2 default CLI constructs. IKEv2 smart defaults support most use cases and hence, we recommend that you override the defaults only if they are required for specific use cases not covered by the defaults.

Perform the following tasks to configure advanced IKEv2 CLI constructs:

Configuring Global IKEv2 Options

Perform this task to configure global IKEv2 options that are independent of peers.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 certificate-cache** *number-of-certificates*
4. **crypto ikev2 cookie-challenge** *number*
5. **crypto ikev2 diagnose error** *number*
6. **crypto ikev2 dpd** *interval retry-interval* {**on-demand** | **periodic**}
7. **crypto ikev2 http-url cert**
8. **crypto ikev2 limit** {**max-in-negotiation-sa** *limit* | **max-sa** *limit*}
9. **crypto ikev2 nat keepalive** *interval*
10. **crypto ikev2 window** *size*
11. **crypto logging ikev2**
12. **end**

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 certificate-cache** *number-of-certificates* **Example:** `Device(config)# crypto ikev2 certificate-cache 750`	Defines the cache size for storing certificates fetched from HTTP URLs.
Step 4	**crypto ikev2 cookie-challenge** *number* **Example:** `Device(config)# crypto ikev2 cookie-challenge 450`	Enables an IKEv2 cookie challenge only when the number of half-open security associations (SAs) exceeds the configured number. • Cookie challenge is disabled by default.
Step 5	**crypto ikev2 diagnose error** *number* **Example:** `Device(config)# crypto ikev2 diagnose error 500`	Enables IKEv2 error diagnostics and defines the number of entries in the exit path database. • IKEv2 error diagnostics is disabled by default.
Step 6	**crypto ikev2 dpd** *interval retry-interval* {**on-demand** | **periodic**} **Example:** `Device(config)# crypto ikev2 dpd 30 6 on-demand`	Allows live checks for peers as follows: • Dead Peer Detection (DPD) is disabled by default. **Note** 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.
Step 7	**crypto ikev2 http-url cert** **Example:** `Device(config)# crypto ikev2 http-url cert`	Enables the HTTP CERT support. • HTTP CERT is disabled by default.
Step 8	**crypto ikev2 limit** {**max-in-negotiation-sa** *limit* | **max-sa** *limit*} **Example:**	Enables connection admission control (CAC). • Connection admission control is enabled by default.
Step 9	**crypto ikev2 nat keepalive** *interval* **Example:** `Device(config)# crypto ikev2 nat keepalive 500`	Enables the Network Address Translation (NAT) keepalive that prevents the deletion of NAT entries in the absence of any traffic when there is NAT between Internet Key Exchange (IKE) peers. • NAT keepalive is disabled by default.
Step 10	**crypto ikev2 window** *size*	Allows multiple IKEv2 request-response pairs in transit.

	Command or Action	Purpose
	Example: `Device(config)# crypto ikev2 window 15`	• The default window size is 5.
Step 11	**crypto logging ikev2** **Example:** `Device(config)# crypto logging ikev2`	Enables IKEv2 syslog messages. • IKEv2 syslog messages are disabled by default.
Step 12	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuring IKEv2 Proposal

Refer to the "IKEv2 Smart Defaults" section for information on the default IKEv2 proposal.

Perform this task to override the default IKEv2 proposal or to manually configure the proposals if you do not want to use the default proposal.

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, the default proposal in the default IKEv2 policy is used in negotiation.

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.

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

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

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 proposal** *name*
4. **encryption** *encryption-type...*
5. **integrity** *integrity-type...*
6. **group** *group-type...*
7. **prf** *prf-algorithm*
8. **end**
9. **show crypto ikev2 proposal** [*name* | **default**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 proposal** *name* **Example:** `Device(config)# crypto ikev2 proposal proposal1`	Overrides the default IKEv2 proposal, defines an IKEv2 proposal name, and enters IKEv2 proposal configuration mode.
Step 4	**encryption** *encryption-type...* **Example:** `Device(config-ikev2-proposal)# encryption aes-cbc-128 aes-cbc-192`	Specifies one or more transforms of the encryption type, which are as follows: • **3des** (No longer recommended) • **aes-cbc-128** • **aes-cbc-192** • **aes-cbc-256** • **aes-gcm-128** • **aes-gcm-256**
Step 5	**integrity** *integrity-type...* **Example:** `Device(config-ikev2-proposal)# integrity sha1`	Specifies one or more transforms of the integrity algorithm type, which are as follows: • The **md5** keyword specifies MD5 (HMAC variant) as the hash algorithm. (No longer recommended) • The **sha1** keyword specifies SHA-1 (HMAC variant) as the hash algorithm. • The **sha256** keyword specifies SHA-2 family 256-bit (HMAC variant) as the hash algorithm. • The **sha384** keyword specifies SHA-2 family 384-bit (HMAC variant) as the hash algorithm. • The **sha512** keyword specifies SHA-2 family 512-bit (HMAC variant) as the hash algorithm. **Note** An integrity algorithm type cannot be specified if you specify Advanced Encryption Standard (AES) in Galois/Counter Mode (AES GCM) as the encryption type.
Step 6	**group** *group-type...*	Specifies the Diffie-Hellman (DH) group identifier.

	Command or Action	Purpose
	Example: `Device(config-ikev2-proposal)# group 14`	• The default DH group identifiers are group 2 and 5 in the IKEv2 proposal. • **1**—768-bit DH (No longer recommended). • **2**—1024-bit DH (No longer recommended). • **5**—1536-bit DH (No longer recommended). • **14**—Specifies the 2048-bit DH group. • **15**—Specifies the 3072-bit DH group. • **16**—Specifies the 4096-bit DH group. • **19**—Specifies the 256-bit elliptic curve DH (ECDH) group. • **20**—Specifies the 384-bit ECDH group. • **24**—Specifies the 2048-bit DH group. The group chosen must be strong enough (have enough bits) to protect the IPsec keys during negotiation. A generally accepted guideline recommends the use of a 2048-bit group after 2013 (until 2030). Either group 14 or group 24 can be selected to meet this guideline. Even if a longer-lived security method is needed, the use of Elliptic Curve Cryptography is recommended, but group 15 and group 16 can also be considered.
Step 7	**prf** *prf-algorithm* **Example:** `Device(config-ikev2-proposal)# prf sha256 sha512`	Specifies one or more of the Pseudo-Random Function (PRF) algorithm, which are as follows: • **md5** • **sha1** • **sha256** • **sha384** • **sha512** **Note** This step is mandatory if the encryption type is AES-GCM—**aes-gmc-128** or **aes-gmc-256**. If the encryption algorithm is not AES-GCM, the PRF algorithm is the same as the specified integrity algorithm. However, you can specify a PRF algorithm, if required.
Step 8	**end** **Example:** `Device(config-ikev2-proposal)# end`	Exits IKEv2 proposal configuration mode and returns to privileged EXEC mode.
Step 9	**show crypto ikev2 proposal** [*name* | **default**] **Example:** `Device# show crypto ikev2 proposal default`	(Optional) Displays the IKEv2 proposal.

What to Do Next

After you create the IKEv2 proposal, attach it to a policy so that the proposal is picked for negotiation. For information about completing this task, see the "Configuring IKEv2 Policy" section.

Configuring IKEv2 Policies

See the "IKEv2 Smart Defaults" section for information about the default IKEv2 policy.

Perform this task to override the default IKEv2 policy or to manually configure the policies if you do not want to use the default policy.

An IKEv2 policy must contain at least one proposal to be considered as complete and can have match statements, which are used as selection criteria to select a policy for negotiation. During the initial exchange, the local address (IPv4 or IPv6) and the Front Door VRF (FVRF) of the negotiating SA are matched with the policy and the proposal is selected.

The following rules apply to the match statements:

- An IKEv2 policy without any match statements will match all peers in the global FVRF.
- An IKEv2 policy can have only one match FVRF statement.
- An IKEv2 policy can have one or more match address local statements.
- When a policy is selected, multiple match statements of the same type are logically ORed and match statements of different types are logically ANDed.
- There is no precedence between match statements of different types.
- Configuration of overlapping policies is considered a misconfiguration. In the case of multiple, possible policy matches, the first policy is selected.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 policy** *name*
4. **proposal** *name*
5. **match fvrf** {*fvrf-name* | **any**}
6. **match address local** {*ipv4-address* | *ipv6-address*}
7. **end**
8. **show crypto ikev2 policy** [*policy-name* | **default**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:**	Enters global configuration mode.

	Command or Action	Purpose
	`Device# configure terminal`	
Step 3	**crypto ikev2 policy** *name* **Example:** `Device(config)# crypto ikev2 policy policy1`	Overrides the default IKEv2 policy, defines an IKEv2 policy name, and enters IKEv2 policy configuration mode.
Step 4	**proposal** *name* **Example:** `Device(config-ikev2-policy)# proposal proposal1`	Specifies the proposals that must be used with the policy. • The proposals are prioritized in the order of listing. **Note** You must specify at least one proposal. You can specify additional proposals with each proposal in a separate statement.
Step 5	**match fvrf** {*fvrf-name* | **any**} **Example:** `Device(config-ikev2-policy)# match fvrf any`	(Optional) Matches the policy based on a user-configured FVRF or any FVRF. • The default is global FVRF. **Note** The **match fvrf any** command must be explicitly configured in order to match any VRF. The FVRF specifies the VRF in which the IKEv2 packets are negotiated.
Step 6	**match address local** {*ipv4-address* | *ipv6-address*} **Example:** `Device(config-ikev2-policy)# match address local 10.0.0.1`	(Optional) Matches the policy based on the local IPv4 or IPv6 address. • The default matches all the addresses in the configured FVRF.
Step 7	**end** **Example:** `Device(config-ikev2-policy)# end`	Exits IKEv2 policy configuration mode and returns to privileged EXEC mode.
Step 8	**show crypto ikev2 policy** [*policy-name* | **default**] **Example:** `Device# show crypto ikev2 policy policy1`	(Optional) Displays the IKEv2 policy.

Configuration Examples for Internet Key Exchange Version 2

Configuration Examples for Basic Internet Key Exchange Version 2 CLI Constructs

Example: Configuring the IKEv2 Key Ring

Example: IKEv2 Key Ring with Multiple Peer Subblocks

The following example shows how to configure an Internet Key Exchange Version 2 (IKEv2) key ring with multiple peer subblocks:

```
crypto ikev2 keyring keyring-1
 peer peer1
  description peer1
  address 209.165.200.225 255.255.255.224
  pre-shared-key key-1
 peer peer2
  description peer2
  hostname peer1.example.com
  pre-shared-key key-2
 peer peer3
  description peer3
  hostname peer3.example.com
  identity key-id abc
  address 209.165.200.228 255.255.255.224
  pre-shared-key key-3
```

Example: IKEv2 Key Ring with Symmetric Preshared Keys Based on an IP Address

The following example shows how to configure an IKEv2 key ring with symmetric preshared keys based on an IP address. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer1
  description peer1
  address 209.165.200.225 255.255.255.224
  pre-shared-key key1
```

The following is the responder's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer2
  description peer2
  address 209.165.200.228 255.255.255.224
  pre-shared-key key1
```

Example: IKEv2 Key Ring with Asymmetric Preshared Keys Based on an IP Address

The following example shows how to configure an IKEv2 key ring with asymmetric preshared keys based on an IP address. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
```

```
 peer peer1
  description peer1 with asymmetric keys
  address 209.165.200.225 255.255.255.224
  pre-shared-key local key1
  pre-shared-key remote key2
```

The following is the responder's key ring:

```
crypto ikev2 keyring keyring-1
 peer peer2
  description peer2 with asymmetric keys
  address 209.165.200.228 255.255.255.224
  pre-shared-key local key2
  pre-shared-key remote key1
```

Example: IKEv2 Key Ring with Asymmetric Preshared Keys Based on a Hostname

The following example shows how to configure an IKEv2 key ring with asymmetric preshared keys based on the hostname. The following is the initiator's key ring:

```
crypto ikev2 keyring keyring-1
 peer host1
  description host1 in example domain
  hostname host1.example.com
  pre-shared-key local key1
  pre-shared-key remote key2
```

The following is the responder's keyring:

```
crypto ikev2 keyring keyring-1
 peer host2
  description host2 in abc domain
  hostname host2.example.com
  pre-shared-key local key2
  pre-shared-key remote key1
```

Example: IKEv2 Key Ring with Symmetric Preshared Keys Based on an Identity

The following example shows how to configure an IKEv2 key ring with symmetric preshared keys based on an identity:

```
crypto ikev2 keyring keyring-4
 peer abc
  description example domain
  identity fqdn example.com
  pre-shared-key abc-key-1
 peer user1
  description user1 in example domain
  identity email user1@example.com
  pre-shared-key abc-key-2
 peer user1-remote
  description user1 example remote users
  identity key-id example
  pre-shared-key example-key-3
```

Example: IKEv2 Key Ring with a Wildcard Key

The following example shows how to configure an IKEv2 key ring with a wildcard key:

```
crypto ikev2 keyring keyring-1
 peer cisco
  description example domain
  address 0.0.0.0 0.0.0.0
  pre-shared-key example-key
```

Example: Matching a Key Ring

The following example shows how a key ring is matched:

```
crypto ikev2 keyring keyring-1
 peer cisco
  description example.com
  address 0.0.0.0 0.0.0.0
  pre-shared-key xyz-key
 peer peer1
  description abc.example.com
  address 10.0.0.0 255.255.0.0
  pre-shared-key abc-key
 peer host1
  description host1@abc.example.com
  address 10.0.0.1
  pre-shared-key host1-example-key
```

In the example shown, the key lookup for peer 10.0.0.1 first matches the wildcard key example-key, then the prefix key example-key, and finally the host key host1-example-key. The best match host1-example-key is used.

```
crypto ikev2 keyring keyring-2
 peer host1
 description host1 in abc.example.com sub-domain
 address 10.0.0.1
 pre-shared-key host1-example-key
 peer host2
  description example domain
  address 0.0.0.0 0.0.0.0
  pre-shared-key example-key
```

In the example shown, 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.

Example: Configuring the Profile

Example: IKEv2 Profile Matched on Remote Identity

The following profile supports peers that identify themselves using fully qualified domain name (FQDN) example.com and authenticate with the RSA signature using trustpoint-remote. The local node authenticates itself with a preshared key using keyring-1.

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

Example: IKEv2 Profile Supporting Two Peers

The following example shows how to configure an IKEv2 profile supporting two peers that use different authentication methods:

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

Example: Configuring FlexVPN with Dynamic Routing Using Certificates and IKEv2 Smart Defaults

The following examples show a connection between a branch device (initiator, using a static virtual tunnel interface [sVTI]) and a central device (responder, using a dynamic virtual tunnel interface [dVTI]) with dynamic routing over the tunnel. The example uses IKEv2 smart defaults, and the authentication is performed using certificates (RSA signatures).

Note A RSA modulus size of 2048 is recommended.

The peers use the FQDN as their IKEv2 identity, and the IKEv2 profile on the responder matches the domain in the identity FQDN.

The configuration on the initiator (branch device) is as follows:

```
hostname branch
ip domain name cisco.com
!
crypto ikev2 profile branch-to-central
 match identity remote fqdn central.cisco.com
 identity local fqdn branch.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
!
crypto ipsec profile svti
 set ikev2-profile branch-to-central
!
interface Tunnel0
 ip address 172.16.0.101 255.255.255.0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel destination 10.0.0.100
 tunnel protection ipsec profile svti
!
interface Ethernet0/0
 ip address 10.0.0.101 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.101.1 255.255.255.0
```

```
!
router rip
 version 2
 passive-interface Ethernet1/0
 network 172.16.0.0
 network 192.168.101.0
 no auto-summary
```

The configuration on the responder (central router) is as follows:

```
hostname central
ip domain name cisco.com
!
crypto ikev2 profile central-to-branch
 match identity remote fqdn domain cisco.com
 identity local fqdn central.cisco.com
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint CA
 virtual-template 1
!
interface Loopback0
 ip address 172.16.0.100 255.255.255.0
!
interface Ethernet0/0
 ip address 10.0.0.100 255.255.255.0
!
interface Ethernet1/0
 ip address 192.168.100.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Loopback0
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile default
!
router rip
 version 2
 passive-interface Ethernet1/0
 network 172.16.0.0
 network 192.168.100.0
 no auto-summary
```

Configuration Examples for Advanced Internet Key Exchange Version 2 CLI Constructs

Example: Configuring the Proposal

Example: IKEv2 Proposal with One Transform for Each Transform Type

This example shows how to configure an IKEv2 proposal with one transform for each transform type:

```
crypto ikev2 proposal proposal-1
 encryption aes-cbc-128
 integrity sha1
 group 14
```

Example: IKEv2 Proposal with Multiple Transforms for Each Transform Type

This example shows how to configure an IKEv2 proposal with multiple transforms for each transform type:

```
crypto ikev2 proposal proposal-2
 encryption aes-cbc-128 aes-cbc-192
 integrity sha1
 group 14
```

Note Cisco no longer recommends using 3DES, MD5 (including HMAC variant), and Diffie-Hellman(DH) groups 1, 2 and 5; instead, you should use AES, SHA-256 and DH Groups 14 or higher. For more information about the latest Cisco cryptographic recommendations, see the Next Generation Encryption (NGE) white paper.

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

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

Example: IKEv2 Proposals on the Initiator and Responder

The following example shows how to configure IKEv2 proposals on the initiator and the responder. The proposal on the initiator is as follows:

```
crypto ikev2 proposal proposal-1
 encryption aes-cbc-192 aes-cbc-128
 integrity sha-256 sha1
 group 14 24
```

The proposal on the responder is as follows:

```
crypto ikev2 proposal proposal-2
 encryption aes-cbc-128 aes-cbc-192
 peer
 integrity sha1 sha-256
 group 24 14
```

The selected proposal will be as follows:

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

In the proposals shown for the initiator and responder, the initiator and responder have conflicting preferences. In this case, the initiator is preferred over the responder.

Example: Configuring the Policy

Example: IKEv2 Policy Matched on a VRF and Local Address

The following example shows how an IKEv2 policy is matched based on a VRF and local address:

```
crypto ikev2 policy policy2
 match vrf vrf1
```

```
 match local address 10.0.0.1
 proposal proposal-1
```

Example: IKEv2 Policy with Multiple Proposals That Match All Peers in a Global VRF

The following example shows how an IKEv2 policy with multiple proposals matches the peers in a global VRF:

```
crypto ikev2 policy policy2
 proposal proposal-A
 proposal proposal-B
 proposal proposal-B
```

Example: IKEv2 Policy That Matches All Peers in Any VRF

The following example shows how an IKEv2 policy matches the peers in any VRF:

```
crypto ikev2 policy policy2
 match vrf any
 proposal proposal-1
```

Example: Matching a Policy

Do not configure overlapping policies. If there are multiple possible policy matches, the best match is used, as shown in the following example:

```
crypto ikev2 policy policy1
 match fvrf fvrf1
crypto ikev2 policy policy2
 match fvrf fvff1
 match local address 10.0.0.1
```

The proposal with FVRF as fvrf1 and the local peer as 10.0.0.1 matches policy1 and policy2, but policy2 is selected because it is the best match.

Where to Go Next

After configuring IKEv2, proceed to configure IPsec VPNs. For more information, see the “Configuring Security for VPNs with IPsec” module.

Additional References for Configuring Internet Key Exchange Version 2 (IKEv2)

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	Configuring Security for VPNs with IPsec
Suite-B ESP transforms	Configuring Security for VPNs with IPsec
Suite-B SHA-2 family (HMAC variant) and elliptic curve (EC) key pair configuration	Configuring Internet Key Exchange for IPsec VPNs
Suite-B elliptic curve Diffie-Hellman (ECDH) support for IPsec SA negotiation	Configuring Internet Key Exchange for IPsec VPNs
Suite-B support for certificate enrollment for a PKI	Configuring Certificate Enrollment for a PKI
Supported standards for use with IKE	Internet Key Exchange for IPsec VPNs Configuration Guide
Recommended cryptographic algorithms	Next Generation Encryption

RFCs

RFC	Title
RFC 4306	*Internet Key Exchange (IKEv2) Protocol*
RFC 4869	*Suite B Cryptographic Suites for IPsec*
RFC 5685	*Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)*

Technical Assistance

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

Feature Information for Configuring Internet Key Exchange Version 2 (IKEv2)

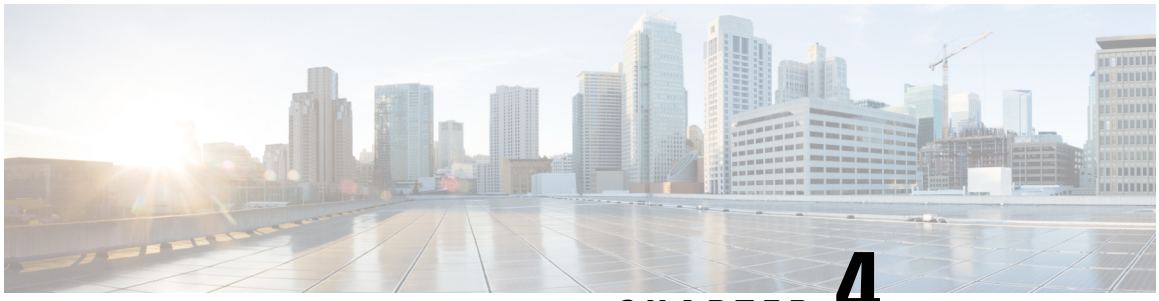
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

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

Table 2: Feature Information for Configuring Internet Key Exchange Version 2 (IKEv2)

Feature Name	Releases	Feature Information
IPv6 Support for IPsec and IKEv2		This feature allows IPv6 addresses to be added to IPsec and IKEv2 protocols. The following commands were introduced or modified: **address (IKEv2 keyring), identity (IKEv2 keyring), identity local, match (IKEv2 policy), match (IKEv2 profile), show crypto ikev2 session, show crypto ikev2 sa, show crypto ikev2 profile, show crypto ikev2 policy, debug crypto condition, clear crypto ikev2 sa.**
Suite-B Support in IOS SW Crypto		Suite-B adds support for the SHA-2 family (HMAC variant) hash algorithm used to authenticate packet data and verify the integrity verification mechanisms for the IKEv2 proposal configuration. HMAC is a variant that provides an additional level of hashing. Suite-B also allows the Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig), as defined in RFC 4754, to be the authentication method for IKEv2. Suite-B requirements comprise of four user interface suites of cryptographic algorithms for use with IKE and IPsec that are described in RFC 4869. Each suite is consists of an encryption algorithm, a digital signature algorithm, a key agreement algorithm, and a hash or message digest algorithm. See the Configuring Security for VPNs with IPsec module for more information about Cisco IOS Suite-B support. The following commands were introduced or modified: **authentication, group, identity (IKEv2 profile), integrity, match (IKEv2 profile).**
Support of AES-GCM as an IKEv2 Cipher on IOS		The AES-GCM Support on IKEv2 feature describes the use of authenticated encryption algorithms with the Encrypted Payload of the Internet Key Exchange version 2 (IKEv2) protocol by adding the Advanced Encryption Standard (AES) in Galois/Counter Mode (AES-GCM). The following commands were introduced or modified: **encryption (IKEv2 proposal), prf, show crypto ikev2 proposal.**

Feature Name	Releases	Feature Information
Tunnel Mode Auto Selection		The Tunnel Mode Auto Selection feature eases the configuration and spares you about knowing the responder's details. This feature automatically applies the tunneling protocol (GRE or IPsec) and transport protocol (IPv4 or IPv6) on the virtual template as soon as the IKE profile creates the virtual access interface. The following commands were introduced or modified: **virtual-template (IKEv2 profile), show crypto ikev2 profile.**



CHAPTER 4

Configuring the FlexVPN Server

This module describes FlexVPN server features, IKEv2 commands required to configure the FlexVPN server, remote access clients, and the supported RADIUS attributes.

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

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restrictions for the FlexVPN Server

Dual-Stack Tunnel Interface and VRF-Aware IPsec

When configuring a dual-stack tunnel interface in a VPN routing and forwarding (VRF)-aware IPsec scenario, you cannot use the **ip vrf forwarding** command to configure an Inside VPN routing and forwarding (IVRF)

instance because this is not a valid configuration. Use the **vrf forwarding** *vrf-name* command to define the IVRF of the tunnel interface, where the *vrf-name* argument is defined using the **vrf definition** command with IPv4 and IPv6 address families inside the definition.

SSO Restrictions

- The Cisco ASR 1000 Series Routers support stateful IPSec sessions on Embedded Services Processor (ESP) switchover. During ESP switchover, all IPSec sessions will stay up and no user intervention is needed to maintain IPSec sessions.
- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.
- The Cisco ASR 1000 Series Router currently does not support Stateful Switchover (SSO) IPSec sessions on Route Processors (RPs). The IPSec sessions will go down on initiation of the switchover, but will come back up when the new RP becomes active. No user intervention is needed. Traffic disruption might happen over the IPSec sessions for the duration of the switchover, until the sessions are back up.
- The Cisco ASR 1000 Series Router does not support stateful ISSU for IPSec sessions. Before performing an ISSU, you must explicitly terminate all existing IPSec sessions or tunnels prior to the operation and reestablish them post ISSU. Specifically, ensure that there are no half-open or half-established IPSec tunnels present before performing ISSU. To do this, we recommend a interface shutdown in the case of interfaces that may initiate a tunnel setup, such as a routing protocol initiating a tunnel setup, or interfaces that have keepalive enabled, or where there is an auto trigger for an IPSec session. Traffic disruption over the IPSec sessions during ISSU is obvious in this case.

Information About the FlexVPN Server

Peer Authentication Using EAP

The FlexVPN server supports peer authentication using the Extensible Authentication protocol (EAP) and acts as a pass-through authenticator relaying EAP messages between the client and the backend EAP server. The backend EAP server is typically a RADIUS server that supports EAP authentication.

Note While a FlexVPN client authenticates the FlexVPN client using EAP, the FlexVPN server must authenticated by using certificates.

The FlexVPN server is configured to authenticate FlexVPN clients that use EAP by configuring the **authentication remote eap** command in IKEv2 profile configuration mode. FlexVPN clients authenticate using EAP by skipping the AUTH payload in the IKE_AUTH request.

If the **query-identity** keyword is configured, the FlexVPN server queries the EAP identity from the client; otherwise, the FlexVPN client's IKEv2 identity is used as the EAP identity. However, if the **query-identity** keyword is not configured and the FlexVPN client's IKEv2 identity is an IPv4 or IPv6 address, the session is terminated because IP addresses cannot be used as the EAP identity.

The FlexVPN server starts the EAP authentication by passing the FlexVPN client's EAP identity to the EAP server; the FlexVPN server then relays EAP messages between the remote access (RA) client and the EAP server until the authentication is complete. If the authentication succeeds, the EAP server is expected to return the authenticated EAP identity to the FlexVPN server in the EAP success message.

After EAP authentication, the EAP identity used for the IKEv2 configuration is obtained from the following sources in the given order:

- The EAP identity provided by the EAP server with the EAP success message.
- The EAP identity queried from the client when the **query-identity** keyword is configured.
- The FlexVPN client IKEv2 identity used as the EAP identity.

The figure below shows IKEv2 exchange for EAP authentication without the **query-identity** keyword.

Figure 1: IKEv2 Exchange Without the query-identity Keyword

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	RADIUS Access-Request/ EAP-Message/EAP-Response/(EAP-ID(IKEv2-ID)) →	
		← RADIUS Access-Challenge/EAP-Message/ EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method)} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (other attributes))
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

209140

The figure below shows the IKEv2 exchange for EAP authentication with the **query-identity** keyword.

Figure 2: IKEv2 Exchange with the query-identity Keyword

IKEv2 RA client	IKEv2 RA server	RADIUS-EAP server
HDR, SAi1, KEi, Ni →		
	← HDR, SAr1, KEr, Nr, [CERTREQ]	
HDR, SK {IDi, [CERTREQ,] [IDr,] SAi2, TSi, TSr} →		
	← HDR, SK {IDr, [CERT,] AUTH, EAP (EAP-request (Identity)) }	
HDR, SK {EAP(EAP-Response(Identity)} →		
	RADIUS Access-Request/ EAP-Message/EAP-Response/(EAP-ID) →	
		← RADIUS Access-Challenge/EAP-Message/ EAP-Request/(EAP-method)
	← HDR, SK {IDr, [CERT,] AUTH, EAP(EAP-Request(EAP-method))}	
HDR, SK {EAP(EAP-Response(EAP-method)} →		
	RADIUS Access-Request/EAP-Message/EAP-Response/EAP-method) →	
		← RADIUS Access-Accept/EAP-Message/EAP-Success (EAP-identity) (other attributes))
	← HDR, SK {EAP (success)}	
HDR, SK {AUTH} →		
	← HDR, SK {AUTH, SAr2, TSi, TSr }	

209141

IKEv2 Configuration Mode

IKEv2 configuration mode allows IKE peers to exchange configuration information such as IP addresses and routes. The configuration information is obtained from IKEv2 authorization. Both pull and push models are supported. The pull model involves the exchange of configuration requests and replies; the push model involves the exchange of configuration sets and acknowledgements.

The following table describes the conditions when the initiator and the responder send different configuration payload types:

Table 3: Configuration Payload Types

Configuration Payload Type	Sent By...	When...
CFG_REQUEST	Initiator	The initiator is the FlexVPN client or if the **config-exchange request** command is enabled in the IKEv2 profile.
CFG_REPLY	Responder	The responder receives the CFG_REQUEST.
CFG_SET	Initiator and responder	Initiator—The **config-exchange set send** command is enabled in the IKEv2 profile. Responder—The CFG_REQUEST is not received, the configuration data is available, and the **config-exchange set send** command is enabled in the IKEv2 profile.
CFG_ACK	Initiator and responder	Initiator—The **config-exchange set accept** command is enabled in the IKEv2 profile. Responder—The **config-exchange set accept** command is enabled in the IKEv2 profile.

Note The commands to send configuration requests and configuration set payloads are enabled by default.

Depending on your release, the IKEv2 initiator can trigger a configuration mode when the initiator is a FlexVPN client, or any static tunnel interface initiating IKEv2 can trigger configuration mode by enabling the **config-mode** command in the IKEv2 profile.

The IKEv2 FlexVPN server supports the following standard IPv4 configuration attributes:

- INTERNAL_IP4_ADDRESS
- INTERNAL_IP4_NETMASK
- INTERNAL_IP4_DNS
- INTERNAL_IP4_NBNS
- INTERNAL_IP4_SUBNET

The IKEv2 FlexVPN server supports the following standard IPv6 configuration attributes:

- INTERNAL_IP6_ADDRESS
- INTERNAL_IP6_DNS
- INTERNAL_IP6_SUBNET

Note IPv6 configuration attributes are only supported by the Microsoft Windows IKEv2 client.

The INTERNAL_IP4_SUBNET and INTERNAL_IP6_SUBNET configuration attributes, controlled by the **route set** and **aaa attribute list** commands in the IKEv2 authorization policy, are not supported when you configure a static virtual tunnel interface (SVTI)-to-SVTI tunnel. In such cases, static routing or dynamic routing must be used instead of the IKEv2-based route exchange.

The IKEv2 FlexVPN server supports the following standard common configuration attribute:

- APPLICATION_VERSION

Note This attribute is only sent for Cisco Anyconnect and FlexVPN clients.

The IKEv2 FlexVPN server supports the following Cisco Unity configuration attributes:

- MODECFG_BANNER
- MODECFG_DEFDOMAIN
- MODECFG_SPLITDNS_NAME
- MODECFG_BACKUPSERVERS
- MODECFG_PFS
- MODECFG_SMARTCARD_REMOVAL_DISCONNECT

Note The Cisco Unity attributes are sent only for Cisco Anyconnect and FlexVPN clients.

The IKEv2 FlexVPN server supports the following Cisco FlexVPN configuration attributes:

- MODECFG_CONFIG_URL
- MODECFG_CONFIG_VERSION

Note The Cisco FlexVPN attributes are sent only for Cisco FlexVPN clients.

The INTERNAL_IP4_ADDRESS attribute value is derived from the following sources in the given order:

- The Framed-IP-Address attribute received in AAA user authorization.
- The local IP address pool.
- The DHCP server.

The DHCP server, if configured, allocates addresses only if the local IP address pool is not configured. However, if an error occurs when allocating IP addresses from the local pool, the next address source DHCP server is not used for allocating the addresses.

The value for INTERNAL_IP4_NETMASK attribute is derived as follows:

- If the IP address is obtained from the DHCP server, the netmask is also obtained from the DHCP server.

- If the IP address is obtained from either the Framed-IP-Address attribute in AAA user authorization or the local IP address pool, the netmask is derived from the IPv4 netmask attribute received in the user or group authorization. If the netmask is not available, the INTERNAL_IP4_NETMASK attribute is not included in the configuration reply. If the netmask is available, the INTERNAL_IP4_NETMASK attribute is included only if the INTERNAL_IP4_ADDRESS attribute is included in the configuration reply.

An IPv4 address is allocated and included in the reply only if the client requests an address. If the client requests multiple IPv4 addresses, only one IPv4 address is sent in the reply. If available, the remaining attributes are included in the reply even though the client does not request them. If the client requests an IPv4 address and the FlexVPN server is unable to assign an address, an INTERNAL_ADDRESS_FAILURE message is returned to the client.

It is always recommended that the prefix length should be used as 128 on ipv6 local pool configuration.

For example, if clients are 4 , **ipv6 local pool pool1 afe0::/126 128** needed to be configured for the prefix length. If clients are 16, **ipv6 local pool pool1 afe0::/124 128** needed to be configured for the prefix length.

IKEv2 Authorization

IKEv2 authorization provides a policy for an authenticated session by using the AAA. The policy can be defined locally or on the RADIUS server, and contains local and/or remote attributes. The username for authorization can either be derived from the peer identity using the **name-mangler** keyword or be directly specified in the command. IKEv2 authorization is mandatory only if the peer requests an IP address via configuration mode.

IKEv2 authorization types are as follows:

- User authorization—Use the **aaa authorization user** command in the IKEv2 profile to enable user authorization. User authorization is based on the user-specific portion of the peer IKE identity such as fqdn-hostname. The attributes from user authorization are called user attributes.
- Group authorization—Use the **aaa authorization group** command in the IKEv2 profile to enable group authorization. Group authorization is based on the generic portion of the peer IKE identity such as fqdn-domain. The attributes from group authorization are called group attributes.
- Implicit user authorization—Use the **aaa authorization user cached** command in the IKEv2 profile to enable implicit user authorization. Implicit authorization is performed as part of EAP authentication or when obtaining the AAA preshared key. The attributes from implicit user authorization are called cached attributes.

Note Depending on your release, the **aaa authorization user cached** command may or may not be available. Explicit user authorization is performed only when implicit user authorization does not return any attributes or does not have the Framed-IP-Address attribute.

Merging and Overriding Attributes

Attributes from different sources are merged before they are used. The precedence of merging attributes is as follows:

- When merging duplicate attributes, the source of the attribute has a higher precedence.
- When merging user and cached attributes, user attributes have higher precedence.

- When merging merged-user-attributes and group attributes, merged-user attributes have a higher precedence, by default. However, this precedence can be reversed using the **aaa author group override** command.

IKEv2 Authorization Policy

An IKEv2 authorization policy defines the local authorization policy and contains local and/or remote attributes. Local attributes, such as VPN routing and forwarding (VRF) and the QOS policy, are applied locally. Remote attributes, such as routes, are pushed to the peer via the configuration mode. Use the **crypto ikev2 authorization policy** command to define the local policy. The IKEv2 authorization policy is referred from the IKEv2 profile via the **aaa authorization** command.

IKEv2 Name Mangler

The IKEv2 name mangler is used to derive the username for IKEv2 authorization and obtain the AAA preshared key from the peer IKE identity.

IKEv2 Multi-SA

The IKEv2 Multi-SA feature allows an IKEv2 Dynamic Virtual Tunnel Interface (DVTI) session on the IKEv2 responder to support multiple IPsec Security Associations (SA). The maximum number of IPsec SAs per DVTI session is either obtained from AAA authorization or configured on the IPsec profile. The value from AAA has a higher priority. Any change to the *max-flow-limit* argument in the IPsec profile is not applied to the current session but is applied to subsequent sessions. The IKEv2 Multi-SA feature makes the configuration of the IKEv2 profile in the IPsec profile optional. This optional configuration allows IPsec DVTI sessions using the same virtual template to have different IKEv2 profiles, thus saving the number of virtual template configurations.

Note The IKEv2 Multi-SA feature allows multiple IPsec SAs that have non-any-any proxies. However, when the IPsec SA proxies are any-any, a single IPsec SA is allowed.

For more information, see the "Multi-SA Support for Dynamic Virtual Tunnel Interfaces for IKEv2" module in the *Security for VPNs with IPsec Configuration Guide*.

Supported RADIUS Attributes

The following tables list the RADIUS attributes supported by the IKEv2 FlexVPN server:

- The Scope field defines the direction of the attribute and the usage on the FlexVPN server or client.
  - Inbound—FlexVPN server to RADIUS
  - Outbound—RADIUS to the FlexVPN server
  - Local—Used locally by the FlexVPN server
  - Remote—Pushed to the client by the FlexVPN server

- The “Local configuration” field specifies the IKEv2 authorization policy command that is used to configure the attribute locally on the FlexVPN server.
- Cisco AV Pair is a Cisco Vendor Specific Attribute (VSA) with vendor-id 9 and vendor-type 1. The VSAs are encapsulated in the Radius IETF attribute 26 Vendor-Specific. The Cisco AV pair is specified as a string of format “protocol:attribute=value”.

Example:

```
cisco-avpair = “ipsec:ipv6-addr-pool=v6-pool”
```

The following example shows the Cisco AV pair for a standard access-list.

```
cisco-avpair = "ipsec:route-set=access-list 99"
```

Table 4: Inbound and Bidirectional IETF RADIUS Attributes

Attribute	Scope
User-Name	Inbound and outbound (bidirectional)
User-Password	Inbound
Calling-Station-Id	Inbound
Service-Type	Inbound
EAP-Message	Bidirectional
Message-Authenticator	Bidirectional

Table 5: Outbound IETF and Cisco AV Pair RADIUS Attributes

Attribute	Type	Scope	Local configuration
Tunnel-Type	IETF	Local	N/A
Tunnel-Medium-Type	IETF	Local	N/A
Tunnel-Password	IETF	Local	N/A
ipsec:ikev2-password-local	Cisco AV Pair	Local	N/A
ipsec:ikev2-password-remote	Cisco AV Pair	Local	N/A
ipsec:addr-pool	Cisco AV Pair	Local	pool
ipsec:group-dhcp-server	Cisco AV Pair	Local	dhcp server
ipsec:dhcp-giaddr	Cisco AV Pair	Local	dhcp giaddr
ipsec:dhcp-timeout	Cisco AV Pair	Local	dhcp timeout
ipsec:ipv6-addr-pool	Cisco AV Pair	Local	ipv6 pool
ipsec:route-set=interface	Cisco AV Pair	Local	route set interface

Attribute	Type	Scope	Local configuration
ipsec:route-set=prefix	Cisco AV Pair	Local	N/A
ipsec:route-accept	Cisco AV Pair	Local	route accept any
ip:interface-config	Cisco AV Pair	Local	aaa attribute list
ipsec:ipsec-flow-limit	Cisco AV Pair	Local	ipsec flow-limit
Framed-IP-Address	IETF	Remote	N/A
Framed-IP-Netmask	IETF	Remote	netmask
ipsec:dns-servers	Cisco AV Pair	Remote	DNS
ipsec:wins-servers	Cisco AV Pair	Remote	wins
ipsec:route-set=access-list (See Note 1.)	Cisco AV Pair	Remote	route set access-list (See Note 1.)
ipsec:addrv6	Cisco AV Pair	Remote	n/a
ipsec:prefix-len	Cisco AV Pair	Remote	n/a
ipsec:ipv6-dns-servers-addr	Cisco AV Pair	Remote	ipv6 dns
ipsec:route-set=access-list ipv6	Cisco AV Pair	Remote	route set access-list ipv6
ipsec:banner	Cisco AV Pair	Remote	banner
ipsec:default-domain	Cisco AV Pair	Remote	def-domain
ipsec:split-dns	Cisco AV Pair	Remote	split-dns
ipsec:ipsec-backup-gateway	Cisco AV Pair	Remote	backup-gateway
ipsec:pfs	Cisco AV Pair	Remote	pfs
ipsec:include-local-lan	Cisco AV Pair	Remote	include-local-lan
ipsec:smartcard-removal-disconnect	Cisco AV Pair	Remote	smartcard-removal- disconnect
ipsec:configuration-url	Cisco AV Pair	Remote	configuration url
ipsec:configuration-version	Cisco AV Pair	Remote	configuration version

Note

- 1. The RADIUS attribute to set an access list on IKEv2 FlexVPN server only supports a standard access list. An extended access list is not supported.

Supported Remote Access Clients

The FlexVPN server interoperates with the Microsoft Windows7 IKEv2 client, Cisco IKEv2 AnyConnect client, and Cisco FlexVPN client.

Microsoft Windows7 IKEv2 Client

The Microsoft Windows 7 IKEv2 client sends an IP address as the Internet Key Exchange (IKE) identity that prevents the Cisco IKEv2 FlexVPN server from segregating remote users based on the IKE identity. To allow the Windows 7 IKEv2 client to send the email address (user@domain) as the IKE identity, apply the hotfix documented in KB975488 (http://support.microsoft.com/kb/975488) on Microsoft Windows 7 and specify the email address string in either the Username field when prompted or the CommonName field in the certificate depending on the authentication method.

For certificate-based authentication, the FlexVPN server and Microsoft Windows 7 client certificates must have an Extended Key Usage (EKU) field as follows:

- For the client certificate, EKU field = client authentication certificate.
- For the server certificate, EKU field = server authentication certificate
- The certificates can be obtained from the Microsoft Certificate Server or the IOS CA server.

For EAP authentication, the Microsoft Windows 7 IKEv2 client expects an EAP identity request before any other EAP requests. Ensure that you configure the **query-identity** keyword in the IKEv2 profile on the IKEv2 FlexVPN server to send an EAP identity request to the client.

Cisco IKEv2 AnyConnect Client

For certificate-based authentication, the FlexVPN server and the AnyConnect client certificates must have an Extended Key Usage (EKU) field as follows:

- For the client certificate, EKU field = client authentication certificate
- For the server certificate, EKU field = server authentication certificate

If the FlexVPN server authenticates to AnyConnect client using certificates, a SubjectAltName extension is required in the FlexVPN server certificate that contains the server's IP address or fully qualified domain name (FQDN). Additionally, HTTP certified URLs must be disabled on the FlexVPN server using the **no crypto ikev2 http-url cert** command.

The following example displays the XML tags specific to EAP-MD5 authentication of IKEv2 sessions in the AnyConnect client profile:

```
<PrimaryProtocol>IPsec
       <StandardAuthenticationOnly>true
           <AuthMethodDuringIKENegotiation>
               EAP-MD5
           </AuthMethodDuringIKENegotiation>
           <IKEIdentity>DEPT24</IKEIdentity>
       </StandardAuthenticationOnly>
</PrimaryProtocol>
```

Note For every flap or FlexVPN tunnel that is enabled, the following message is displayed:

```
*Jan 22 22:52:09.833: %SYS-5-CONFIG_P: Configured programmatically by process Crypto INT
from console as console
*Jan 22 22:52:09.840: %LINEPROTO-5-UPDOWN: Line protocol on Interface Virtual-Access2,
changed state to up
```

For more information, refer to AnyConnect client 3.0 documentation at this link: http://www.cisco.com/en/US/docs/security/vpn_client/anyconnect/anyconnect30/release/notes/anyconnect30rn.html#wp1268255.

How to Configure the FlexVPN Server

Configuring the IKEv2 Profile for the FlexVPN Server

This task describes the IKEv2 profile commands required for configuring the FlexVPN server in addition to the basic IKEv2 profile commands. Refer to the "Configuring IKEv2 Profile (Basic)" task in the *Configuring Internet Key Exchange Version 2 (IKEv2)* feature module for information about configuring the basic IKEv2 profile.

Perform this task to configure the IKEv2 profile for the FlexVPN Server:

Step 1 **enable**

Example:

Enables privileged EXEC mode.

```
Device> enable
```

Enter your password, if prompted.

Step 2 **configure terminal**

Example:

Enters the global configuration mode.

```
Device# configure terminal
```

Step 3 **crypto ikev2 profile** *profile-name*

Defines an IKEv2 profile name and enters IKEv2 profile configuration mode.

Example:

```
Device(config)# crypto ikev2 profile profile1
```

Step 4 **aaa authentication eap** *list-name*

Example:

```
Device(config-ikev2-profile)# aaa authentication eap list1
```

(Optional) Specifies the AAA authentication list for the EAP authentication when implementing the IKEv2 remote access server.

- **eap**—Specifies the external EAP server.
- *list-name*—The AAA authentication list name.

Step 5 **authentication** {**local** {**rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig** | **eap** [**gtc** | **md5** | **ms-chapv2**] [**username** *username*] [**password** {**0** | **6**} *password*}]} | **remote** {**eap** [**query-identity** | **timeout** *seconds*] | **rsa-sig** | **pre-share** [**key** {**0** | **6**} *password*}] | **ecdsa-sig**}}

Example:

```
Device(config-ikev2-profile)# authentication local ecdsa-sig
```

Specifies the local or remote authentication method.

- **rsa-sig**—Specifies RSA-sig as the authentication method.
- **pre-share**—Specifies the preshared key as the authentication method.
- **ecdsa-sig**—Specifies ECDSA-sig as the authentication method.
- **eap**—Specifies EAP as the remote authentication method.
- **query-identity**—Queries the EAP identity from the peer.
- **timeout** *seconds*—Specifies the duration, in seconds, to wait for the next IKE_AUTH request after sending the first IKE_AUTH response.

Note You can specify only one local authentication method but multiple remote authentication methods.

Step 6 Execute both or one of the following:

- **aaa authorization user** {**eap** | **psk**} {**cached** | **list** *aaa-listname* [*aaa-username* | **name-mangler** *mangler-name*]}
- **aaa authorization user cert list** *aaa-listname* {*aaa-username* | **name-mangler** *mangler-name*}

Example:

```
Device(config-ikev2-profile)# aaa authorization user eap cached
```

Example:

```
Device(config-ikev2-profile)# aaa authorization user cert list list1 name-mangler mangler1
```

Specifies the AAA method list and username for user authorization.

- **user**—Specifies user authorization.
- **cert**—Specifies that the peers must be authenticated using certificates.
- **eap**—Specifies that the peers must be authenticated using EAP.
- **psk**—Specifies that the peers must be authenticated using preshared keys.
- **cached**—Specifies that the attributes received during EAP authentication or obtained from the AAA preshared key must be cached.
- *aaa-listname*—AAA method list name.
- *aaa-username*—Specifies the username that must be used in the AAA authorization request.
- **name-mangler**—Specifies the name mangler that derives the AAA authorization username from the peer identity.
- *mangler-name*—Name mangler to be used.

Note
- For **psk** and **eap** authentication methods, specifying the *aaa-username* argument or the **name-mangler** keyword is optional and if not specified, the peer identity is used as the username.
- For **psk** and **eap** authentication methods, you can simultaneously configure two variants for user authorization with the **cached** and **list** keyword respectively.
- Specifying the *aaa-username* argument or the **name-mangler** keyword is mandatory for **cert** authentication, as the peer identity of type distinguished name (DN) cannot be used.

Step 7 Execute both or one of the following:

- **aaa authorization group** [**override**] {**eap** | **psk**} **list** *aaa-listname* [*aaa-username* | **name-mangler** *mangler-name*]
- **aaa authorization group** [**override**] **cert list** *aaa-listname* {*aaa-username* | **name-mangler** *mangler-name*}

Example:

```
Device(config-ikev2-profile)# aaa authorization group override psk list list1
```

Example:

```
Device(config-ikev2-profile)# aaa authorization group cert list list1 name-mangler mangler1
```

Specifies the AAA method list and username for group authorization.

- **group**—Specifies group authorization.
- **override**—(Optional) Specifies that attributes from group authorization should take precedence while merging attributes. By default, user attributes take precedence.
- **cert**—Specifies that peers must be authenticated using certificates.
- **eap**—Specifies that peers must be authenticated using EAP.
- **psk**—Specifies that peers must be authenticated using preshared keys.
- *aaa-listname*—AAA method list name.
- *aaa-username*—Username that must be used in the AAA authorization request.
- **name-mangler**—Specifies the name mangler that derives the AAA authorization username from the peer identity.
- *mangler-name*—Name mangler to be used.

Note
- For **psk** and **eap** authentication methods, specifying the *aaa-username* argument or the **name-mangler** keyword is optional and if not specified, the peer identity is used as the username.
- For **psk** and **eap** authentication methods, you can simultaneously configure two variants for user authorization with the **cached** and **list** keyword respectively.
- Specifying the *aaa-username* argument or the **name-mangler** keyword is mandatory for **cert** authentication, as the peer identity of type distinguished name (DN) cannot be used.

Step 8 **config-exchange** {**request** | **set** {**accept** | **send**}}

Example:

```
Device(config-ikev2-profile)# config-exchange set accept
```

(Optional) Enables configuration exchange options.

- **request**—Enables the configuration exchange request.

- **set**—Enables the configuration exchange request set options.
- **accept**—Accepts the configuration exchange request set.
- **send**—Enables sending of the configuration exchange set.

Note The request and set options are enabled by default.

Step 9 **end**

Example:

```
Device(config-ikev2-profile)# end
```

Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Configuring the IKEv2 Name Mangler

Perform this task to specify the IKEv2 name mangler, which is used to derive a name for authorization requests and obtain AAA preshared keys. The name is derived from specified portions of different forms of remote IKE identities or the EAP identity. The name mangler specified here is referred to in the IKEv2 profile.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 name-mangler** *mangler-name*
4. **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**}
5. **eap** {**all** | **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} | **prefix** | **suffix** {**delimiter** {**.** | **@** | \}}}
6. **email** {**all** | **domain** | **username**}
7. **fqdn** {**all** | **domain** | **hostname**}
8. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 name-mangler** *mangler-name* **Example:** `Device(config)# crypto ikev2 name-mangler mangler1`	Defines a name mangler and enters IKEv2 name mangler configuration mode.

	Command or Action	Purpose
Step 4	**dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} **Example:** `Device(config-ikev2-name-mangler)# dn state`	Derives the name from any of the following fields in the remote identity of type DN (distinguished name). • **common-name** • **country** • **domain** • **locality** • **organization** • **organization-unit** • **state**

Step 5	**eap** {**all** | **dn** {**common-name** | **country** | **domain** | **locality** | **organization** | **organization-unit** | **state**} | **prefix** | **suffix** {**delimiter** {**.** | **@** | \}}}
	Example:
	`Device(config-ikev2-name-mangler)# eap prefix delimiter @`
		Derives the name from the remote identity of type EAP (Extensible Authentication Protocol).
		• **all**—Derives the name from the entire EAP identity.
		• **dn**—Derives the name from any of the following fields in the remote EAP identity of type DN:
		  • **common-name**
		  • **country**
		  • **domain**
		  • **locality**
		  • **organization**
		  • **organization-unit**
		  • **state**
		• **prefix**—Derives the name from the prefix in the EAP identity.
		• **suffix**—Derives the name from the suffix in the EAP identity.
		• **delimiter** {**.** | **@** | \}—Specifies the delimiter in the EAP identity that separates the prefix and the suffix.

Step 6	**email** {**all** | **domain** | **username**} **Example:** `Device(config-ikev2-name-mangler)# email username`	Derives the name from the remote identity of type e-mail. • **all**—Derives the name from the entire remote IKE identity of type e-mail. • **domain**—Derives the name from the domain part of the remote IKE identity. • **username**—Derives the name from the username part of the remote IKE identity.

	Command or Action	Purpose
Step 7	**fqdn** {**all** | **domain** | **hostname**} **Example:** `Device(config-ikev2-name-mangler)# fqdn domain`	Derives the name from the remote identity of type FQDN (Fully Qualified Domain Name). • **all**—Derives the name from the entire remote IKE identity of type FQDN. • **domain**—Derives the name from the domain part of the remote IKE identity. • **hostname**—Derives the name from the hostname part of the remote IKE identity.
Step 8	**end** **Example:** `Device(config-ikev2-name-mangler)# end`	Exits IKEv2 name mangler configuration mode and returns to privileged EXEC mode.

Configuring the IKEv2 Authorization Policy

Perform this task to configure the IKEv2 authorization policy.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 authorization policy** *policy-name*
4. **aaa attribute list** *list-name*
5. **backup-gateway** *string*
6. **banner** *banner-text*
7. **configuration url** *url*
8. **configuration version** *version*
9. **def-domain** *domain-name*
10. **dhcp** {**giaddr** *ip-address* \| **server** {*ip-address* \| *hostname*} \| **timeout** *seconds*}
11. [**ipv6**] **dns** *primary-server* [*secondary-server*]
12. **include-local-lan**
13. **ipsec flow-limit** *number*
14. **netmask** *mask*
15. **pfs**
16. [**ipv6**] **pool** *name*
17. **route set** {**interface** *interface* \| **access-list** {*access-list-name* \| *access-list-number* \| **ipv6** *access-list-name*}}
18. **route accept any** [**tag** *value*] [**distance** *value*]
19. **route redistribute** *protocol* [**route-map** *map-name*]
20. **route set remote** {**ipv4** *ip-address mask* \| **ipv6** *ip-address/mask*}
21. **smartcard-removal-disconnect**
22. **split-dns** *string*
23. **session-lifetime** *seconds*

24. **route set access-list** {*acl-number* | [**ipv6**] *acl-name*}
25. **wins** *primary-server* [*secondary-server*]
26. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 authorization policy** *policy-name* **Example:** `Device(config)# crypto ikev2 authorization policy policy1`	Specifies the IKEv2 authorization policy and enters IKEv2 authorization policy configuration mode.
Step 4	**aaa attribute list** *list-name* **Example:** `Device(config-ikev2-author-policy)# aaa attribute list list1`	Specifies an AAA attribute list. **Note** The AAA attribute list referred to in this command should be defined in global configuration mode.
Step 5	**backup-gateway** *string* **Example:** `Device(config-ikev2-author-policy)# backup-gateway gateway1`	Allows you to specify up to ten backup server names. This parameter is pushed to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies the backup servers that the client can use.
Step 6	**banner** *banner-text* **Example:** `Device(config-ikev2-author-policy)# banner This is IKEv2`	Specifies the banner. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute.
Step 7	**configuration url** *url* **Example:** `Device(config-ikev2-author-policy)# configuration url http://www.cisco.com`	Specifies the configuration URL. This parameter is sent to the client via the nonstandard Cisco FlexVPN configuration attribute. The client can use this URL to download the configuration.
Step 8	**configuration version** *version* **Example:** `Device(config-ikev2-author-policy)# configuration version 2.4`	Specifies the configuration version. This parameter is sent to the client via the nonstandard Cisco FlexVPN configuration attribute. This parameter is sent with the configuration URL to specify the version that the client can download.
Step 9	**def-domain** *domain-name* **Example:**	Specifies the default domain. This parameter is sent to the client via the nonstandard Cisco Unity configuration

	Command or Action	Purpose
	`Device(config-ikev2-author-policy)# def-domain cisco`	attribute. This parameter specifies the default domain that the client can use.
Step 10	**dhcp** {**giaddr** *ip-address* | **server** {*ip-address* | *hostname*} | **timeout** *seconds*} **Example:** `Device(config-ikev2-author-policy)# dhcp giaddr 192.0.2.1`	Specifies the DHCP server to lease an IP address that is assigned to the remote access client. • **giaddr** *ip-address*—Specifies the gateway IP address (giaddr). • **server** {*ip-address* | *hostname*}—Specifies the IP address or hostname of the DHCP server. The hostname is resolved during configuration. • **timeout** *seconds*—Specifies the wait time in seconds for the response from the DHCP server. **Note** You can specify only one DHCP server. It is assumed that the DHCP server can be reached via the global routing table, and therefore, the DHCP packets are forwarded to the global routing table.
Step 11	[**ipv6**] **dns** *primary-server* [*secondary-server*] **Example:** `Device(config-ikev2-author-policy)# dns 198.51.100.1 198.51.100.100`	Specifies the IP addresses of primary and secondary Domain Name Service (DNS) servers that are sent to the client in the configuration reply. • **ipv6**—(Optional) Specifies an IPv6 address for the DNS server. To specify an IPv4 address, execute the command without this keyword. • *primary-server*—IP address of the primary DNS server. • *secondary-server*—(Optional) IP address of the secondary DNS server.
Step 12	**include-local-lan** **Example:** `Device(config-ikev2-author-policy)# include-local-lan`	Includes local LAN. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute.
Step 13	**ipsec flow-limit** *number* **Example:** `Device(config-ikev2-author-policy)# ipsec flow-limit 12500`	Specifies the maximum number of IPsec SAs that an IKev2 dVTI session on the IKev2 responder can have. The range is from 0 to 50000. By default, the command is disabled, and there is no limit on the number of IPsec flows per dVTI session. A value of 0 will not allow any IPsec SAs.
Step 14	**netmask** *mask* **Example:**	Specifies the netmask of the subnet from which the IP address is assigned to the client. • *mask*—Subnet mask address.

	Command or Action	Purpose
	`Device(config-ikev2-author-policy)# netmask 255.255.255.0`	
Step 15	**pfs** **Example:** `Device(config-ikev2-author-policy)# pfs`	Enables Password Forward Secrecy (PFS). This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies whether the client should use PFS.
Step 16	[**ipv6**] **pool** *name* **Example:** `Device(config-ikev2-author-policy)# pool abc`	Defines a local IP address pool for assigning IP addresses to the remote access client. • **ipv6**—(Optional) Specifies an IPv6 address pool. To specify an IPv4 address, execute the command without this keyword.. • *name*—Name of the local IP address pool. **Note** The local IP address pool must already be defined using the **ip local pool** command.
Step 17	**route set** {**interface** *interface* | **access-list** {*access-list-name* | *access-list-number* | **ipv6** *access-list-name*}} **Example:** `Device(config-ikev2-author-policy)# route set interface`	Specifies the route set parameters to the peer via configuration mode and allows running routing protocols such as Border Gateway Protocol (BGP) over VPN. • **interface**—Specifies the route interface. • **access-list**—Specifies the route access list. • *access-list-name*—Access list name. • *access-list-number*—Standard access list number. • **ipv6**—Specifies an IPv6 access list.
Step 18	**route accept any** [**tag** *value*] [**distance** *value*] **Example:** `Device(config-ikev2-author-policy)# route accept any tag 10`	Filters the routes received from the peer and specify the tag and metric values to install these routes. • **any**—Accepts all routes received from the peer. • **tag** *value*—(Optional) Specifies the tag ID for the static routes added by IKEv2. The range is from 1 to 497777. • **distance** *value*—(Optional) Specifies the distance for the static routes added by IKEv2. The range is from 1 to 255.
Step 19	**route redistribute** *protocol* [**route-map** *map-name*] **Example:** `Device(config-ikev2-author-policy)# route redistribute connected`	Filters the routes received from the peer and specify the tag and metric values to install these routes. • *protocol*—Source protocol from which routes are redistributed. It can be one of the following keywords: connected or static.

	Command or Action	Purpose
		• **route-map** *map-name*—(Optional) Route map that should be filtered to import routes from one source routing protocol to another routing protocol. If a map name is not specified, all routes are redistributed.
Step 20	**route set remote** {**ipv4** *ip-address mask* | **ipv6** *ip-address/mask*} **Example:** `Device(config-ikev2-author-policy)# route set remote ipv6 2001:DB8::1/32`	Configures IP addresses of inside networks.
Step 21	**smartcard-removal-disconnect** **Example:** `Device(config-ikev2-author-policy)# smartcard-removal-disconnect`	Enables smartcard removal disconnect. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies that the client should terminate the session when the smart card is removed.
Step 22	**split-dns** *string* **Example:** `Device(config-ikev2-author-policy)# split-dns abc1`	Allows you to specify up to ten split domain names. This parameter is sent to the client via the nonstandard Cisco Unity configuration attribute. This parameter specifies the domain names that the client should use for private networks.
Step 23	**session-lifetime** *seconds* **Example:** `Device(config-ikev2-author-policy)# session-lifetime 1000`	Specifies the IKEv2 session lifetime. • **seconds** *seconds*—The range is from 120 to 25920000, which converts to two minutes to 300 days.
Step 24	**route set access-list** {*acl-number* | [**ipv6**] *acl-name*} **Example:** `Device(config-ikev2-client-config-group)# route set access-list 110`	Specifies the subnets that are pushed to the remote peer via configuration mode. • *acl-number*—Access list number (ACL). The ACL number can only be specified for an IPv4 ACL. • **ipv6**—(Optional) Specifies an IPv6 access control list (ACL). To specify an IPv4 attribute, execute the command without this keyword. • *acl-name*—Access list name. **Note** You can only specify standard, simple access lists for IPv4 addresses.
Step 25	**wins** *primary-server* [*secondary-server*] **Example:** `Device(config-ikev2-author-policy)# wins 203.0.113.1 203.0.113.115`	Specifies the internal Windows Internet Naming Service (WINS) server addresses that are sent to the client in the configuration reply. • *primary-server*—IP address of the primary WINS server. • *secondary-server*—(Optional) IP address of the secondary WINS server.

	Command or Action	Purpose
Step 26	**end** **Example:** `Device(config-ikev2-author-policy)# end`	Exits IKEv2 authorization policy configuration mode and returns to privileged EXEC mode.

Configuration Examples for the FlexVPN Server

Example: Configuring the FlexVPN Server

Example: Configuring the FlexVPN Server to Authenticate Peers Using EAP

This example shows how to configure the FlexVPN server to authenticate peers using EAP.

```
aaa new-model
!
aaa group server radius eap-server
 server 192.168.2.1
!
aaa authentication login eap-list group eap-server
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto ikev2 profile ikev2-profile1
 match identity remote address 0.0.0.0
 authentication local rsa-sig
 authentication remote eap query-identity
 pki trustpoint trustpoint1
 aaa authentication eap eap-list
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.1 key key1
!
```

Example: Configuring the FlexVPN Server for Group Authorization (External AAA)

The following example shows how to configure the FlexVPN server for group authentication through an external AAA, which would be the RADIUS or TACACS server.

```
aaa new-model
!
aaa group server radius cisco-acs
 server 192.168.2.2
!
aaa authorization network group-author-list group cisco-acs
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 name-mangler group-author-mangler
 dn domain
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list group-author-list name-mangler group-author-mangler
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.2 key key2
!
```

Example: Configuring the FlexVPN Server for Group Authorization (Local AAA)

The following example shows how to configure the FlexVPN server for group authorization through the local AAA using the IKEv2 authorization policy. The authorization policy specifies standard IPv4 and IPv6 attributes, and Cisco Unity, and FlexVPN attributes to be sent to the client through configuration mode. The authorization policy also specifies per user attributes through **aaa attribute list** command for local use.

```
aaa new-model
!
aaa authorization network local-group-author-list local
!
!
aaa attribute list attr-list1
 attribute type interface-config "ip mtu 1100"
 attribute type interface-config "tunnel key 10"
!

crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
```

```
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 authorization policy author-policy1
 pool pool1
 dhcp server 192.168.4.1
 dhcp timeout 10
 dhcp giaddr 192.168.1.1
 dns 10.1.1.1 10.1.1.2
 route set access-list acl1
 wins 192.168.1.2 192.168.1.3
 netmask 255.0.0.0
 banner ^C flexvpn server ^C
 configuration url http://www.abc.com
 configuration version 10
 def-domain abc.com
 split-dns dns1
 split-dns dns2
 split-dns dns3
 backup-gateway gw1
 backup-gateway gw2
 backup-gateway gw3
 smartcard-removal-disconnect
 include-local-lan
 pfs
 aaa attribute list attr-list1
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list local-group-author-list author-policy1
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
ip local pool pool11 192.168.2.10 192.168.2.100
!
ip access-list extended acl-1
 permit ip 192.168.3.10 192.168.4.100 any
 permit ip 192.168.10.1 192.168.10.100 any
!
```

Example: Configuring the FlexVPN Server for User Authorization

The following example shows how to configure the FlexVPN server for user authentication.

```
aaa new-model
```

```
!
aaa group server radius cisco-acs
 server 192.168.2.2
!
aaa authorization network user-author-list group cisco-acs
!
crypto pki trustpoint trustpoint1
 enrollment url http:// 192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 name-mangler user-author-mangler
 dn common-name
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization user cert list user-author-list name-mangler user-author-mangler
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ip address 192.168.1.1 255.255.255.0
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile ipsec-profile1
!
radius-server host 192.168.2.2 key key2
!
```

Example: Configuring the FlexVPN Server for IPv6 Session with IPv6 Configuration Attributes

The following example shows how to configure the FlexVPN server for an IPv6 dynamic Virtual Tunnel Interfaces (dVTI) session. The example uses local AAA group authorization using the IKEv2 authorization policy. The IPv6 configuration attributes are configured under the IKEv2 authorization policy.

```
aaa new-model
!
aaa authorization network local-group-author-list local
!
crypto pki trustpoint trustpoint1
 enrollment url http://192.168.3.1:80
 revocation-check crl
!
crypto pki certificate map certmap1 1
 subject-name co cisco
!
crypto ikev2 authorization policy author-policy1
 ipv6 pool v6-pool
 ipv6 dns 2001:DB8:1::11 2001:DB8:1::12
 ipv6 subnet-acl v6-acl
```

```
!
crypto ikev2 profile ikev2-profile1
 match certificate certmap1
 authentication local rsa-sig
 authentication remote rsa-sig
 pki trustpoint trustpoint1
 aaa authorization group cert list local-group-author-list author-policy1
 virtual-template 1
!
crypto ipsec transform-set transform1 esp-aes esp-sha-hmac
!
crypto ipsec profile ipsec-profile1
 set transform-set trans transform1
 set ikev2-profile ikev2-profile1
!
interface Ethernet0/0
 ipv6 address 2001:DB8:1::1/32
!
interface Virtual-Template1 type tunnel
 ipv6 unnumbered Ethernet0/0
 tunnel mode ipsec ipv6
 tunnel protection ipsec profile ipsec-profile1
!
ipv6 local pool v6-pool 2001:DB8:1::10/32 48
!
ipv6 access-list v6-acl
 permit ipv6 host 2001:DB8:1::20 any
 permit ipv6 host 2001:DB8:1::30 any
!
```

Example: Configuring AnyConnect Profile Download

The following example shows how to configure the FlexVPN AnyConnect Profile Download feature:

Note You do not modify the Local Policy files on the Anyconnect Client machine. After the configuration of Anyconnect Profile Download feature on IKEv2, the required XML profiles get automatically downloaded on the client device.

Note You should disable either the HTTPS server (ip http secure-server) or SSL policy (crypto ssl policy) for the profile download feature, otherwise, if both these features are enabled at the same time and the device receives an incoming SSL VPN connection, the device may crash.

```
no ip http secure-server
crypto ssl policy ssl-policy
  pki trustpoint CA1 sign
  ip address local 10.0.0.1 port 443
  no shutdown
crypto ssl profile ssl_prof
    match policy ssl-policy
crypto vpn anyconnect profile ANY-PROF bootflash:profile.xml
crypto ikev2 profile ikev2_profile
    anyconnect profile ANY-PROF
```

Additional References for Configuring the FlexVPN Server

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
Cisco AnyConnect Secure Mobility Client	https://www.cisco.com/c/en/us/support/security/anyconnect-secure-mobility-client/tsd-products-support-series-home.html
IPsec configuration	*Configuring Security for VPNs with IPsec*
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

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

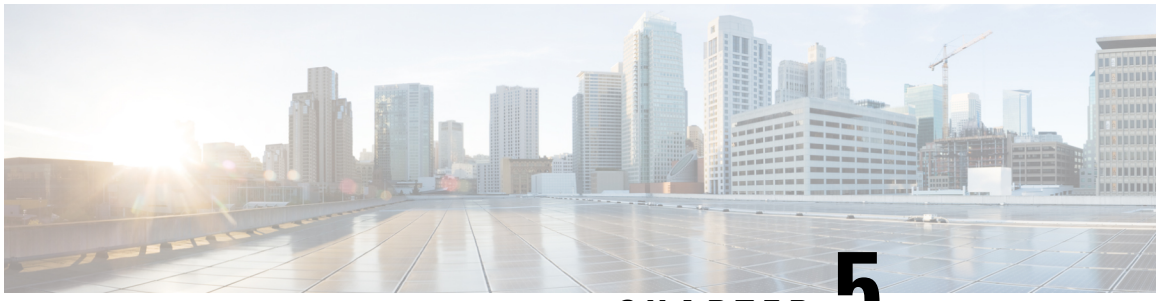
Feature Information for Configuring the FlexVPN Server

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

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

Table 6: Feature Information for Configuring the FlexVPN Server

Feature Name	Releases	Feature Information
IKEv2 headend support for remote access clients	Cisco IOS XE Release 3.5S	This features provides IKEv2 support for Anyconnect 3.0, FlexVPN hardware client, and multi SA support for VTI. The following commands were introduced or modified: **aaa attribute list, backup-gateway, banner, config-mode set, configuration url, configuration version, def-domain, dhcp, dns, include-local-lan, max flow limit, pfs, pool, route accept, route set interface, smartcard-removal-disconnect, split-dns, subnet-acl.**



CHAPTER 5

Configuring the FlexVPN Client

This module describes the FlexVPN client features and the Internet Key Exchange Version 2 (IKEv2) commands required to configure the FlexVPN client.

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.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Restrictions for the FlexVPN Client

EAP as the Local Authentication Method

- Extensible Authentication Protocol (EAP) as the local authentication method, is supported only on the IKEv2 initiator, and as the remote authentication, is supported only on the IKEv2 responder.

- If EAP is specified as the local authentication method, the remote authentication method must be certificate based.
- If the **authentication remote eap query-identity** command is not configured on the FlexVPN server, the client cannot have an IPv4 or IPv6 address as the local identity because these IP addresses cannot be used as the username for the EAP authentication method.

Dual-Stack Tunnel Interface and VRF-Aware IPsec

When configuring a dual-stack tunnel interface in a VPN routing and forwarding (VRF)-aware IPsec scenario, you cannot use the **ip vrf forwarding** command to configure an Inside VPN routing and forwarding (IVRF) instance because this is not a valid configuration. Use the **vrf forwarding** *vrf-name* command to define the IVRF of the tunnel interface, where the *vrf-name* argument is defined using the **vrf definition** command with IPv4 and IPv6 address families inside the definition.

SSO Restrictions

- The Cisco ASR 1000 Series Routers support stateful IPSec sessions on Embedded Services Processor (ESP) switchover. During ESP switchover, all IPSec sessions will stay up and no user intervention is needed to maintain IPSec sessions.
- For an ESP reload (no standby ESP), the SA sequence number restarts from 0. The peer router drops packets that do not have the expected sequence number. You may need to explicitly reestablish IPSec sessions to work around this issue for systems that have a single ESP after an ESP reload. Traffic disruption might happen over the IPSec sessions in such cases for the duration of the reload.
- The Cisco ASR 1000 Series Router currently does not support Stateful Switchover (SSO) IPSec sessions on Route Processors (RPs). The IPSec sessions will go down on initiation of the switchover, but will come back up when the new RP becomes active. No user intervention is needed. Traffic disruption might happen over the IPSec sessions for the duration of the switchover, until the sessions are back up.
- The Cisco ASR 1000 Series Router does not support stateful ISSU for IPSec sessions. Before performing an ISSU, you must explicitly terminate all existing IPSec sessions or tunnels prior to the operation and reestablish them post ISSU. Specifically, ensure that there are no half-open or half-established IPSec tunnels present before performing ISSU. To do this, we recommend a interface shutdown in the case of interfaces that may initiate a tunnel setup, such as a routing protocol initiating a tunnel setup, or interfaces that have keepalive enabled, or where there is an auto trigger for an IPSec session. Traffic disruption over the IPSec sessions during ISSU is obvious in this case.

Information About the FlexVPN Client

IKEv2 FlexVPN Client

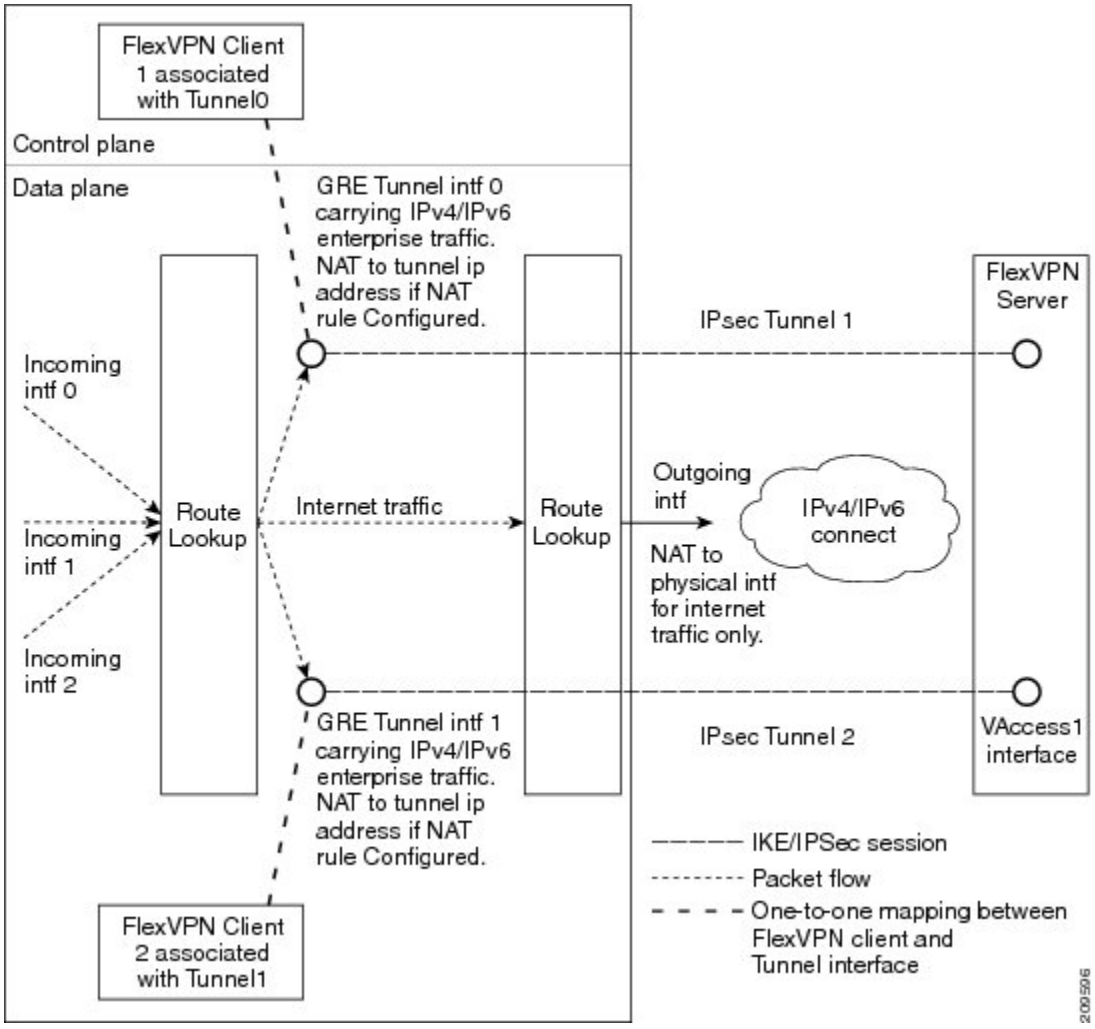
The IKEv2 FlexVPN Client feature establishes a secure IPsec VPN tunnel between a FlexVPN client and a FlexVPN server. The IKEv2 FlexVPN Client feature provides the following benefits:

- Unified tunnel infrastructure
- IPv4/IPv6 proxy support over IPv4/IPv6 transport
- Backward compatibility with some features supported by EasyVPN

- Flexibility for running dynamic routing protocols

Each FlexVPN client is associated with a unique tunnel interface, which implies that the IPsec security association (SA) retrieved by the specific FlexVPN client is bound to the tunnel interface. The figure below shows the association between the FlexVPN client and the tunnel interface.

Figure 3: Association of the FlexVPN Client and the Tunnel Interface



The sequence of operation is as follows:

- Routing—The FlexVPN server pushes the network list as part of the mode configuration response. The client adds routes on the tunnel interface to these networks. As part of the configuration mode set, the client sends the routes to its network. The IP address is configured on the tunnel interface so that the server can add routes to the client-side network.
- NAT—Network Address Translation (NAT) rules must be configured explicitly using route maps. If the rules match, the hosts behind the FlexVPN client are translated to the tunnel IP address. This IP address can be obtained as one of the attributes pushed during mode configuration by the FlexVPN server.
- Encapsulation and encryption—Generic routing encapsulation (GRE) and IPsec encapsulation modes are supported. GRE supports both IPv4 and IPv6 traffic. The traffic that reaches the tunnel interface is

encapsulated by the GRE header, followed by IPsec protection. The encrypted traffic is then routed to the outgoing interface.

The features supported by the FlexVPN client are described in the following sections:

Tunnel Activation

The FlexVPN client can be connected automatically or manually through user intervention. The FlexVPN client connects automatically to the tunnel when the FlexVPN configuration is complete. If the tunnel times out or fails, the tunnel automatically reconnects and retries the connection indefinitely. To configure an automatic tunnel connection, use the **connect** command with the **auto** keyword in the IKEv2 FlexVPN profile.

In a manual connection, the FlexVPN client waits for user intervention to execute a command before establishing a connection. When the client times out or fails to connect, subsequent connections require user intervention. To configure a manual connection, use the **crypto ikev2 client flexvpn connect** command with the *flexvpn-name* argument in privileged EXEC mode. To terminate the connection, use the **clear crypto ikev2 client flexvpn connect** command with the *flexvpn-name* argument.

Tracking-Based Tunnel Activation

The Tracking-Based Tunnel Activation feature is mainly used in backup scenarios. The FlexVPN client registers with the tracking system to obtain notifications for change in the state of an object. This notification prompts the client to perform an appropriate action for tunnel activation. The **track** keyword in the **connect** command informs the tracking process that the client is interested in tracking an object, which is identified by an object number. The tracking process, in turn, informs the client when the state of the objects changes.

If the **track** keyword in the **connect** command is set to activate the tunnel when the object goes up, the client triggers the connection upon receiving the notification that the object is in the UP state. If the **track** keyword in the **connect** command is set to activate the tunnel when the object goes down, the client triggers the connection upon receiving the notification that the object is in the DOWN state.

Backup Features

A FlexVPN client can connect to various peers or servers in a predetermined order. The list of peers is called the gateway list or backup gateway list and is built using the following lists:

- Static backup gateway list or static list
- Downloaded backup gateway list or downloaded list

The static backup gateway list is configured in the FlexVPN profile by providing a list of peers with a sequence number. The downloaded backup gateway list is downloaded dynamically and is obtained during the mode configuration response. The downloaded list complements the static gateway list to build the backup gateway list. The downloaded list is inserted after the peer from which the list is downloaded.

If an existing connection with a peer from the gateway list goes down, the client tries to establish a connection with the next peer in the gateway list. If a downloaded list is available and connection with a static peer fails, the client tries to connect, in sequence, with the peers from the downloaded list. If the client fails to establish a connection with all the peers in the downloaded list, the client tries to connect to the next peer in the static list, and the downloaded list is deleted.

Backup Gateways

Use the **peer** command to add a peer to the backup gateway list. To remove the backup gateway list, use the **no peer** command.

Peers are ordered by preference; the lower the sequence number, the higher the preference.

If a connection is established with a new peer and the peer is not a part of the downloaded list, the peer adds the downloaded list to the backup gateway list, and the existing backup gateway list is replaced with the new list.

You can configure a static peer and attach it to a track object. A peer is a "possible peer" if the track object of the peer is in the UP state.

Note Peers that are not attached to a track object, including peers in the downloaded list, are classified as "possible peers" because these peers are always in the UP state.

The peer selection process works as follows: when a connection is established, the gateway list is looked up and the first possible peer is selected. A peer is selected according to the following rule: a static peer can be associated with the track object with a desired status (UP or DOWN). If the status of the track object matches the configured status, the peer is said to be a "possible peer."

Note If the peer is identified by either a Domain Name Service (DNS) name or a fully qualified domain name (FQDN), the name is resolved dynamically.

The peer selection process is followed by the selection of a new peer or when the existing criteria fail, which happens in the following scenarios:

- The active peer stops responding to liveness checks.
- The DNS resolution of the peer name fails.
- The IKE negotiation with the peer fails.
- The peer is no longer a "possible peer" (its corresponding track object goes DOWN).

Note When you configure multiple FlexVPN peers on a FlexVPN client and when you clear the IKEv2 SA on the primary peer, the clearance will trigger a new peer selection on the client.

Reactivate Primary Peer

The Reactivate Primary Peer feature ensures that the highest-priority peer is always connected. If the track object of the highest-priority peer matches the object status, the existing connection with the lower-priority peer is disconnected, and the connection to the higher-priority peer is established. Use the **peer reactivate** command to enable this feature.

Note A track object must be associated with statically configured peers.

Dial Backup (Primary or Backup Tunnels)

The FlexVPN client registers with the tracking system to get notifications about the change in the state of the object. The **connect track** command is used to inform the tracking process that the client is interested in tracking an object, which is identified by the object number. The tracking process, in turn, informs the client when the state of this objects changes. This notification prompts the client to take further action to bring up or bring down the primary or backup connections when the state of the tracked object is UP or DOWN.

The Dial Backup feature can be configured as follows:

- When both primary and backup tunnels are FlexVPN tunnels,
  - Any one tunnel is active at a time.
  - Both client profiles are configured using the **connect track** command, referencing the same track object.
  - If the primary tunnel tracks the status when the object is UP, the secondary tunnel tracks the status of the object when the object is DOWN.
- When one tunnel is the FlexVPN tunnel,
  - The remaining tunnels can be on any secured connection.
  - The primary connection is not FlexVPN, and the backup connection is FlexVPN.
  - The client profile is configured using the **connect track** command with an object, which traces the ability to reach the primary peer through the primary outgoing interface.

Backup Group

The Backup Group feature allows the FlexVPN client to omit a peer when a FlexVPN client that belongs to a group has established a session with the same peer. When a FlexVPN client belonging to a group initiates a connection with a peer, the FlexVPN client validates if another FlexVPN client in the same group has established a session with the same peer. If a connection exists, the FlexVPN client omits this peer and validates the next peer in the sequence. Use the **backup group** command with the *group-number* argument to configure the backup group.

Dual FlexVPN Support

The Dual FlexVPN Support feature provides the ability to configure two FlexVPN tunnels that share the same inside and outside interfaces. The two FlexVPN tunnels use route injections to direct appropriate traffic through the corresponding tunnel interface. When the tunnel is up, the tunnel “learns” the network list from the server. If the server forwards a network list, FlexVPN installs specific routes to the destination networks in its routing table, directing the traffic to these networks out of the tunnel interface.

Note Only one FlexVPN connection can be established with a default route through the tunnel interface.

Split DNS Support

The Split DNS functionality enables the FlexVPN client to act as a Domain Name System (DNS) proxy. During FlexVPN negotiations, the DNS list is downloaded during mode configuration. This list is configured as a DNS view list on the inside interfaces associated with the FlexVPN profile. The view list is used to match requests based on the domain names with the DNS query and then forward the match requests to the DNS server. Other DNS queries are used to match the default view (global DNS configuration) and are forwarded to the ISP DNS.

If no inside interfaces are mentioned in the FlexVPN client profile, the DNS view is applied to all interfaces except the tunnel interface and the tunnel source interfaces of all configured profiles. When the DNS query request reaches the inside interface, the matching DNS view is obtained, and the request is forwarded to the DNS IP address.

NAT

The Network Address Translation (NAT) feature in FlexVPN enables traffic to be translated to an IP address based on the interface to which the traffic is routed. If a packet is received on one interface that is configured with the **ip nat inside** command and is being sent out another interface that is configured with the **ip nat outside** command, the packet is translated to the IP address configured on the second interface.

Network List from the Server

Routes for enterprise traffic are dynamically installed by a client through the tunnel interface. The traffic takes the default route via the outgoing physical interface. The enterprise traffic is translated to the tunnel IP address, and the Internet traffic is translated to the external outgoing interface IP address.

Default Route List from the Server

A default route must be configured on the device with the higher sequence number via the tunnel interface. The tunnel interface is configured with the **ip nat outside** command, and the IP address of the tunnel interface is assigned by the IP address sent by the client. The enterprise traffic from inside interfaces is translated to the sent address. NAT is achieved by configuring NAT rules with the help of route maps. The route maps define rules based on the outgoing interface, by which the globally configured NAT rules are applied based on routing.

IPv4 traffic going out the tunnel interface is translated to the sent IPv4 address.

Note If NAT is not required, NAT rules associated with the tunnel interface must not be configured.

How the FlexVPN Client learns about the Network List

The FlexVPN client learns about the list of networks behind a peer in one of the following ways:

- Mode configuration push—The FlexVPN server sends the list of network attributes as a configuration mode parameter to the client. The FlexVPN client installs the routes to these networks through the tunnel interface that has the highest metric. The client also communicates its networks to the server in the mode configuration set or acknowledgment (SET/ACK) exchange so that the server can add those routes via the virtual access interface.
- Running routing protocols—The FlexVPN client and server run routing protocols over the tunnel interface to establish network routes, which allows the client and the server the flexibility to add or remove networks without disconnecting the existing session. The tunnel addresses are communicated during mode configuration to establish routes with peers.

WINS NBNS and DOMAIN Name

The FlexVPN server pushes the domain name, Windows Internet Naming Service (WINS), or NetBios Name Server (NBNS) attributes during mode configuration. These attributes are dynamically updated to the DHCP server that runs on the FlexVPN client.

Event Tracing

The Event Tracing feature is used for debugging purposes. Events posted to the FlexVPN client are logged, and the information is used for debugging. Event tracing is a combination of a fast mechanism that logs a few bytes of trace information in a buffer area and a display mechanism that extracts and decodes the debug data. The FlexVPN client maintains its buffer and can be enabled during normal operation.

Extensible Authentication Protocol as a Local Authentication Method

The FlexVPN client supports EAP as a local authentication method. Supported EAP authentication methods are Microsoft Challenge Handshake Authentication Protocol Version 2 (MSCHAPv2), message digest algorithm 5 (MD5), and Generic Token Card (GTC). The EAP authentication process is as follows:

- Use the **authentication local eap** command in IKEv2 profile configuration mode to authenticate the FlexVPN client by using EAP.
- After the FlexVPN client receives the IKE_AUTH response from the peer, enter the **crypto eap credentials** command.
- If the EAP-Identity Request is received in the IKE_AUTH response, the EAP username and password must be specified.
- If an EAP-Identity Request is not received in the IKE_AUTH response, only the password is specified because the local IKEv2 identity is used as the username.

Note EAP as the local authentication method must be used with the FlexVPN client, but EAP can also be used on the IKEv2 initiator. If the EAP server initially proposes an unsupported authentication method, the FlexVPN EAP initiator responds with an EAP Negative Acknowledgment (NAK) packet, requesting EAP-MSCHAPv2, EAP-MD5, or EAP-GTC as the desired authentication method. The FlexVPN EAP responder selects one of the authentication methods.

How to Configure the FlexVPN Client

Configuring the IKEv2 VPN Client Profile

This task describes the IKEv2 commands required for configuring the FlexVPN client and the basic IKEv2 commands. Refer to the “Configuring Basic Internet Key Exchange Version 2 CLI Constructs” task in the *Configuring Internet Key Exchange Version 2 (IKEv2)* module for information about configuring the basic IKEv2 profile.

Note When you enter a typo in authorization list under ikev2 profile, it automatically goes back to the default list.

Refer to the “How to Configure the FlexVPN Client” section for information about configuring an IKEv2 profile for the FlexVPN server.

Configuring the Tunnel Interface

Perform this task to configure the tunnel interface that is referred to by the FlexVPN client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface tunnel** *number*
4. **ip address** {*ipv4-address* | **negotiated**}
5. **tunnel mode gre ip**
6. **tunnel mode ipsec ipv4**
7. **tunnel source** {*ip-address* | *interface* | **dynamic**}
8. **tunnel destination dynamic**
9. **tunnel protection ipsec-profile** *profile-name*
10. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface tunnel** *number* **Example:** `Device(config)# interface tunnel 1`	Creates a tunnel interface and enters interface configuration mode.
Step 4	**ip address** {*ipv4-address*	**negotiated**} **Example:** `Device(config-if)# ip address negotiated`	(Optional) Assigns an IPv4 address to the tunnel interface.
Step 5	**tunnel mode gre ip** **Example:** `Device(config-if)# tunnel mode gre ip`	(Optional) Enables generic route encapsulation (GRE) mode for the tunnel interface.
Step 6	**tunnel mode ipsec ipv4** **Example:** `Device(config-if)# tunnel mode ipsec ipv4`	(Optional) Enables IPsec encapsulation.
Step 7	**tunnel source** {*ip-address*	*interface*	**dynamic**} **Example:** `Device(config-if)# tunnel source 10.0.0.1`	Specifies the source for the tunnel interface.

	Command or Action	Purpose
Step 8	**tunnel destination dynamic** **Example:** `Device(config-if)# tunnel destination dynamic`	Specifies the destination for the tunnel interface.
Step 9	**tunnel protection ipsec-profile** *profile-name* **Example:** `Device(config-if)# tunnel protection ipsec-profile ipsecprofile1`	Associates a tunnel interface with an IPsec profile.
Step 10	**end** **Example:** `Device(config-if)# end`	Exits interface configuration mode and returns to privileged EXEC mode.

Configuring the FlexVPN Client

Use the **monitor event-trace flexvpn** command to enable event tracing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 client flexvpn** *client-name*
4. **peer** *sequence* {*ipv4-address* | *ipv6-address* | **fqdn** *fqdn-name* [**dynamic** | **ipv6**]} [**track** *track-number* [**up** | **down**]]
5. **connect** {**manual** | **auto** | **track** *track-number* [**up** | **down**]}
6. **client inside** *interface-type interface-number*
7. **client connect tunnel** *interface-number*
8. **source** *sequence-number interface-type interface-number* **track** *track-number*
9. **peer reactivate**
10. **backup group** {*group-number* | **default**}
11. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 client flexvpn** *client-name* **Example:**	Defines an IKEv2 FlexVPN client profile and enters IKEv2 FlexVPN client profile configuration mode.

	Command or Action	Purpose
	`Device(config)# crypto ikev2 client flexvpn client1`	
Step 4	**peer** *sequence* {*ipv4-address* | *ipv6-address* | **fqdn** *fqdn-name* [**dynamic** | **ipv6**]} [**track** *track-number* [**up** | **down**]] **Example:** `Device(config-ikev2-flexvpn)# peer 1 10.0.0.1`	Defines a static peer using an IP address or hostname.
Step 5	**connect** {**manual** | **auto** | **track** *track-number* [**up** | **down**]} **Example:** `Device(config-ikev2-flexvpn)# connect track 10 up`	Connects the FlexVPN tunnel. **Note** Any change to this command terminates the active session.
Step 6	**client inside** *interface-type interface-number* **Example:** `Device(config-ikev2-flexvpn)# client inside GigabitEthernet 0/1`	(Optional) Specifies the inside interface. • You can specify more than one inside interface in a FlexVPN client profile. The inside interfaces can be shared across FlexVPN client profiles. **Note** Any change to this command terminates the active session.
Step 7	**client connect tunnel** *interface-number* **Example:** `Device(config-ikev2-flexvpn)# client connect tunnel 1`	Assigns the tunnel interface created in the "Configuring the Tunnel Interface" task to the FlexVPN client. • You can configure only one tunnel interface for a FlexVPN client profile. **Note** Any change to this command terminates the active session.
Step 8	**source** *sequence-number interface-type interface-number* **track** *track-number* **Example:** `Device(config-ikev2-flexvpn)# source 1 GigabitEthernet 0/1 track 11`	Adds sequence numbers to the tunnel source address. • The tunnel source address has the lowest sequence number for which the track object number is in UP state. **Note** Any change to this command terminates the active session.
Step 9	**peer reactivate** **Example:** `Device(config-ikev2-flexvpn)# peer reactivate`	Enables the reactivate primary peer feature.
Step 10	**backup group** {*group-number* | **default**} **Example:** `Device(config-ikev2-flexvpn)# backup group default`	Assigns the client to a backup group. • By default, all clients belong to backup group 0.

	Command or Action	Purpose
		Note Any change to this command terminates the active session.
Step 11	**end** **Example:** `Device(config-ikev2-flexvpn)# end`	Exits IKEv2 FlexVPN client profile configuration mode and returns to privileged EXEC mode.

Configuring EAP as the Local Authentication Method

Perform this task to configure Extensible Authentication Protocol (EAP) as the local authentication method on the FlexVPN client.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **authentication local eap**
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**authentication local eap** **Example:** `Device(config-ikev2-profile)# authentication local eap`	Specifies EAP as the local authentication method. **Note** This command is supported only on the IKEv2 initiator.
Step 5	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Configuration Examples for the FlexVPN Client

Example: Configuring the IKEv2 FlexVPN Client Profile

The following example shows how to configure the IKEv2 FlexVPN client profile:

```
crypto ikev2 client flexvpn flex
  peer 1 10.0.0.1
  connect manual
  client connect Tunnel0
!
crypto ikev2 authorization policy flex
 subnet-acl 199
 route set interface
 route accept any
!
crypto ikev2 keyring key
 peer dvti
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco
 !
crypto ikev2 profile prof
 match identity remote address 10.0.0.1 255.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring key
 aaa authorization group psk list local-group-author-list flex
 config-mode set
!
crypto ipsec transform-set trans esp-aes
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set pfs group2
 set ikev2-profile prof
!
interface Tunnel0
 ip address negotiated
 tunnel source Ethernet0/0
 tunnel destination dynamic
 tunnel mode ipsec ipv4
 tunnel protection ipsec-profile ipsecprof
!
interface Ethernet0/0
 ip address 172.16.0.1 255.240.0.0
 ip virtual-reassembly in
!
 ip route 0.0.0.0 0.0.0.0 2.2.2.2
access-list 199 permit ip 10.20.20.20 0.0.0.255 any
access-list 199 permit ip 10.30.30.30 0.0.0.255 any
```

Example: Configuring EAP as a Local Authentication Method

The following example shows how to configure EAP as a local authentication method:

```
crypto ikev2 profile profile1
 authentication remote rsa-sig
 authentication local eap
```

When the session is brought up, a prompt appears to enter the EAP credentials, as follows:

```
Enter the command "crypto eap credentials profile1"
Device# crypto eap credentials profile1

Enter the Username for profile profile1: cisco
Enter the password for username cisco
```

Additional References for Configuring the FlexVPN Client

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
IPsec configuration	*Configuring Security for VPNs with IPsec*
Recommended cryptographic algorithms	Next Generation Encryption

Technical Assistance

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

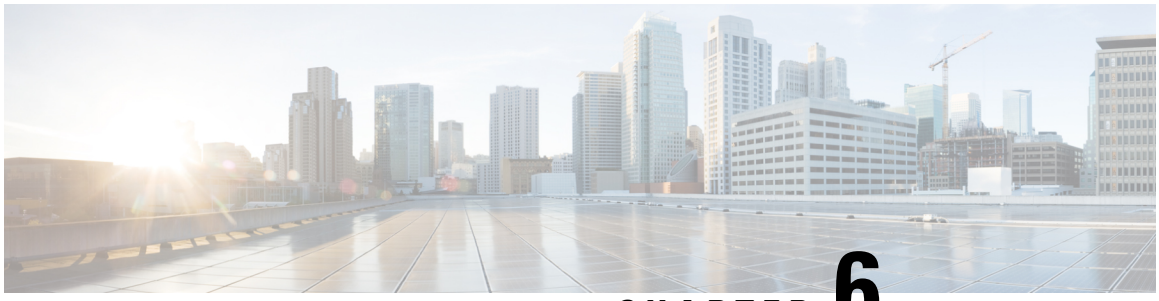
Feature Information for Configuring the FlexVPN Client

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

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

Table 7: Feature Information for Configuring FlexVPN Client

Feature Name	Releases	Feature Information
IKEv2 Remote Access Hardware Client		The IKEv2 Remote Access Hardware Client feature provides support for remote access connectivity and the extensions necessary to support diverse solutions such as mobility, NAT traversal, reliability, and enhanced denial of service (DoS) attack resilience. The following commands were introduced or modified: **backup group, client connect tunnel, client inside, connect, crypto ikev2 client flexvpn, interface, ip address, peer, peer reactivate, source tunnel destination, tunnel mode, tunnel protection, tunnel source.**
IPv6 Remote Access for IPsec VPN		The IPv6 Remote Access for IPsec VPN feature provides IPv6 support and support for EAP as the local authentication method for the IKEv2 FlexVPN client. The following commands were modified: **authentication (IKEv2 profile), peer.**



CHAPTER 6

Configuring IKEv2 Load Balancer

The IKEv2 Load Balancer feature provides support for enabling clusters of FlexVPN gateways and distributes incoming Internet Key Exchange Version 2 (IKEv2) connection requests among FlexVPN gateways. This feature redirects the incoming FlexVPN or AnyConnect client requests to the least loaded FlexVPN gateway based on the system and crypto load factors.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for IKEv2 Load Balancer

- For the server-side configuration, the Hot Standby Router Protocol (HSRP) and FlexVPN server (IKEv2 profile) must be configured.
- For the client-side configuration, the FlexVPN client must be configured.

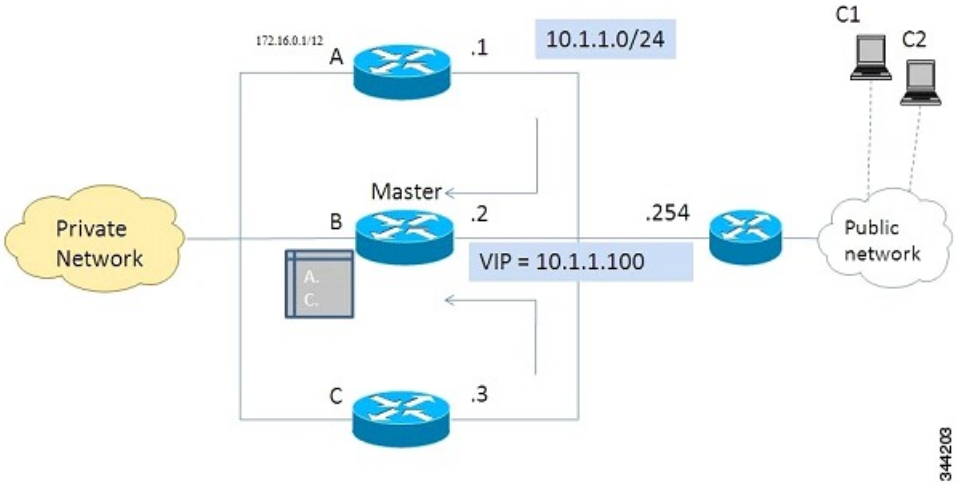
Information About IKEv2 Load Balancer

Overview of IKEv2 Load Balancer

The IKEv2 Load Balancer Support feature provides a Cluster Load Balancing (CLB) solution by redirecting requests from remote access clients to the Least Loaded Gateway (LLG) in the Hot Standby Router Protocol (HSRP) group or cluster. An HSRP cluster is a group of gateways or FlexVPN servers in a LAN or in an enterprise network. The CLB solution works with the Internet Key Exchange Version 2 (IKEv2) redirect mechanism defined in RFC 5685 by redirecting requests to the LLG in the HSRP cluster.

The figure below shows the working of the IKEv2 cluster load balancing solution.

Figure 4: IKEv2 Cluster Load Balancing Solution



1. An active HSRP gateway is elected as “primary” in the HSRP group and takes ownership of the Virtual IP address (VIP) for the group. The primary maintains a list of gateways in the cluster, keeps track of the load on each gateway, and redirects the FlexVPN client requests to the LLG.
2. The remaining gateways, termed as “subordinates,” send load updates to the primary at periodic intervals.
3. When an IKEv2 client connects to the HSRP VIP, the request first reaches the primary, which in turn, redirects the request to the LLG in the cluster.

The components of the CLB solution are as follows:

- HSRP
- CLB primary
- CLB subordinate
- CLB communication
- IKEv2 redirects mechanism

Hot Standby Router Protocol

Hot Standby Router Protocol (HSRP) is used to elect the primary HSRP or Active Router (AR). For HSRP to elect a designated device, you must configure the VIP for one device in the group. This address is learned by other devices in the group. The IP address that is assigned to the primary is used as the VIP for the group. The HSRP active router (also called primary CLB) receives the IKEv2 requests and redirects these requests to the LLG in the cluster. The redirection is performed at the IKEv2 protocol level thereby achieving the following:

- All requests from the FlexVPN client reach the primary HSRP as the VIP is configured on the FlexVPN clients. The configuration of FlexVPN clients is minimized because the FlexVPN clients must only know the VIP of the HSRP cluster.
- The primary CLB is run on the same gateway as the primary HSRP, thereby maintaining the load information of all subordinate CLBs. The primary CLB enables effective redirection of requests and avoids multiple redirects and loops.

Primary CLB

A primary CLB runs on the primary HSRP or Active Router (AR). The primary receives updates from subordinate CLBs and sorts them based on their load condition to calculate the least loaded gateway (LLG). The primary sends the IP address of the LLG to IKEv2 (on the FlexVPN server). The IP address is sent to the initiator (FlexVPN client), which initiates an IKEv2 session with the LLG. The primary redirects incoming IKEv2 client connections towards the LLG. For more information, see section "IKEv2 Redirect Mechanism, on page 84."

Note "CLB nodes" are used where both a primary CLB and CLB subordinate must be specified.

Subordinate CLB

A CLB subordinate runs on all devices in an HSRP group except on the Active Router (AR). The subordinates are responsible for sending periodic load updates to the server. A CLB subordinate is a fully functional IKEv2 gateway which supplies information to the primary CLB. Apart from updates, CLB subordinates send messages for aliveness assurance to the primary CLB.

CLB Load Management Mechanism

The CLB Load Management Mechanism is a TCP-based protocol that runs between the primary CLB and the CLB subordinates. The CLB load management mechanism informs the primary CLB about the load on the CLB subordinates. Based on this information, the primary CLB selects the LLG to handle the session on each new incoming IKEv2 connection.

Benefits of IKEv2 Load Balancer

- The IKEv2 Load Balancer Support feature is easy to configure and cost-effective.
- A FlexVPN client need not know the IP addresses of all gateways in the cluster. The client need only know the virtual IP address of the cluster.
- The entire crypto session is redirected to a node in the cluster.

IKEv2 Redirect Mechanism

The IKEv2 redirect mechanism enables a VPN gateway to redirect a FlexVPN client request to another VPN gateway based on load conditions and maintenance requirements.

The IKEv2 redirect mechanism is performed on security association (SA) initialization (IKE_SA_INIT) and on SA authentication (IKE_AUTH).

Redirect During IKEv2 Initial Exchange (SA Initialization)

A FlexVPN client, or an AnyConnect client indicates support for Internet Key Exchange Version 2 (IKEv2) redirect mechanism by including a REDIRECT_SUPPORTED notification message in the initial IKE_SA_INIT request. Use the **crypto ikev2 redirect client** command to enable the redirect mechanism on a client. Use the **crypto ikev2 redirect gateway init** command to enable redirect at IKE_SA_INIT on the gateway.

To redirect an IKEv2 request to another new gateway, the gateway that receives the IKE_SA_INIT request selects the IP address or the fully qualified domain name (FQDN) of the new gateway (in this case, the LLG) with help of the crypto load balancer (CLB) module. The gateway replies with an IKE_SA_INIT response that contains a REDIRECT notification message. The notification includes information such as the new gateway and the nonce value from the payload in the IKE_SA_INIT request. When a client receives the IKE_SA_INIT response, it verifies the nonce value sent in the IKE_SA_INIT request and the gateway information provided in the redirect notification, and confirms whether the redirect notification is as per the configuration.

Note If the nonce value does not match, the client discards the response and waits for another response, thereby preventing denial of service (DoS) attacks on the initiator. DoS attacks could be caused by an attacker injecting incorrect redirect payloads in IKE_SA_INIT responses.

In the IKE_SA_INIT exchange with the new gateway, the client message contains the REDIRECTED_FROM notification payload. The REDIRECTED_FROM notification payload consists of the IP address of the original VPN gateway that redirected the client. The IKEv2 exchange then proceeds as it would have proceeded with the original gateway.

Note The client may be redirected again by the new gateway if the new gateway also cannot serve the client. The client does not include the REDIRECT_SUPPORTED payload again in the IKE_SA_INIT exchange with the new gateway after the redirect. The presence of the REDIRECTED_FROM notification payload in the IKE_SA_INIT exchange with the new gateway indicates to the new gateway that the client supports the IKEv2 redirects mechanism.

Redirect During IKE_AUTH Exchange (SA Authentication)

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. Use the **crypto ikev2 redirect gateway auth** command to enable the redirect mechanism on the gateway. Use the **redirect gateway auth** command to enable redirect on authentication for selected IKEv2 profiles.

In this method, the client authorization payload is verified before sending the redirect notification payload. A client also verifies the gateway authorization payload before acting on the redirect notification. As the

authorization payload is exchanged and successfully verified, the IKEv2 security association (SA) is validated successfully and the INITIAL_CONTACT is processed to decide on redirecting the request. If there is a redirect, the gateway creates the IKE SA and sends the IKE_AUTH response with the redirect notification.

A child SA is not created in this method. The IKE_AUTH does not contain a payload pertaining to a child SA. When the client receives the IKE_AUTH response, the client verifies the gateway authentication payload and deletes the IKEv2 SA with the gateway by sending a delete notification. The client acts on the redirect notification payload to establish connection with the new gateway. The client does not wait for an acknowledgment for the delete notification before establishing a connection with the new gateway. If the IKE_AUTH exchange involves the Extensible Authentication Protocol (EAP) authentication, the gateway has the choice of sending the redirect payload in the first or last IKE_AUTH response. The EAP authentication is included in the first IKE_AUTH response because it is not necessary to provide credentials for each redirect.

Compatibility and Interoperability

The IKEv2 redirect mechanism is based on RFC 5685. The gateway (IKEv2 responder) is compatible with clients (IKEv2 initiator) that implement the standard. Similarly, the client (initiator) implementation must be compatible with third party servers (responder) implementing the standard. The load management mechanism is Cisco proprietary and is only supported on Cisco IOS devices.

Handling Redirect Loops

A client request could be redirected multiple times in a sequence because of either an incorrect configuration or a denial of service (DoS) attack. In some cases, a client could enter a loop with two or more gateways redirecting the client to the other gateway thereby denying service to the client. To prevent this, a client can be configured, using the **crypto ikev2 redirect client** command with the **max-redirects** *number* keyword argument pair, to not accept more than a specific number of redirects for a particular IKEv2 security association (SA) setup.

IKEv2 Cluster Reconnect

The IKEv2 cluster reconnect feature allows Cisco AnyConnect client to reconnect to any server in the cluster. The **crypto ikev2 reconnect key** is introduced on the server to encrypt the opaque data pushed to the client. During failure detection, the client does reconnect with new or existing server without having to prompt for authentication credentials again.

There are only two key index values, 1 and 2 and at any point in time, any one of the keys configured using this will be active. The Cisco IOS server will be able to decrypt the reconnect data as long as the reconnect key is configured using the reconnect key CLI on the IOS server. This is true even if the key is only the back-up key.

This feature does not support when the **anyconnect-eap** keyword as authentication method in the IKEv2 profile through the **authentication** command.

Note This feature is available on Cisco IOS devices configured to work as Cisco AnyConnect server. The AnyConnect client software version that supports this feature are 4.2 and future releases. This feature is applicable for new deployments only. Once this feature is enabled on the Cisco IOS server, older releases of Cisco AnyConnect clients will not be supported.

How to Configure IKEv2 Load Balancer

Configuring the Server Cluster

Configuring an HSRP Group for Load Balancing

Perform this task to configure a single Hot Standby Router Protocol (HSRP) group for a cluster.

Hot Standby Router Protocol (HSRP) is used to elect the primary HSRP or Active Router (AR). For HSRP to elect a designated device, you must configure the VIP for one device in the group. This address is learned by other devices in the group. The IP address that is assigned to the primary is used as the VIP for the group. The HSRP active router (also called primary CLB) receives the IKEv2 requests and redirects these requests to the LLG in the cluster. The redirection is performed at the IKEv2 protocol level thereby achieving the following:

- All requests from the FlexVPN client reach the primary HSRP as the VIP is configured on the FlexVPN clients. The configuration of FlexVPN clients is minimized because the FlexVPN clients must only know the VIP of the HSRP cluster.
- The primary CLB is run on the same gateway as the primary HSRP, thereby maintaining the load information of all CLB subordinates. The CLB primary enables effective redirection of requests and avoids multiple redirects and loops.

Note This task describes the minimum commands required to configure an HSRP group for load balancing.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **interface** *type number*
4. **ip address** *ip-address mask* [**secondary**]
5. **standby** [*group-number*] **priority** *priority*
6. **standby** *group-name*
7. **exit**
8. Repeat Steps 3 to 7 to configure an HSRP group for another cluster.

DETAILED STEPS

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

	Command or Action	Purpose
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**interface** *type number* **Example:** `Device(config)# interface GigabitEthernet 0/0/0`	Configures an interface type and enters interface configuration mode.
Step 4	**ip address** *ip-address mask* [**secondary**] **Example:** `Device(config-if)# ip address 10.0.0.1 255.255.255.0`	Specifies a primary or secondary IP address for an interface.
Step 5	**standby** [*group-number*] **priority** *priority* **Example:** `Device(config-if)# standby 1 priority 110`	Configures the HSRP priority.
Step 6	**standby** *group-name* **Example:** `Device(config-if)# standby group1`	Specifies the name of the HSRP standby group.
Step 7	**exit** **Example:** `Device(config-if)# exit`	Exits to global configuration mode.
Step 8	Repeat Steps 3 to 7 to configure an HSRP group for another cluster.	—

Configuring the Load Management Mechanism

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 cluster**
4. **holdtime** *milliseconds*
5. **master** {**overload-limit** *percent* | **weight** {**crypto-load** *weight-number* | **system-load** *weight-number*}}
6. **port** *port-number*
7. **slave** {**hello** *milliseconds* | **max-session** *number* | **priority** *number* | **update** *milliseconds*}
8. **standby-group** *group-name*
9. **shutdown**
10. **exit**
11. **crypto ikev2 reconnect key** *key index* **active** *name*
12. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 cluster** **Example:** `Device(config)# crypto ikev2 cluster`	Defines an IKEv2 cluster policy and enters IKEv2 cluster configuration mode.
Step 4	**holdtime** *milliseconds* **Example:** `Device(config-ikev2-cluster)# holdtime 10000`	(Optional) Specifies the time, in milliseconds, to receive messages from a peer. • If no messages are received within the configured time, the peer is declared "dead."
Step 5	**master** {**overload-limit** *percent* | **weight** {**crypto-load** *weight-number* | **system-load** *weight-number*}} **Example:** `Device(config-ikev2-cluster)# master weight crypto-load 10`	Specifies settings for the primary in the HSRP cluster. • **overload-limit** *percent*—The threshold load of the cluster. The load limit to decide when a device is busy and to ignore it when redirecting it for requests. • **weight**—Specifies the weight of a load attribute. Range: 0 to 100. Default: 100. • **crypto-load** *weight-number*—The IKE and IPsec security association (SA) load. • **system-load** *weight-number*—The system and memory load.
Step 6	**port** *port-number* **Example:** `Device(config-ikev2-cluster)# port 2000`	(Optional) Specifies the cluster primary listen port.
Step 7	**slave** {**hello** *milliseconds* | **max-session** *number* | **priority** *number* | **update** *milliseconds*} **Example:** `Device(config-ikev2-cluster)# slave max-session 90`	Specifies settings for subordinate gateways in the HSRP group. • **hello** *milliseconds*—The hello interval, in milliseconds, for a subordinate gateway. • **max-session** *number*—The maximum number of SAs allowed on a subordinate. This keyword is mandatory and cannot be skipped. • **priority** *number*—The subordinate priority.

	Command or Action	Purpose
		• **update** *milliseconds*—The interval, in milliseconds, between two update messages for a subordinate gateway.
Step 8	**standby-group** *group-name* **Example:** `Device(config-ikev2-cluster)# standby-group group1`	Defines the HSRP group containing the subordinates. • *group-name*—The group name is derived from the *group-name* argument specified in the **standby name** command.
Step 9	**shutdown** **Example:** `Device(config-ikev2-cluster)# shutdown`	(Optional) Disables the IKEv2 cluster policy.
Step 10	**exit** **Example:** `Device(config-ikev2-cluster)# exit`	Exits IKEv2 cluster configuration mode and returns to global configuration mode.
Step 11	**crypto ikev2 reconnect key** *key index* **active** *name* **Example:** `Device(config)# crypto ikev2 reconnect key 1 active test123`	Enables the IKEv2 opaque data support for session reconnect. **Note** The ikev2 cluster reconnect feature is enabled for encryption only when the **active** keyword is present in the **ikev2 reconnect key active** *name key-string*. The **active** keyword is mandatory to enable the cluster reconnect feature. If you use the **ikev2 reconnect key** *key-name key-string* command without the **active** keyword in the command, the headend will only be able to decrypt.
Step 12	**end** **Example:** `Device(config-ikev2-cluster)# end`	Exits IKEv2 cluster configuration mode and returns to privileged EXEC mode.

Activating the IKEv2 Redirect Mechanism on the Server

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 redirect gateway init**
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 redirect gateway init** **Example:** `Device(config)# crypto ikev2 redirect gateway init`	Enables the IKEv2 redirect mechanism on the gateway during SA initiation.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Activating the IKEv2 Redirect Mechanism on the Client

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 redirect client** [**max-redirects** *number*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 redirect client** [**max-redirects** *number*] **Example:** `Device(config)# crypto ikev2 redirect client max-redirects 15`	Enables the IKEv2 redirect mechanism on the FlexVPN client. • **max-redirects** *number*—(Optional) Specifies the maximum number of redirects that can be configured on the FlexVPN client for redirect loop detection.

	Command or Action	Purpose
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for IKEv2 Load Balancer

Example: Configuring an HSRP Group for Load Balancing

The following example shows RouterA configured as the active router for an Hot Standby Router Protocol (HSRP) group with a priority of 110. The default priority level is 100. This HSRP group is assigned the group name of group1. The group name is referred in the cluster policy.

```
Device(config)# hostname RouterA
Device(config)# interface GigabitEthernet 0/0/0
Device(config-if)# ip address 10.0.0.1 255.255.255.0
Device(config-if)# standby 1 priority 110
Device(config-if)# standby group1
Device(config-if)# end
```

Example: Configuring the Load Management Mechanism

The following example shows how to configure the load management mechanism in IKEv2:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 cluster
Device(config-ikev2-cluster)# holdtime 10000
Device(config-ikev2-cluster)# master crypto-load 10
Device(config-ikev2-cluster)# port 2000
Device(config-ikev2-cluster)# slave priority 90
Device(config-ikev2-cluster)# standby-group group1
Device(config-ikev2-cluster)# shutdown
Device(config-ikev2-cluster)# end
```

Example: Configuring the Redirect Mechanism

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

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

Example: Configuring the Cluster Reconnect Key

The following example shows how to enable the reconnect key on a server:

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 reconnect key 1 active key
Device(config)# crypto ikev2 reconnect key 2 test
Device(config)# end
```

Additional References

Related Documents

Related Topic	Document Title
Cisco IOS commands	Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
HSRP configuration	*Configuring HSRP*
HSRP commands	Cisco IOS First Hop Redundancy Protocols Command Reference

Standards and RFCs

Standard/RFC	Title
RFC 5685	*Redirect Mechanism for the Internet Key Exchange Protocol Version 2 (IKEv2)*

Technical Assistance

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

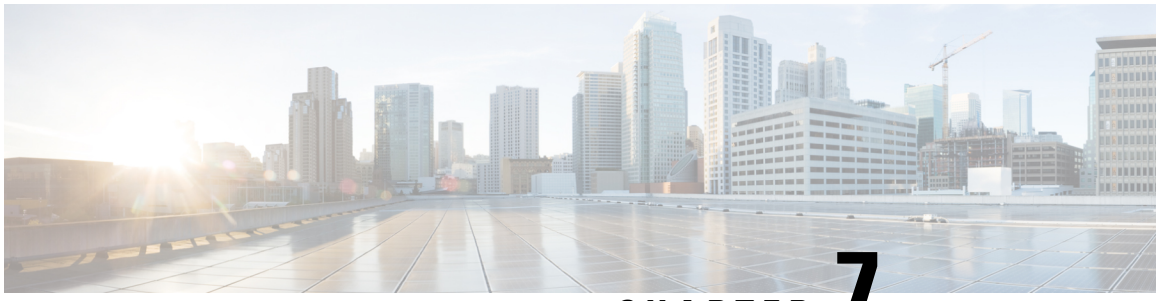
Feature Information for IKEv2 Load Balancer

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

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

Table 8: Feature Information for IKEv2 Load Balancer

Feature Name	Releases	Feature Information
IKEv2 fast convergence with cluster reconnect for Anyconnect		The IKEv2 fast convergence with cluster reconnect for Anyconnect feature enables the Cisco AnyConnect client to reconnect to any server in the cluster. The following command was introduced or modified: **crypto ikev2 reconnect key**
IKEv2 Load Balancer Support		The IKEv2 Load Balancer Support feature distributes incoming IKEv2 requests from FlexVPN clients among IKEv2 FlexVPN servers or gateways by redirecting requests to the least loaded gateway. The following commands were introduced or modified: **crypto ikev2 cluster, crypto ikev2 redirect, holdtime, primary (IKEv2), port (IKEv2), redirect gateway, subordinate (IKEv2), standby-group, show crypto ikev2 cluster, show crypto ikev2 sa.**



CHAPTER 7

Configuring IKEv2 Fragmentation

The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Information About Configuring IKEv2 Fragmentation

IKEv2 Fragmentation

The Internet Key Exchange Version 2 (IKEv2) fragmentation protocol splits large IKEv2 message into a set of smaller ones, called IKE Fragment Messages. The IKEv2 fragmentation methodology, implemented on Cisco IOS software through the IKEv2 Remote Access Headend feature, is a Cisco proprietary method, which restricts interoperability with non-Cisco peers. The fragmentation is performed only on an encrypted IKEv2 packet, and hence, a peer cannot decrypt or authenticate the message until the peer receives all fragments. The IKE Fragmentation adhering to RFC feature implements the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document by encrypting packets after fragmentation, enabling interoperability with non-Cisco peers while continuing to support the Cisco proprietary fragmentation method.

Note By default, IKEv2 fragmentation is disabled, though show run all shows crypto ikev2 fragmentation mtu is 576 B.

Negotiation Between Peers

Effective with the IKE Fragmentation adhering to RFC feature, the support for the IETF standard fragmentation method is added the IKE_SA_INIT message as a notify payload, while Cisco proprietary Fragmentation method continues to be indicated using the Vendor ID payload in the same IKE_SA_INIT message. When fragmentation is enabled, support for both methodologies is displayed as appropriate in the **show crypto ikev2 sa detail** command. The maximum transmission unit (MTU) is configured locally and is not negotiated or exchanged along with the messages. After the INIT exchange, the peers in a network configured with either methodology are aware of the authentication method that must be used and whether the AUTH message can be fragmented.

The following is a sample output from device when debug is enabled showing capability negotiation in INIT request message.

```
*Oct 14 08:45:24.732: IKEv2:(SESSION ID = 0,SA ID = 1):Next payload: SA, version: 2.0
Exchange type: IKE_SA_INIT, flags: INITIATOR Message id: 0, length: 524
Payload contents:
SA Next payload: KE, reserved: 0x0, length: 144
…
Security protocol id: IKE, spi size: 0, type: NAT_DETECTION_DESTINATION_IP
NOTIFY(IKEV2_FRAGMENTATION_SUPPORTED) Next payload: VID, reserved: 0x0, length: 8
Security protocol id: Unknown - 0, spi size: 0, type: IKEV2_FRAGMENTATION_SUPPORTED
VID Next payload: NONE, reserved: 0x0, length: 20
```

In the above output, the INIT request contains the intiator's message to a responder indicating support for both IETF standard fragmentation method and Cisco proprietary fragmentation method through the IKEV2_FRAGMENTATION_SUPPORTED and VID values in the message.

The following is a sample output from device when debug is enabled showing capability negotiation in INIT response message.

```
*Oct 14 08:45:24.732: IKEv2:(SESSION ID = 0,SA ID = 1):Next payload: SA, version: 2.0
Exchange type: IKE_SA_INIT, flags: INITIATOR Message id: 0, length: 524
Payload contents:
SA Next payload: KE, reserved: 0x0, length: 144
last proposal: 0x0, reserved: 0x0, length: 140
…
NOTIFY(IKEV2_FRAGMENTATION_SUPPORTED) Next payload: VID, reserved: 0x0, length: 8
Security protocol id: Unknown - 0, spi size: 0, type: IKEV2_FRAGMENTATION_SUPPORTED <--------
 Response, supporting both
VID Next payload: NONE, reserved: 0x0, length: 20 <-------- Response, supporting both
```

In the above output, the response request contains the responder's message to the initiator indicating support for both IETF standard fragmentation method and Cisco proprietary fragmentation method through the IKEV2_FRAGMENTATION_SUPPORTED and VID values in the message.

Fragmentation Support for Older Releases

To ensure fragmentation support for older releases having Cisco proprietary fragmentation method, IKEv2 continues to use the Vendor ID along with the IKEv2 notification payload type for the IETF standard

fragmentation method. If both fragmentation methods are supported, IKEv2 prefers the IETF standard fragmentation method.

The following table indicates how the fragmentation type is determined based on the capability of peers. CISCO refers to Cisco proprietary fragmentation method and STD refers to the IETF standard fragmentation method.

Peer 1 Capability	Peer 2 Capability	Active Fragmentation Type on the Security Association
STD + CISCO	STD + CISCO	STD
STD	STD	STD
CISCO	CISCO	CISCO
CISCO	STD + CISCO	CISCO
STD	STD + CISCO	STD
STD	CISCO	None
None	None or STD + CISCO or STD or CISCO	None

Encryption, Decryption, and Retransmission of Fragments

Fragmentation and Encryption

A packet is fragmented either based on the maximum transmission unit (MTU) value specified in the **crypto ikev2 fragmentation** command or the default MTU value. IKE messages that only contain the encrypted payload are fragmented. A new payload type—Encrypted and Authenticated Fragment—in the announcement message indicates the fragment number out of the total fragments. This payload is annotated as SKF and the value is 53.

Before the outgoing packet is encrypted, the packet length is checked. The security association established is also verified if the SA is enabled with the IETF standard fragmentation method. The following is a sample output from device displaying the transmission of fragmented packets.

```
*Oct 16 10:31:22.221: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 1 OF Total Fragments: 3
*Oct 16 10:31:22.222: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 2 OF Total Fragments: 3
*Oct 16 10:31:22.222: IKEv2:(SESSION ID = 0,SA ID = 3):Next payload: SKF, version: 2.0
Exchange type: INFORMATIONAL, flags: INITIATOR Message id: 1, length: 244
Payload contents:
SKF Next payload: COOP, reserved: 0x90, length: 216
SKF Fragment number: 3 OF Total Fragments: 3
```

The line “SKF Next payload: COOP, reserved: 0x90, length: 216” and “SKF Fragment number: 1 OF Total Fragments: 3” indicate that the message is a Cooperative key server announcement (ANN) packet fragmented into three fragments.

Decryption and Defragmentation

When incoming fragments are received on a responder, each fragment is decrypted and stored temporarily. During defragmentation (assembling the fragments to the original pack), duplicate fragments, fragment numbers outside of total fragment number, and fragments having an entirely different fragment number are dropped. The fragments are added in ascending order of fragment number and not according to the received order), that way, packet assembly is faster. However, out of order fragments are allowed and processed. Each fragment is verified to ensure that all fragments that pertain to a message are received. If all fragments are received, the packet is assembled from the fragments and processed as a newly received message. Acknowledgment (ACK) message is sent when the original packet is assembled, and not for each fragment.

Retransmissions

IKEv2 retransmissions happen as prompted by IKEv2 retransmission timers. The fragments once constructed and sent out for the first time, are held in a list, ready to be resent when the retransmission timers are triggered. When a retransmitted request is received, IKEv2 resends the response. The response is resent when the first fragment (#1) retransmission is received. The remaining fragment numbers are ignored, thereby allowing faster processing of the response.

Enabling Fragmentation

Use the **crypto ikev2 fragmentation** command to globally enable fragmentation per security association (SA). Fragmentation is enabled on SA when both peers indicate support for fragmentation after INIT exchange on each peers, to be used for IKE_AUTH exchange.

Note This command was introduced through IKEv2 Remote Access Headend feature and has not changed.

You can specify the maximum transmission unit (MTU), in bytes, using the **mtu** *mtu-size* keyword-argument pair. The MTU size refers to the IP or UDP encapsulated IKEv2 packets. The MTU range is from 68 to 1500 bytes. The default MTU size is 576 for IPv4 packets and 1280 bytes for IPv6 packets.

Effective with the IKE Fragmentation adhering to RFC feature, the **crypto ikev2 fragmentation** command:

- Affects future SAs only and does not affect the existing, old SAs.
- Supports Cisco proprietary fragmentation method and the IETF standard fragmentation method.

The **show crypto ikev2 sa detail** command displays the following information:

- The fragmentation method enabled on the peer. If the enabled fragmentation method is IETF standard fragmentation, the output displays the MTU, which is in use.
- Whether fragmentation is enabled on both peers or enabled on the local peer only.

IPv6 Support

The IKE Fragmentation adhering to RFC feature adds support for fragmenting IPv6 packets in IPv6 IKE endpoints when the IETF standard fragmentation method is used. The default MTU value is 1280 bytes and is used when the MTU is not specified in the **crypto ikev2 fragmentation** command. The MTU used in fragmentation is displayed in the output of the **show crypto ikev2 sa detail** command.

How to Configure Configuring IKEv2 Fragmentation

Configuring IKEv2 Fragmentation

Perform this task to enable automatic fragmentation of large IKEv2 packets.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 fragmentation** [**mtu** *mtu-size*]
4. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 fragmentation** [**mtu** *mtu-size*] **Example:** `Device(config)# crypto ikev2 fragmentation mtu 100`	Configures IKEv2 fragmentation. • The MTU range is from 96 to 1500 bytes. The default MTU size is 576 for IPv4 packets and 1280 bytes for IPv6 packets. **Note** The MTU size refers to the IP or UDP encapsulated IKEv2 packets.
Step 4	**end** **Example:** `Device(config)# end`	Exits global configuration mode and returns to privileged EXEC mode.

Configuration Examples for Configuring IKEv2 Fragmentation

Example: IETF Fragmentation Enabled Displaying Configured MTU

The following is a sample output stating IETF standard fragmentation method is enabled. This statement is displayed when the responder supports IETF standard fragmentation method also. The output also displays the MTU in use.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none IN-NEG
Encr: Unknown - 0, PRF: Unknown - 0, Hash: None, DH Grp:0, Auth sign: Unknown - 0, Auth
verify: Unknown - 0
Life/Active Time: 86400/0 sec
CE id: 0, Session-id: 0
Status Description: Initiator waiting for INIT response
Local spi: 2CD1BEADB7C20854 Remote spi: 0000000000000000
Local id: 10.0.8.3
Remote id:
Local req msg id: 0 Remote req msg id: 0
Local next msg id: 1 Remote next msg id: 0
Local req queued: 0 Remote req queued: 0
Local window: 1 Remote window: 1
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 272 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

IPv6 Crypto IKEv2 SA
```

Example: IETF Standard Fragmentation Method Configured on the Initiator

The following is a sample output displaying IETF standard fragmentation method configured on the initiator, and the responder supports Cisco proprietary fragmentation method.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/59 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 84350219051DB9E3 Remote spi: 52A8BB3898E8B5CF
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 4 Remote req msg id: 0
```

```
Local next msg id: 4 Remote next msg id: 0
Local req queued: 4 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes

IPv6 Crypto IKEv2 SA
```

The following is a sample output displaying the responder's configuration. Note that the output displays Cisco proprietary fragmentation method as configured, not enabled.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.9.4/848 10.0.8.3/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/52 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 52A8BB3898E8B5CF Remote spi: 84350219051DB9E3
Local id: 10.0.9.4
Remote id: 10.0.8.3
Local req msg id: 0 Remote req msg id: 4
Local next msg id: 0 Remote next msg id: 4
Local req queued: 0 Remote req queued: 4
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No

IPv6 Crypto IKEv2 SA
```

The following example displays that the initiator supports IETF standard fragmentation method, whereas the responder does not support fragmentation. Note that the output states IETF standard fragmentation method is configured and not enabled.

```
Device# show crypto ikev2 sa detail
IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.8.3/848 10.0.9.4/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/44 sec
CE id: 1004, Session-id: 2
Status Description: Negotiation done
Local spi: 03534703287D9CA1 Remote spi: 146E1CFA68008A92
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 4 Remote req msg id: 0
Local next msg id: 4 Remote next msg id: 0
Local req queued: 4 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
```

```
IETF Std Fragmentation configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes
```

The following is a sample output displaying the responder's configuration. Note the statement "Fragmentation not configured."

```
Device# show crypto ikev2 sa detail
IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
1 10.0.9.4/848 10.0.8.3/848 none/none READY
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/23 sec
CE id: 1004, Session-id: 4
Status Description: Negotiation done
Local spi: 146E1CFA68008A92 Remote spi: 03534703287D9CA1
Local id: 10.0.9.4
Remote id: 10.0.8.3
Local req msg id: 0 Remote req msg id: 3
Local next msg id: 0 Remote next msg id: 3
Local req queued: 0 Remote req queued: 3
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No
```

Example: IETF Standard Fragmentation Method not Configured on the Initiator

The following is a sample output displaying no fragmentation method configured on the initiator.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

Tunnel-id Local Remote fvrf/ivrf Status
2 10.0.8.3/848 10.0.9.4/848 none/none DELETE
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/28 sec
CE id: 1001, Session-id: 1
Status Description: Deleting IKE SA
Local spi: 1A375C00C1D157CF Remote spi: DB50F1BC58814FFA
Local id: 10.0.8.3
Remote id: 10.0.9.4
Local req msg id: 2 Remote req msg id: 4
Local next msg id: 4 Remote next msg id: 5
Local req queued: 2 Remote req queued: 4
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
Fragmentation not configured.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No
```

```
IPv6 Crypto IKEv2 SA
```

Example: IPv6 Support for Fragmentation

This following example shows fragmentation on FlexVPN endpoints—hub and spoke. The following configuration pertains to the hub, which is configured with a maximum transmission unit (MTU) of 1300 for fragmenting the packets.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

IPv6 Crypto IKEv2 SA

Tunnel-id fvrf/ivrf Status
1 none/none READY
Local 4001::2000:3/500
Remote 4001::2000:1/500
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/64 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
Local spi: 45BA0D30D0EB5FFF Remote spi: 8D7B5A8389CEB8B3
Local id: R2.cisco.com
Remote id: R1.cisco.com
Local req msg id: 3 Remote req msg id: 0
Local next msg id: 3 Remote next msg id: 0
Local req queued: 3 Remote req queued: 0
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 1272 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : Yes
Remote subnets:
10.0.0.251 255.255.255.255
IPv6 Remote subnets:
3001::/112
5001::/64
```

The following configuration pertains to the spoke, which is configured with the default MTU.

```
Device# show crypto ikev2 sa detail

IPv4 Crypto IKEv2 SA

IPv6 Crypto IKEv2 SA

Tunnel-id fvrf/ivrf Status
1 none/none READY
Local 4001::2000:1/500
Remote 4001::2000:3/500
Encr: AES-CBC, keysize: 256, PRF: SHA512, Hash: SHA512, DH Grp:5, Auth sign: PSK, Auth
verify: PSK
Life/Active Time: 86400/58 sec
CE id: 1001, Session-id: 1
Status Description: Negotiation done
```

```
Local spi: 8D7B5A8389CEB8B3 Remote spi: 45BA0D30D0EB5FFF
Local id: R1.cisco.com
Remote id: R2.cisco.com
Local req msg id: 0 Remote req msg id: 3
Local next msg id: 0 Remote next msg id: 3
Local req queued: 0 Remote req queued: 3
Local window: 5 Remote window: 5
DPD configured for 0 seconds, retry 0
IETF Std Fragmentation enabled.
IETF Std Fragmentation MTU in use: 1232 bytes.
Extended Authentication not configured.
NAT-T is not detected
Cisco Trust Security SGT is disabled
Initiator of SA : No
Remote subnets:
10.0.0.3 255.255.255.255
```

Additional References for Configuring IKEv2 Fragmentation

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security Commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Standards and RFCs

Standard/RFC	Title
IKEv2 Fragmentation	*draft-ietf-ipsecme-ikev2-fragmentation-10*

Technical Assistance

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

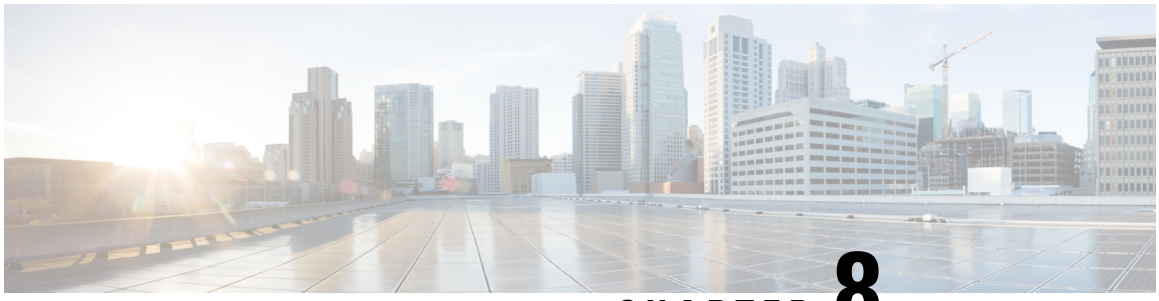
Feature Information for Configuring IKEv2 Fragmentation

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

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

Table 9: Feature Information for Configuring IKEv2 Fragmentation

Feature Name	Releases	Feature Information
IKEv2 Fragmentation adhering to RFC		The IKE Fragmentation adhering to RFC feature implements fragmentation of Internet Key Exchange Version 2 (IKEv2) packets as proposed in the IETF **draft-ietf-ipsecme-ikev2-fragmentation-10** document. The following command was modified: **show crypto ikev2 sa**.



CHAPTER 8

Configuring IKEv2 Reconnect

The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Configuring IKEv2 Reconnect

- You must enable the BypassDownloader function in the AnyConnectLocalPolicy file by setting the <BypassDownloader> value to true. If your device does not support SSL, the BypassDownloader function will not work.You must disable the function manually by setting the <BypassDownloader> value to false, else the connection will fail.

Restrictions for Configuring IKEv2 Reconnect

- The preshared key authorization method cannot be configured on the Internet Key Exchange Version 2 (IKEv2) profile. This is because the IOS IKEv2 support for AutoReconnect feature of AnyConnect

feature uses the preshared key authorization method and configuring the preshared key on the same IKEv2 profile may lead to confusion.

- The following commands cannot be configured on the IKEv2 profile: **authentication local pre-share**, **authentication remote pre-share**, **keyring**, **aaa authorization group psk**, and **aaa authorization user psk**.

Information About Configured IKEv2 Reconnect

IKEv2 and Cisco AnyConnect Client Reconnect Feature

The Auto Reconnect feature in the Cisco AnyConnect client helps the Cisco AnyConnect VPN client to remember the session for a period of time and to resume the connection after establishing the secure channel. As the Cisco AnyConnect Client is extensively used with Internet Key Exchange Version 2 (IKEv2), IKEv2 extends the Auto Reconnect feature support on Cisco IOS software through the IOS IKEv2 support for Auto Reconnect feature of AnyConnect feature.

Auto Reconnect in the Cisco AnyConnect client occurs in the following scenarios:

- The intermediate network is down. The Cisco AnyConnect client tries to resume the session when it is up.
- The Cisco AnyConnect client device switches between networks. This results in source IP or port change, which brings down the existing security association (SA) and, hence, the Cisco AnyConnect client tries to resume the SA using the Auto Reconnect feature.
- The Cisco AnyConnect client device tries to resume SA after returning from sleep or hibernate mode.

Advantages of Using the Auto Reconnect Feature

- The copy attributes used in the original session are reused without querying the authentication, authorization, and accounting (AAA) server.
- The Cisco IOS gateway does not have to contact the RADIUS server for reconnecting to the client.
- No user interaction for authentication or authorization is needed during resuming the session.
- The authentication method is the preshared key when reconnecting a session. This authentication method is quick compared to other authentication methods (that include Rivest, Shamir, and Adelman (RSA) signature authentication method, Elliptic Curve Digital Signature Algorithm (ECDSA) signature (ECDSA-sig) authentication method, and Extensible Authentication Protocol (EAP) authentication method). The preshared key authentication method helps in resuming a session on the IOS software with minimal resources.
- The unused security associations (SAs) are removed thereby freeing the crypto resources.

Auto Reconnect and DPD

Dead Peer Detection (DPD) is configured to confirm the availability of a peer send by sending queries to a peer. If there are no responses from the peer, the security association created for that peer is deleted. You need not configure DPD in a reconnect profile if DPD configured on the FlexVPN server because in both configuration scenarios, the purpose is the same . However, if the feature is enabled, DPD is queued as on demand DPD in IKEv2, which also stores the platform specific handle when deleting the SA.

Message Exchanges Between Cisco IOS Gateway and Cisco AnyConnect Client

The Cisco AnyConnect client contacts the Cisco IOS gateway to establish a security association (SA). During authorization or AUTH exchange (CFGMODE_REQ payload of IKE_AUTH request), IKEv2 checks if the IOS IKEv2 support for the Auto Reconnect feature of AnyConnect feature is enabled in the IKEv2 profile using the **reconnect** command, selects the IKEv2 policy of the chosen IKEv2 profile, and sends the session ID and the session token attributes to the Cisco AnyConnect client in CFGMODE_REPLY payload of the IKE_AUTH response. The authorization method is the preshared key between the client and Cisco IOS software for the SA.

IKEv2 periodically sends dead peer detection (DPD) messages to the Cisco AnyConnect client to validate if the client is active. The Cisco AnyConnect client responds to the DPD messages, which the Cisco IOS gateway understands as an active client and creates a security association (SA) with the client. However, if the client does not reconnect within 30 minutes, which is the default reconnect timeout period, the Cisco IOS gateway assumes that the client is inactive and deletes the SA for that client. The Cisco AnyConnect client needs to start a fresh connection.

Use the **show crypto ikev2 stats reconnect** command to view the connection statistics and the **clear crypto ikev2 session** command to delete the SA with the client.

How to Configure IKEv2 Reconnect

Enabling IKEv2 Reconnect

Perform this task to enable the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **reconnect** [**timeout** *seconds*]
5. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.

	Command or Action	Purpose
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile and enters IKEv2 profile configuration mode.
Step 4	**reconnect** [**timeout** *seconds*] **Example:** `Device(config-ikev2-profile)# reconnect timeout 900`	Enables the IKEv2 support for the Auto Reconnect feature.
Step 5	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.

Troubleshooting IKEv2 Reconnect Configuration

Use the following commands to verify or clear the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature configuration.

SUMMARY STEPS

1. **enable**
2. **show crypto ikev2 stats reconnect**

DETAILED STEPS

Step 1 **enable**

Enables privileged EXEC mode.

- Enter your password if prompted.

Example:

```
Device> enable
```

Step 2 **show crypto ikev2 stats reconnect**

Displays the reconnect statistics.

Example:

```
Device# show crypto ikev2 stats reconnect

Total incoming reconnect connection:        10
Success reconnect connection:               10
Failed reconnect connection:                0
Reconnect capable active session count:     4
Reconnect capable inactive session count:   6
```

Configuration Examples for Configuring IKEv2 Reconnect

Example: Enabling IKEv2 Reconnect

The following example shows how to enable the IOS IKEv2 support for AutoReconnect feature of AnyConnect feature.

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 profile profile1
Device(config-ikev2-profile)# reconnect timeout 600
Device(config-ikev2-profile)# end
```

Additional References for Configuring IKEv2 Reconnect

Related Documents

Related Topic	Document Title
Cisco IOS commands	*Cisco IOS Master Command List, All Releases*
Security commands	• *Cisco IOS Security Command Reference Commands A to C* • *Cisco IOS Security Command Reference Commands D to L* • *Cisco IOS Security Command Reference Commands M to R* • *Cisco IOS Security Command Reference Commands S to Z*
Cisco AnyConnect VPN Client Information	*Cisco AnyConnect VPN Client Administrator Guide, Release 2.4*

Technical Assistance

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

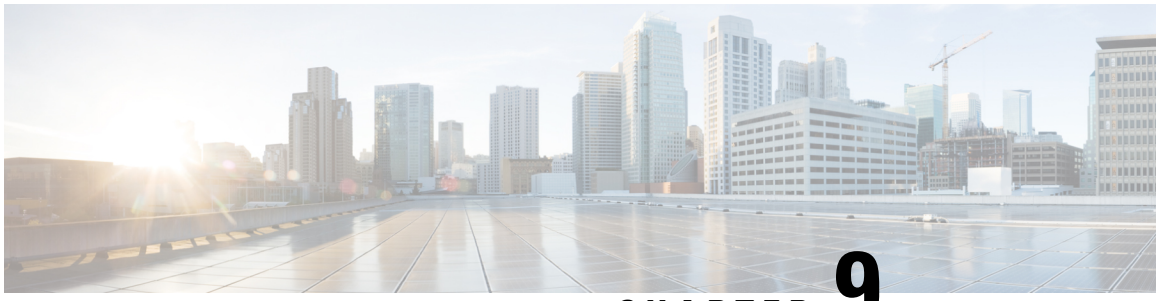
Feature Information for Configuring IKEv2 Reconnect

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

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

Table 10: Feature Information for Configuring IKEv2 Reconnect

Feature Name	Releases	Feature Information
IOS IKEv2 support for AutoReconnect feature of AnyConnect		The IOS IKEv2 support for AutoReconnect feature of AnyConnect feature helps in reestablishing IKEv2 negotiation without user interaction with the Cisco AnyConnect client. The following commands were introduced or modified: **clear crypto ikev2 stats, reconnect, show crypto ikev2 stats.**



CHAPTER 9

Configuring IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Information About IKEv2 Packet of Disconnect

Disconnect Request

The Packet of Disconnect (POD) is a RADIUS disconnect_request packet and is intended to be used in situations where the authenticating agent server wants to disconnect a crypto session.

When the POD is Needed

The Packet of Disconnect is required in the following situations:

- Enforce reauthentication—As a network administrator, you might want to terminate a user on FlexVPN server to forcefully reauthenticate if a session is connected for a very long duration.

- Apply a new policy—As a network administrator, you may want to terminate an active crypto session and apply the new policy on the session when the client reconnects.
- Free resources—A session may need to be terminated to free resources and exit rekey.

IKEv2 Packet of Disconnect

The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature uses the RADIUS Packet of Disconnect (POD) feature to delete a crypto session. The crypto session is deleted to update VPN users to the new user or group policy on the AAA server.

1. AAA passes the attribute key-value pair list, provided by the RADIUS server, to IKEv2.
2. IKEv2 parses the list and locates the Audit-Session-ID, a Cisco AV pair, as a key and validates the pair value.
3. IKEv2 searches the session and deletes the specific session.
4. IKEv2 notifies AAA and AAA notifies the RADIUS server.
5. The session pertaining to the Audit-Session-ID is deleted.

Parameters in IKEv2 Packet of Disconnect

RFC 3576 specifies the following POD codes that are supported for IKEv2 Packet of Disconnect:

- 40 - Disconnect-Request
- 41 - Disconnect-ACK
- 42 - Disconnect-NAK

The Disconnect-ACK code indicates that a session existed for an audit-session-ID and that the session, pertaining to an audit-session-ID was terminated successfully. The Disconnect-NACK code indicates that there are no session corresponding to the audit-session-ID. No reply message is sent to the gateway.

How to Configure IKEv2 Packet of Disconnect

Configuring AAA on the FlexVPN Server

There is no IKEv2-specific configuration required on the FlexVPN server for the IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature. You only need to configure authentication, authorization, and accounting (AAA) on the FlexVPN server. For additional information on AAA configuration, see .

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**

5. **client** {*hostname* | *ip-address*} [**server-key** *string* | **vrf** *vrf-id*]
6. **port** *number*
7. **end**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables AAA globally.
Step 4	**aaa server radius dynamic-author** **Example:**	Sets up the local AAA server for the dynamic authorization service and enters dynamic authorization local server configuration mode. • In this mode, the RADIUS application commands are configured.
Step 5	**client** {*hostname* | *ip-address*} [**server-key** *string* | **vrf** *vrf-id*] **Example:** `Device(config-locsvr-da-radius)# client 192.168.0.5 server-key cisco`	Configures the IP address or hostname of the AAA server client. • Use the **server-key** keyword and *string* argument to configure the server key at the client level. **Note** Configuring the server key at the client level overrides the server key configured at the global level.
Step 6	**port** *number* **Example:** `Device(config-locsvr-da-radius)# port 1812`	Configures the UDP port.
Step 7	**end** **Example:** `Device(config-locsvr-da-radius)# end`	Exits dynamic authorization local server configuration mode and returns to privileged EXEC mode.

Configuration Examples for IKEv2 Packet of Disconnect

Example: Terminating an IKEv2 Session

The following is a sample output from the **show aaa sessions** command. This command must be executed to identify the IKEv2 session that needs to be terminated.

```
Device# show aaa sessions

Total sessions since last reload: 32
Session Id: 3
   Unique Id: 14
   User Name: *not available*
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
Session Id: 30
   Unique Id: 41
   User Name: pskuser2.g1.engdt.com
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
Session Id: 32
   Unique Id: 43
   User Name: pskuser4.g2.engdt.com
   IP Address: 0.0.0.0
   Idle Time: 0
   CT Call Handle: 0
```

In the above output, ID 41 and 43 pertain to IKEv2 sessions. Optionally, you can run the **show aaa user** command to view detailed information about the session.

```
Device# show aaa user 41

Unique id 41 is currently in use.
  No data for type 0
  No data for type EXEC
  No data for type CONN
  NET: Username=(n/a)
    Session Id=0000001E Unique Id=00000029
    Start Sent=0 Stop Only=N
    stop_has_been_sent=N
    Method List=0
    Attribute list:
      7FBD9783CCF0 0 00000001 session-id(408) 4 30(1E)
      7FBD9783CD30 0 00000001 start_time(418) 4 Nov 04 2014 00:20:23
--------
  No data for type CMD
  No data for type SYSTEM
  No data for type VRRS
  No data for type RM CALL
  No data for type RM VPDN
  No data for type AUTH PROXY
  No data for type DOT1X
  No data for type CALL
  No data for type VPDN-TUNNEL
  No data for type VPDN-TUNNEL-LINK
  IPSEC-TUNNEL: Username=pskuser2.g1.engdt.com
```

```
    Session Id=0000001E Unique Id=00000029
    Start Sent=1 Stop Only=N
    stop_has_been_sent=N
    Method List=7FBDA6E05A68 : Name = accnt_prof
    Attribute list:
      7FBD9783CCF0 0 00000001 session-id(408) 4 30(1E)
      7FBD9783CD30 0 00000001 start_time(418) 4 Nov 04 2014 00:20:23
      7FBD9783CD70 0 00000082 formatted-clid(37) 13 192.168.202.2
     7FBD9783CDB0 0 0000008A audit-session-id(819) 37 L2L433010101ZO2L4C0A8CA02ZH119404ZP37

      7FBD9783CDF0 0 00000081 isakmp-phase1-id(737) 21 pskuser2.g1.engdt.com
      7FBD9783BF80 0 00000002 isakmp-initator-ip(738) 4 192.168.202.2
--------
  No data for type MCAST
  No data for type RESOURCE
  No data for type SSG
  No data for type IDENTITY
  No data for type ConnectedApps
Accounting:
  log=0x400018041
  Events recorded :
    CALL START
    ATTR REPLACE
    INTERIM START
    INTERIM STOP
    IPSEC TNL UP
  update method(s) :
    NONE
  update interval = 0
  Outstanding Stop Records : 0
  Dynamic attribute list:
    7FBD9783BF80 0 00000001 connect-progress(75) 4 No Progress
    7FBD9783BFC0 0 00000001 pre-session-time(334) 4 0(0)
    7FBD9783C000 0 00000001 elapsed_time(414) 4 341(155)
    7FBD9783C040 0 00000001 bytes_in(146) 4 0(0)
    7FBD9783C080 0 00000001 bytes_out(311) 4 0(0)
    7FBD9783CCF0 0 00000001 pre-bytes-in(330) 4 0(0)
    7FBD9783CD30 0 00000001 pre-bytes-out(331) 4 0(0)
    7FBD9783CD70 0 00000001 paks_in(147) 4 0(0)
    7FBD9783CDB0 0 00000001 paks_out(312) 4 0(0)
    7FBD9783CDF0 0 00000001 pre-paks-in(332) 4 0(0)
    7FBD9783BA20 0 00000001 pre-paks-out(333) 4 0(0)
Debg: No data available
Radi: No data available
Interface:
  TTY Num = -1
  Stop Received = 0
  Byte/Packet Counts till Call Start:
    Start Bytes In = 0             Start Bytes Out = 0
    Start Paks  In = 0             Start Paks  Out = 0
  Byte/Packet Counts till Service Up:
    Pre Bytes In = 0             Pre Bytes Out = 0
    Pre Paks  In = 0             Pre Paks  Out = 0
  Cumulative Byte/Packet Counts :
    Bytes In = 0             Bytes Out = 0
    Paks  In = 0             Paks  Out = 0
  StartTime = 00:20:23 IST Nov 4 2014
  AuthenTime = 00:20:23 IST Nov 4 2014
  Component = VPN IPSEC
Authen: service=NONE type=NONE method=NONE
Kerb: No data available
Meth: No data available
Preauth: No Preauth data.
General:
```

```
  Unique Id = 00000029
  Session Id = 0000001E
  Session Server Key = 1771D693
  Attribute List:
PerU: No data available
Service Profile: No Service Profile data.
Unkn: No data available
Unkn: No data available
```

Note the audit-session-id in the above output, which is L2L433010101ZO2L4C0A8CA02ZH119404ZP37. The following sample output is displayed on the FlexVPN server on starting an accounting session starts with a RADIUS server.

```
Nov  4 00:26:49.908 IST: RADIUS/ENCODE: Best Local IP-Address 192.168.202.1 for Radius-Server
 9.45.15.144
Nov  4 00:26:49.908 IST: RADIUS(0000002C): Send Accounting-Request to 9.45.15.144:1813 id
1646/231, len 288
Nov  4 00:26:49.908 IST: RADIUS:  authenticator 29 63 0C 79 C1 5E F2 0E - F3 CA 36 DD A3
55 C1 DE
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Session-Id     [44]  10  "00000021"
Nov  4 00:26:49.908 IST: RADIUS:  Calling-Station-Id  [31]  15  "192.168.202.2"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  64
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   58
"audit-session-id=L2L433010101ZO2L4C0A8CA02ZH11941194ZN3A"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  46
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   40
"isakmp-phase1-id=pskuser1.g1.engdt.com"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  40
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair       [1]   34
"isakmp-initator-ip=192.168.202.2"
Nov  4 00:26:49.908 IST: RADIUS:  User-Name           [1]   23  "pskuser1.g1.engdt.com"
Nov  4 00:26:49.908 IST: RADIUS:  Vendor, Cisco       [26]  36
Nov  4 00:26:49.908 IST: RADIUS:   Cisco AVpair      [1]   30  "connect-progress=No Progress"
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Authentic      [45]  6   Local
[2]
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Status-Type    [40]  6   Start
[1]
Nov  4 00:26:49.908 IST: RADIUS:  NAS-IP-Address      [4]   6   192.168.202.1
Nov  4 00:26:49.908 IST: RADIUS:  home-hl-prefix      [151] 10  "D33648D8"
Nov  4 00:26:49.908 IST: RADIUS:  Acct-Delay-Time     [41]  6   0
Nov  4 00:26:49.908 IST: RADIUS(0000002C): Sending a IPv4 Radius Packet
```

The following output is displayed on the FlexVPN server when disconnecting a session for a specific audit-session-id. The terminate session request is sent to the RADIUS server via a RADIUS client. In this example, the session for the audit-session-ID, which is L2L433010101ZO2L4C0A8CA02ZH119404ZP37 is terminated and, hence, not visible in the output.

```
Nov  4 00:32:29.004 IST: RADIUS: POD  received from id 216 9.45.15.144:50567, POD Request,
 len 84
Nov  4 00:32:29.004 IST: POD: 9.45.15.144 request queued
Nov  4 00:32:29.004 IST:  ++++++ POD Attribute List ++++++
Nov  4 00:32:29.004 IST: 7FBD9783D3A8 0 00000089 audit-session-id(819) 39
L2L433010101ZO2L4C0A8CA02ZH11941194ZN3B
Nov  4 00:32:29.004 IST:
Nov  4 00:32:29.004 IST: POD: Sending ACK from port 1812 to 9.45.15.144/50567

Nov  4 00:32:29.005 IST: IKEv2:(SESSION ID = 59,SA ID = 2):Check for existing active SA
Nov  4 00:32:29.006 IST: IKEv2:in_octets 0, out_octets 0
Nov  4 00:32:29.006 IST: IKEv2:in_packets 0, out_packets 0
Nov  4 00:32:29.006 IST: IKEv2:(SA ID = 2):[IKEv2 -> AAA] Accounting stop request sent
successfully
Nov  4 00:32:29.006 IST: IKEv2:(SESSION ID = 59,SA ID = 2):Delete all IKE SAs
```

```
Nov  4 00:32:29.010 IST: RADIUS/ENCODE(0000002D):Orig. component type = VPN IPSEC
Nov  4 00:32:29.010 IST: RADIUS(0000002D): Config NAS IP: 0.0.0.0
Nov  4 00:32:29.010 IST: RADIUS(0000002D): Config NAS IPv6: ::
Nov  4 00:32:29.010 IST: RADIUS(0000002D): sending
Nov  4 00:32:29.011 IST: RADIUS/ENCODE: Best Local IP-Address 192.168.202.1 for Radius-Server
 9.45.15.144
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Send Accounting-Request to 9.45.15.144:1813 id
1646/246, len 356
Nov  4 00:32:29.011 IST: RADIUS:  authenticator 52 88 5E CB 8B FA 1E C1 - CC EF 73 75 89
73 CA 95
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Session-Id     [44]  10  "00000022"
Nov  4 00:32:29.011 IST: RADIUS:  Calling-Station-Id  [31]  15  "192.168.202.2"
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  64
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   58
"audit-session-id=L2L433010101ZO2L4C0A8CA02ZH11941194ZN3B"
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  46
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   40
"isakmp-phase1-id=pskuser1.g1.engdt.com"
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  40
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   34
"isakmp-initator-ip=192.168.202.2"
Nov  4 00:32:29.011 IST: RADIUS:  User-Name           [1]   23  "pskuser1.g1.engdt.com"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Authentic      [45]  6   Local
[2]
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  36
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair      [1]   30  "connect-progress=No Progress"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Session-Time   [46]  6   56
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Input-Octets   [42]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Output-Octets  [43]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Input-Packets  [47]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Output-Packets [48]  6   0
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Terminate-Cause[49]  6   none
[0]
Nov  4 00:32:29.011 IST: RADIUS:  Vendor, Cisco       [26]  32
Nov  4 00:32:29.011 IST: RADIUS:   Cisco AVpair       [1]   26  "disc-cause-ext=No Reason"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Status-Type    [40]  6   Stop
[2]
Nov  4 00:32:29.011 IST: RADIUS:  NAS-IP-Address      [4]   6   192.168.202.1
Nov  4 00:32:29.011 IST: RADIUS:  home-hl-prefix      [151] 10  "E2F80C34"
Nov  4 00:32:29.011 IST: RADIUS:  Acct-Delay-Time     [41]  6   0
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Sending a IPv4 Radius Packet
Nov  4 00:32:29.011 IST: RADIUS(0000002D): Started 5 sec timeout
```

The following output is displayed when there is no valid session for the specific audit-session-ID. This happens if there is no session pertaining to the specific audit-session-id when the session is terminated already. Note the NACK message that is sent back to the FlexVPN server

```
Nov  4 00:30:31.905 IST: RADIUS: POD  received from id 131 9.45.15.144:52986, POD Request,
 len 84
Nov  4 00:30:31.905 IST: POD: 9.45.15.144 request queued
Nov  4 00:30:31.905 IST:  ++++++ POD Attribute List ++++++
Nov  4 00:30:31.905 IST: 7FBD9783BA20 0 00000089 audit-session-id(819) 39
L2L433010101ZO2L4C0A8CA02ZH11941194ZN3A
Nov  4 00:30:31.905 IST:
Nov  4 00:30:31.906 IST: POD: 9.45.15.144 Unsupported attribute type 26 for component
Nov  4 00:30:31.906 IST: POD: 9.45.15.144 user  0.0.0.0i sessid 0x0 key 0x0 DROPPED
Nov  4 00:30:31.906 IST: POD: Added Reply Message: No Matching Session
Nov  4 00:30:31.906 IST: POD: Added NACK Error Cause: Invalid Request
Nov  4 00:30:31.906 IST: POD: Sending NAK from port 1812 to 9.45.15.144/52986
Nov  4 00:30:31.906 IST: RADIUS:  18  21  4E6F204D61746368696E672053657373696F6E
Nov  4 00:30:31.906 IST: RADIUS:  101 6   00000194
```

Additional References for IKEv2 Packet of Disconnect

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z
RADIUS Packet of Disconnect	RADIUS Packet of Disconnect RADIUS Packet of Disconnect

Standards and RFCs

Standard/RFC	Title
RFC 3576	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*
RFC 5176	*Dynamic Authorization Extensions to Remote Authentication Dial In User Service (RADIUS)*

Technical Assistance

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

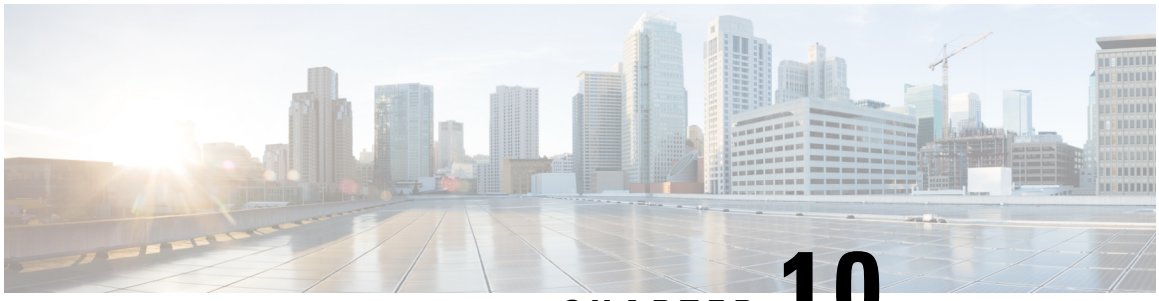
Feature Information for IKEv2 Packet of Disconnect

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

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

Table 11: Feature Information for IKEv2 Packet of Disconnect

Feature Name	Releases	Feature Information
IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect		The IKEv2 Remote Access Change of Authorization (CoA)—Packet of Disconnect feature terminates an active crypto IKEv2 session on Cisco supported devices. No commands were introduced by this feature.



CHAPTER 10

Configuring IKEv2 Change of Authorization Support

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for IKEv2 Change of Authorization Support

- IKEv2 must be registered as a component, via a registry entry, on Cisco AAA component.

Restrictions for IKEv2 Change of Authorization Support

- This feature supports change of authorization (CoA) packets received from RADIUS-based AAA server only.

Information About IKEv2 Change of Authorization Support

RADIUS Change of Authorization

The RADIUS Change of Authorization (CoA) provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is authenticated. When a policy changes for a user or user group in AAA, administrators can send the RADIUS CoA packets from the AAA server such as a Cisco Secure Access Control Server (ACS) to reinitialize authentication and apply the new policy.

A standard RADIUS interface is typically used in a pulled model, in which the request originates from a device attached to a network and the response is sent from the queried servers. Cisco software supports the RADIUS CoA request defined in RFC 5176 that is used in a pushed model, in which the request originates from the external server to the device attached to the network, and enables the dynamic reconfiguring of sessions from external authentication, authorization, and accounting (AAA) or policy servers.

Fore more information on RADIUS CoA, see *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS Release 15M&T* or *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS XE Release 3S*

Working of Change of Authorization on IKEv2

The FlexVPN - IKEv2 CoA for QoS and ACL feature allows to change attributes of an active IKEv2 crypto session to apply a new authorization attributes. The Cisco AAA component receives a Change of Authorization (CoA) packet from a AAA server and checks if the received CoA packet is meant for any of the components registered with it. If a component sees that the CoA packet is meant for itself, it processes it further. Based on the fields in the CoA packet, if the packet is relevant for a given component, such as IKEv2, the packet is consumed by that component. AAA will not forward the packet to the next component in the list.

In case of this feature, after IKEv2 receives a CoA packet, IKEv2 verifies the CoA packet for the Cisco (AV) pairs. IKEv2 identifies the session based on the audit-session-id which is already stored in the RADIUS server.

If the CoA packet contains attributes not supported by IKEv2, IKEv2 discards the packet and sends a CoA-NACK to AAA component.

Supported AV Pairs for IKEv2 Change of Authorization

The FlexVPN - IKEv2 CoA for QoS and ACL feature supports the following Cisco AV pairs:

- ip:interface-config
- ip:sub-policy-In
- ip:sub-policy-Out
- ip:sub-qos-policy-in
- ip:sub-qos-policy-out
- ipsec:inacl
- ipsec:outacl

How to Configure IKEv2 Change of Authorization Support

Configuring Change of Authorization on the FlexVPN Server

There is no IKEv2-specific configuration required for this feature. on the FlexVPN server for the IKEv2 Change of Authorization (CoA) Support feature. You only need to configure the RADIUS Change of Authorization on the FlexVPN server. For more information on AAA configuration, see the "RADIUS Change of Authorization" feature module in the *Authentication, Authorization, and Accounting Configuration Guide, Cisco IOS Release 15M&T*.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **aaa new-model**
4. **aaa server radius dynamic-author**
5. **client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string*
6. **port** *port-number*
7. **auth-type** {**any** | **all** | **session-key**}
8. **ignore session-key**
9. **ignore server-key**
10. **exit**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**aaa new-model** **Example:** `Device(config)# aaa new-model`	Enables authentication, authorization, and accounting (AAA) globally.
Step 4	**aaa server radius dynamic-author** **Example:** `Device(config)# aaa server radius dynamic-author`	Enters dynamic authorization local server configuration mode and specifies a RADIUS client from which a device accepts Change of Authorization (CoA) and disconnect requests. Configures the device as a AAA server to facilitate interaction with an external policy server.

	Command or Action	Purpose
Step 5	**client** {*ip-address* | *name* [**vrf** *vrf-name*]} **server-key** [**0** | **7**] *string* **Example:** `Device(config-locsvr-da-radius)# client 10.0.0.1`	Configures the RADIUS key to be shared between a device and RADIUS clients.
Step 6	**port** *port-number* **Example:** `Device(config-locsvr-da-radius)# port 3799`	Specifies the port on which a device listens for RADIUS requests from configured RADIUS clients. **Note** The default port for packet of disconnect is 1700. Port 3799 is required to interoperate with ACS 5.1.
Step 7	**auth-type** {**any** | **all** | **session-key**} **Example:** `Device(config-locsvr-da-radius)# auth-type all`	Specifies the type of authorization that the device must use for RADIUS clients. The client must match the configured attributes for authorization.
Step 8	**ignore session-key** **Example:** `Device(config-locsvr-da-radius)# ignore session-key`	(Optional) Configures the device to ignore the session key.
Step 9	**ignore server-key** **Example:** `Device(config-locsvr-da-radius)# ignore server-key`	(Optional) Configures the device to ignore the server key.
Step 10	**exit** **Example:** `Device(config-locsvr-da-radius)# exit`	Returns to global configuration mode.

Verifying IKEv2 Change of Authorization Support on Cisco ASR 1000 Series Router

Use the following show commands to view the success of change of authorization (CoA) on Cisco ASR 1000 Series Aggregation Services Routers.

SUMMARY STEPS

1. **enable**
2. **show platform hardware qfp active feature qos all output all**
3. **show platform hardware qfp active feature qos all input all**

DETAILED STEPS

Step 1 **enable**

Example:

```
Device> enable
```

Enables privileged EXEC mode.

- Enter your password if prompted.

Step 2 **show platform hardware qfp active feature qos all output all**

Example:

```
Device# show platform hardware qfp active feature qos all output all

Interface: Virtual-Access1, QFP if_h: 14, Num Targets: 1
  Target: Out, Num UIDBs: 1
    UIDB #: 0
    Hierarchy level: 0, Num matching iftgts: 1
    Policy name: aaa-out-policy, Policy id: 9679472
    Parent Class Idx: 0, Parent Class ID: 0
      IF Tgt#: 0, ifh: 14, member_ifh: 0, link_idx: 0
        PSQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593, Match index: 0
            Class name: class-default, Policy name: aaa-out-policy
              psqd[0-3]: 0x00000000 0x00000000 0x00000001 0x00000000
        ISQD specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              isqd[0-3]:  0x88e78ec0 0x00000000 0x00000000 0x00000000
              (cache) isqd[0-3]: 0x88e78ec0 0x00000000 0x00000000 0x00000000
        Police specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              Policer id: 0x20000002
              hw_policer[0-3]:       0x4000047e 0x00163ac8 0x00000000 0x00000000
              cache hw_policer[0-3]: 0x4000047e 0x00163ac8 0x00000000 0x00000000
              conform stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
              exceed stats  (paks/octets): 0x0000000000000000, : 0x0000000000000000
              violate stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
              police_info:           0x00000000
              cache police_info:     0x00000000
        Queue specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              No queue configured
        Schedule specifics:
          Target Index: 0, Num Classes: 1
            Class index: 0, Class object id: 1593
            Class name: class-default, Policy name: aaa-out-policy
              No schedule info (no queue configured)
```

Displays platform-specific information if CoA was successful.

Step 3 **show platform hardware qfp active feature qos all input all**

Example:

```
Device#  show platform hardware qfp active feature qos all input all

Interface: Virtual-Access1, QFP if_h: 14, Num Targets: 1
  Target: In, Num UIDBs: 1
    UIDB #: 0
```

```
Hierarchy level: 0, Num matching iftgts: 1
Policy name: aaa-in-policy, Policy id: 980784
Parent Class Idx: 0, Parent Class ID: 0
  IF Tgt#: 0, ifh: 14, member_ifh: 0, link_idx: 0
    PSQD specifics:
      Target Index: 0, Num Classes: 1
        Class index: 0, Class object id: 1593, Match index: 0
        Class name: class-default, Policy name: aaa-in-policy
          psqd[0-3]: 0x00000000 0x00000000 0x00000001 0x00000000
    ISQD specifics:
      Target Index: 0, Num Classes: 1
        Class index: 0, Class object id: 1593
        Class name: class-default, Policy name: aaa-in-policy
          isqd[0-3]:  0x88d49748 0x00000001 0x00000000 0x00000000
          (cache) isqd[0-3]: 0x88d49748 0x00000001 0x00000000 0x00000000
    Police specifics:
      Target Index: 0, Num Classes: 1
        Class index: 0, Class object id: 1593
        Class name: class-default, Policy name: aaa-in-policy
          Policer id: 0x20000003
          hw_policer[0-3]:       0x10000140 0x00113a29 0x00000000 0x00000000
          cache hw_policer[0-3]: 0x10000140 0x00113a29 0x00000000 0x00000000
          conform stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
          exceed stats  (paks/octets): 0x0000000000000000, : 0x0000000000000000
          violate stats (paks/octets): 0x0000000000000000, : 0x0000000000000000
          police_info:           0x00000000
          cache police_info:     0x00000000
    Queue specifics:
      Target Index: 0, Num Classes: 1
        Class index: 0, Class object id: 1593
        Class name: class-default, Policy name: aaa-in-policy
          No queue configured
    Schedule specifics:
      Target Index: 0, Num Classes: 1
        Class index: 0, Class object id: 1593
        Class name: class-default, Policy name: aaa-in-policy
          No schedule info (no queue configured)
```

Displays the feature status.

Configuration Examples for IKEv2 Change of Authorization Support

Example: Triggering a Change of Authorization

The following sample output is displayed when an administrator triggers a change of authorization (CoA). The session is identified based on the audit-session-id, a dynamic string, which is an encoded form of 6 tuple information of a session with peer.

IKEv2 receives a change of authorization (CoA) packet from a RADIUS server. The session is identified based on audit-session-id.

```
*Oct  6 23:38:55.250: RADIUS: COA  received from id 125 10.106.210.176:58712, CoA Request,
 len 257
*Oct  6 23:38:55.251: COA: 10.106.210.176 request queued
```

```
*Oct  6 23:38:55.251: RADIUS:  authenticator BD 97 5E BA B2 EB C1 C5 - 1A 14 51 3D C2 C8
66 3F
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  62
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   56
"audit-session-id=L2L44D010102ZO2L44D010101ZI1F401F4ZO2"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  52
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   46
"ip:interface-config=service-policy input pol"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  35
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   29  "ip:sub-qos-policy-out=2M-IN"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  36
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair      [1]   30  "ip:sub-qos-policy-in=aaa-pol"
*Oct  6 23:38:55.251: RADIUS:  Vendor, Cisco       [26]  52
*Oct  6 23:38:55.251: RADIUS:   Cisco AVpair       [1]   46
"ip:interface-config=service-policy output 2M"
*Oct  6 23:38:55.251: COA: Message Authenticator missing or failed decode

*Oct  6 23:38:55.251:  ++++++ CoA Attribute List ++++++
*Oct  6 23:38:55.251: 421C9694 0 00000089 audit-session-id(819) 37
L2L44D010102ZO2L44D010101ZI1F401F4ZO2
*Oct  6 23:38:55.251: 421C9584 0 00000081 interface-config(222) 24 service-policy input pol
*Oct  6 23:38:55.251: 421C95B8 0 00000081 sub-qos-policy-out(423) 5 2M-IN
*Oct  6 23:38:55.251: 421C95EC 0 00000081 sub-qos-policy-in(421) 7 aaa-pol
*Oct  6 23:38:55.251: 421C9620 0 00000081 interface-config(222) 24 service-policy output
2M
*Oct  6 23:38:55.251:
*Oct  6 23:38:55.251: COA: Added NACK Error Cause: Success
```

Additional References for IKEv2 Change of Authorization Support

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Technical Assistance

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

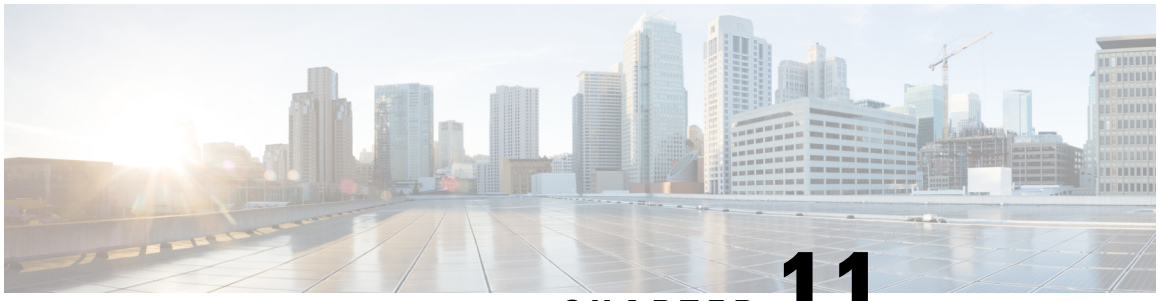
Feature Information for IKEv2 Change of Authorization Support

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

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

Table 12: Feature Information for IKEv2 Change of Authorization Support

Feature Name	Releases	Feature Information
FlexVPN - IKEv2 CoA for QoS and ACL		The FlexVPN - IKEv2 CoA for QoS and ACL feature supports RADIUS Change of Authorization (CoA) on an active IKEv2 crypto session. No commands were modified or updated by this feature.



CHAPTER 11

Configuring Aggregate Authentication

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server.

Finding Feature Information

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

Use Cisco Feature Navigator to find information about platform support and Cisco software image support. To access Cisco Feature Navigator, go to https://cfnng.cisco.com/. An account on Cisco.com is not required.

Prerequisites for Configuring Aggregate Authentication

- You must enable the BypassDownloader function in the AnyConnectLocalPolicy file by setting the <BypassDownloader> value to true. If your device does not support SSL, the BypassDownloader function will not work.You must disable the function manually by setting the <BypassDownloader> value to false, else the connection will fail.

Information for Configuring Aggregate Authentication

Cisco AnyConnect and FlexVPN

To establish a VPN connection, the VPN client must obtain user credentials using authentication methods such as, extensible authentication protocol (EAP), Extended Authentication (XAUTH), etc. and forward the user credentials to a hub, which contacts an access control server. The access control server sends an external database or active directory (AD) to validate the credentials.

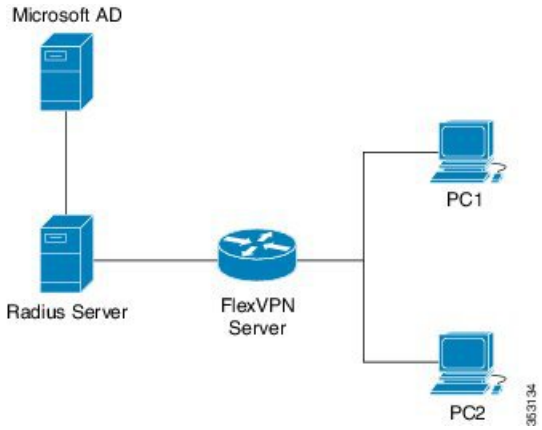
FlexVPN server (as a hub) works with Cisco Secure Access Control Server to validate user credentials to establish VPN connections. However, Cisco AnyConnect uses EAP to obtain user credentials and does not support XAUTH. On the other hand, Cisco Secure Access Control Server does not support EAP-MD5 with external database (in this case AD). This leads to a scenario where either Cisco Secure Access Control Server must support EAP-MD5 or FlexVPN must authenticate the information from Cisco AnyConnect separately and connect separately with Cisco Secure Access Control Server. FlexVPN can use the Aggregate Authentication method to authentication information from Cisco AnyConnect. Implementing aggregate authentication method on FlexVPN server would provide a window to add more feature support on Cisco IOS software.

The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet using Cisco AnyConnect and FlexVPN server. This is a server-specific feature and works with Cisco AnyConnect.

How Aggregate Authentication Works

Internet Key Exchange Version 2 supports Cisco AnyConnect that uses the proprietary AnyConnect EAP authentication method by implementing basic aggregate authentication where authentication is performed via authentication, authorization, and accounting (AAA) using the remote RADIUS server. The following is an example of a network topology explains aggregate authentication implementation on Cisco IOS software.

Figure 5: FlexVPN Server Connected to RADIUS Server



In this diagram:

- Cisco Secure Access Control Server acts as a RADIUS server for authorization.

- The credentials are stored in Microsoft Active Directory, which acts as the active directory for authentication.

Note Microsoft Active Directory is referred for example purpose only. It does not matter where the credentials are stored.

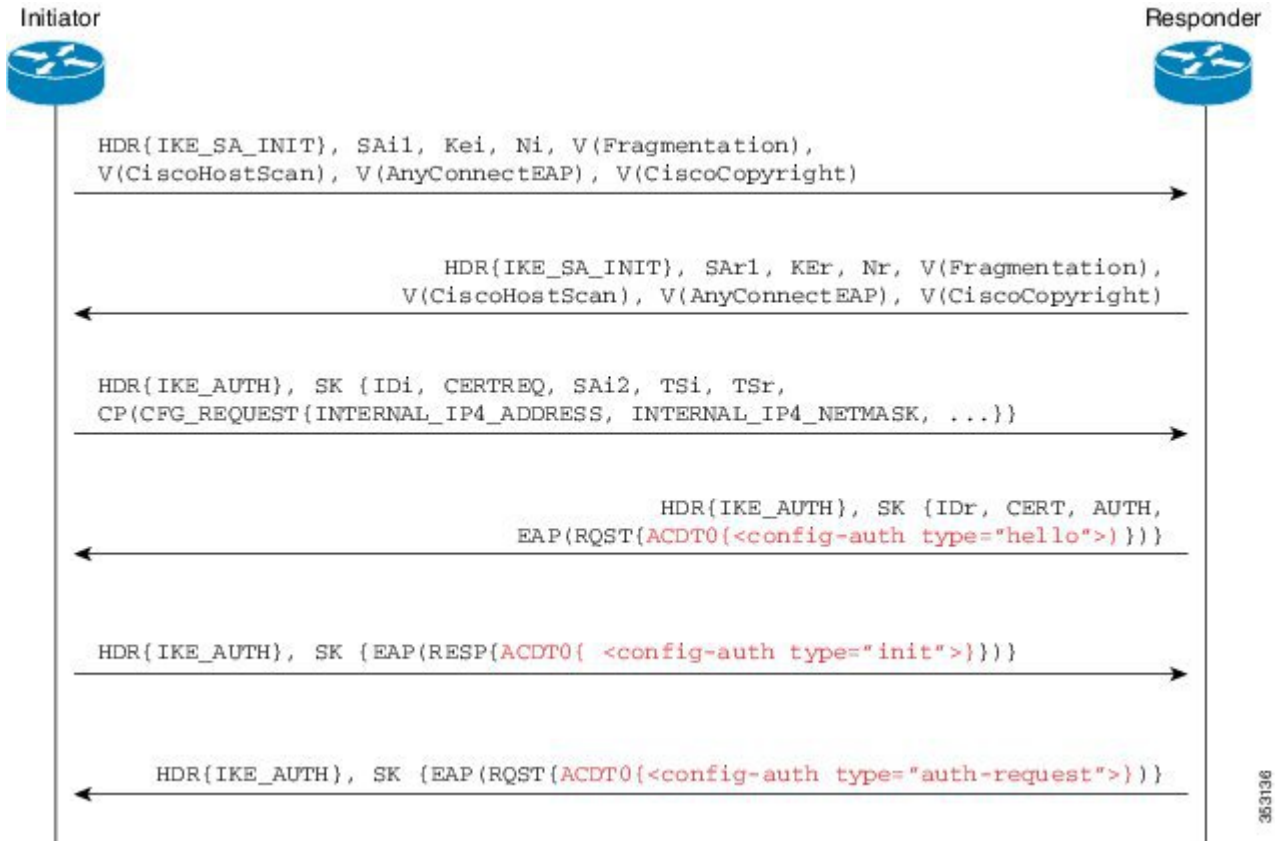
- Cisco device acts as FlexVPN server.
- Windows 7 PC acts as Cisco AnyConnect client.

1. To initiate a VPN connection, Cisco AnyConnect client verifies a FlexVPN server using certifications.
2. After verifying the certificates, Cisco AnyConnect client sends Cisco AnyConnect EAP loaded message to FlexVPN server.
3. When FlexVPN server receives Cisco AnyConnect EAP loaded message from Cisco AnyConnect, FlexVPN server downloads the message and strips the message of EAP.
4. FlexVPN establishes a connection with RADIUS server for authorization and Microsoft Active Directory (AD) for authentication, and forwards the stripped message to verify the credentials provided by Cisco AnyConnect client.
5. When the credentials are verified and approved by RADIUS server and Microsoft Active Directory (AD), an appropriate reply is sent to FlexVPN server, which in turn replies to Cisco AnyConnect and a VPN connection is established.

IKE Exchanges Using Cisco AnyConnect EAP

Authentication in IKE using AnyConnect EAP is a variation of the standards EAP model as described in RFC 3748. When using AnyConnect EAP the public configuration or authentication XML is transported via EAP payloads. The following figure illustrates the typical message flow used by Cisco AnyConnect .

Figure 6: IKE Exchanges using AnyConnect EAP



1. Cisco AnyConnect client initiates IKE connection to FlexVPN server. The client sends vendor ID payloads to indicate support for Cisco AnyConnect EAP in addition to the typical IKE payloads. The client identifies itself as a Cisco product by including the Cisco copyright vendor ID.
2. The server gateway sends vendor ID payloads to indicate fragmentation and AnyConnect EAP support and identifies itself as a Cisco product by including the Cisco copyright vendor ID.
3. The configuration payload requests the tunnel configuration. The client indicates its desire to use Cisco AnyConnect EAP authentication by omitting the AUTH Payload from this message.
4. The Aggregate Authentication and Configuration protocol is carried over EAP
5. FlexVPN server sends a EAP success message.
6. Cisco AnyConnect client sends the AUTH payload.
7. FlexVPN server sends the AUTH payload and the tunnel configuration attributes that Cisco AnyConnect client requested.

Dual-Factor Authentication Support with IKEv2

The aggregate authentication implementation on Cisco IOS software can be extended for dual-factor authentication. Double authentication can be done by introducing new AnyConnect EAP exchange during Aggregate Authentication which exchange and validate the device certificate information. This mechanism of authenticating 'device' as well as 'user' is called 'Double Authentication'.

Note AnyConnect EAP is AnyConnect client specific authentication method and does not apply to any other client.

How to Configure Aggregate Authentication

Configuring the FlexVPN Server for Aggregate Authentication

Perform this task to configure aggregate authentication on the FlexVPN server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 profile** *profile-name*
4. **aaa accounting anyconnect-eap** *list-name*
5. **match identity remote key-id** *opaque-string*
6. **authentication remote anyconnect-eap aggregate** [**cert-request**]
7. **authentication local rsa-sig**
8. **pki trustpoint** *trustpoint-label*
9. **aaa authentication anyconnect-eap** *list-name*
10. **aaa authorization group anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name*
11. **aaa authorization user anyconnect-eap cached**
12. **aaa authorization user anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name*
13. **end**
14. **show crypto ikev2 session detailed**

DETAILED STEPS

	Command or Action	Purpose
Step 1	**enable** **Example:** `Device> enable`	Enables privileged EXEC mode. • Enter your password if prompted.
Step 2	**configure terminal** **Example:** `Device# configure terminal`	Enters global configuration mode.
Step 3	**crypto ikev2 profile** *profile-name* **Example:** `Device(config)# crypto ikev2 profile profile1`	Defines an IKEv2 profile name and enters IKEv2 profile configuration mode.

	Command or Action	Purpose
Step 4	**aaa accounting anyconnect-eap** *list-name* **Example:** `Device(config-ikev2-profile)# aaa accounting anyconnect-eap list1`	Enables authentication, authorization, and accounting (AAA) accounting method lists when the IKEv2 remote authentication method is AnyConnect EAP.
Step 5	**match identity remote key-id** *opaque-string* **Example:** `Device(config-ikev2-profile)# match identity remote key-id aggauth_user3@abc.com`	Matches a profile based on the identity of the type remote key ID.
Step 6	**authentication remote anyconnect-eap aggregate** [**cert-request**] **Example:** `Device(config-ikev2-profile)# authentication remote anyconnect-eap aggregate cert-request`	Specifies aggregate authentication for Cisco AnyConnect EAP. • **cert-request** - requests certificate from Cisco AnyConnect client for double authentication.
Step 7	**authentication local rsa-sig** **Example:** `Device(config-ikev2-profile)# authentication local rsa-sig`	Specifies Rivest, Shamir, and Adelman (RSA) signature as the local authentication method.
Step 8	**pki trustpoint** *trustpoint-label* **Example:** `Device(config-ikev2-profile)# pki trustpoint CA1`	Specifies Public Key Infrastructure (PKI) trustpoints for use with the RSA signature authentication method.
Step 9	**aaa authentication anyconnect-eap** *list-name* **Example:** `Device(config-ikev2-profile)# aaa authentication anyconnect-eap list1`	Specifies authentication, authorization, and accounting (AAA) authentication list for Cisco AnyConnect EAP authentication. • **anyconnect-eap**—Specifies AAA AnyConnect EAP authentication. • *list-name*—The AAA authentication list name.
Step 10	**aaa authorization group anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name* **Example:** `Device(config-ikev2-profile)# aaa authorization group anyconnect-eap list list1 name-mangler mangler1`	Specifies the AAA authorization for each group policy when the remote authentication method is AnyConnect EAP and derives the name mangler.
Step 11	**aaa authorization user anyconnect-eap cached** **Example:** `Device(config-ikev2-profile)# aaa authorization user anyconnect-eap cached`	Specifies the AAA authorization for each user policy when the remote authentication method is AnyConnect EAP and uses cached attributes from the AnyConnect EAP authentication.

	Command or Action	Purpose
Step 12	**aaa authorization user anyconnect-eap list** *aaa-listname* **name-mangler** *mangler-name* **Example:** `Device(config-ikev2-profile)# aaa authorization user anyconnect-eap list list1 name-mangler mangler1`	Specifies the AAA method list for the remote authentication method and derives the name mangler.
Step 13	**end** **Example:** `Device(config-ikev2-profile)# end`	Exits IKEv2 profile configuration mode and returns to privileged EXEC mode.
Step 14	**show crypto ikev2 session detailed** **Example:** `Device# show crypto ikev2 session detailed`	Displays the status of active Internet Key Exchange Version 2 (IKEv2) sessions.

Configuration Examples for Aggregate Authentication

Example: Configuring Aggregate Authentication

The following example shows how to configure aggregate authentication on the FlexVPN server to enable the establishment of a secure tunnel between Cisco AnyConnect Client and FlexVPN server.

```
Device> enable
Device# configure terminal
Device(config)# crypto ikev2 profile profile1
Device(config-ikev2-profile)# aaa accounting anyconnect-eap list1
Device(config-ikev2-profile)# match identity remote key-id aggauth_user1@example.com
Device(config-ikev2-profile)# authentication remote anyconnect-eap aggregate cert-request
Device(config-ikev2-profile)# authentication local rsa-sig
Device(config-ikev2-profile)# pki trustpoint CA1
Device(config-ikev2-profile)# aaa authentication anyconnect-eap list1
Device(config-ikev2-profile)# aaa authorization group anyconnect-eap list list1 name-mangler
 mangler1
Device(config-ikev2-profile)# aaa authorization user anyconnect-eap cached
Device(config-ikev2-profile)# aaa authorization user anyconnect-eap list list1 name-mangler
 mangler1
Device(config-ikev2-profile)# end
```

AdditionalReferencesforConfiguringAggregateAuthentication

Related Documents

Related Topic	Document Title
Cisco IOS commands	Cisco IOS Master Command List, All Releases

Related Topic	Document Title
Security commands	• Cisco IOS Security Command Reference Commands A to C • Cisco IOS Security Command Reference Commands D to L • Cisco IOS Security Command Reference Commands M to R • Cisco IOS Security Command Reference Commands S to Z

Technical Assistance

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

Feature Information for Configuring Aggregate Authentication

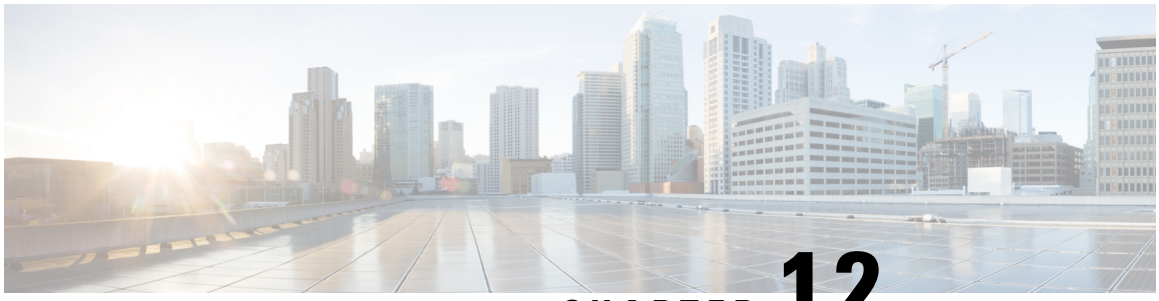
The following table provides release information about the feature or features described in this module. This table lists only the software release that introduced support for a given feature in a given software release train. Unless noted otherwise, subsequent releases of that software release train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

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

Table 13: Feature Information for Configuring Aggregate Authentication

Feature Name	Releases	Feature Information
Dual-Factor Authentication support with IKEv2		Dual-Factor Authentication support with IKEv2 supports certificate request from Cisco AnyConnect client for double authentication. The following command was modified: **authentication (IKEv2 profile)**.

Feature Name	Releases	Feature Information
FlexVPN RA - Aggregate Auth Support for AnyConnect		The FlexVPN RA - Aggregate Auth Support for AnyConnect feature implements aggregate authentication method by extending support for Cisco AnyConnect client that uses the proprietary AnyConnect EAP authentication method to establish a secure tunnel over the Internet between Cisco AnyConnect client and FlexVPN server. The following commands were introduced or modified: **aaa accounting (IKEv2 profile)**, **aaa authentication (IKEv2 profile)**, **aaa authorization (IKEv2 profile)**, **authentication (IKEv2 profile)**, **show crypto ikev2 profile**, **show crypto ikev2 session**.



CHAPTER 12

Appendix: FlexVPN RADIUS Attributes

This chapter describes the RADIUS attributes supported by FlexVPN server.

FlexVPN RADIUS Attributes

The following are the RADIUS attributes categories used by FlexVPN Server:

- Inbound and bidirectional IETF RADIUS attributes
- Outbound Local
- Outbound Remote

Note For inbound attributes sent by the FlexVPN server to RADIUS that are not listed below, the value is set by the AAA system.

Attribute	User-Name
Type	IETF
Format	String
Attribute ID	1
Description	This attribute is sent by the FlexVPN server to Radius and is derived as follows: • AAA based preshared keys—Peer IKEv2 identity • EAP authentication—Peer EAP identity • User or group authorization—Output of the name mangler or the string specified in the IKEv2 profile authorization commands • Accounting—Peer EAP identity or IKEv2 identity This attribute may also be received from Radius in Access-Accept after successful EAP authentication and specifies the authenticated peer EAP identity.

Attribute	User-Password
Type	IETF
Format	String
Attribute ID	2
Description	This attribute is sent by the FlexVPN server to RADIUS and is derived as follows: • AAA based preshared keys—“cisco” • User/group authorization—“cisco”

Attribute	Calling-Station-ID
Type	IETF
Format	String
Attribute ID	31
Description	This attribute is sent by FlexVPN server to RADIUS and is derived as follows: • AAA based pre-shared keys—IKEv2 initiator address • EAP authentication—IKEv2 initiator address • User/group authorization—IKEv2 initiator address

Attribute	Service-Type
Type	IETF
Format	String
Attribute ID	6
Description	This attribute is used by FlexVPN server for EAP authentication and the value of this attribute is set to ‘Login’.

Attribute	EAP-Message
Type	IETF
Format	String
Attribute ID	79
Description	This attribute is used by FlexVPN server for EAP authentication to relay EAP packets between EAP server and the Remote Access Client.

Attribute	Message-Authenticator

Type	IETF
Format	String
Attribute ID	80
Description	This attribute is sent by FlexVPN server for EAP authentication. The value for this attribute is set by AAA subsystem.

Attribute	Framed-Pool
Type	IETF
Format	String
Attribute ID	88
Local config	pool name
Radius config	Framed-Pool=*pool-name*
Description	Specifies the name of IPv4 address pool that is used by FlexVPN server to allocate the IPv4 address to assign to the client. The allocated address is pushed to client via IKEv2 standard config attribute INTERNAL_IP4_ADDRESS.

Attribute	ipsec:group-dhcp-server
Type	Cisco AV Pair
Format	String
Local config	dhcp server {*ipddr* | *host*}
Radius config	cisco-avpair="ipsec: group-dhcp-server=*ipaddr*"
Description	Specifies the IPv4 DHCP server that is used by FlexVPN server to lease IPv4 address to assign to the client. The leased address is pushed to client via IKEv2 standard config attribute INTERNAL_IP4_ADDRESS.

Attribute	ipsec:dhcp-giaddr
Type	Cisco AV Pair
Format	IPaddr
Local config	dhcp giaddr ipaddr
Radius config	cisco-avpair="psec: dhcp-giaddr=*ipaddr*"
Description	Specifies the IPv4 DHCP gateway IP address that is used by FlexVPN server to contact the DCHP server.

Attribute	ipsec:dhcp-timeout

Type	Cisco AV Pair
Format	Integer
Local config	dhcp timeout *seconds*
Radius config	cisco-avpair="ipsec:dhcp-timeout=*seconds*"
Description	Specifies the time to wait for response from IPv4 DHCP server that is used by FlexVPN server to timeout response from the DHCP server.

Attribute	ipsec:ipv6-addr-pool
Type	Cisco AV Pair
Format	String
Local config	ipv6 *pool name*
Radius config	cisco-avpair="ipsec:ipv6-addr-pool=*pool-name*"
Description	Specifies the name of IPv6 address pool used by FlexVPN server to allocate the IPv6 address to assign to the client. The allocated address is pushed to the client via IKEv2 standard config attribute INTERNAL_IP6_ADDRESS.

Attribute	ipsec:route-set=prefix
Type	Cisco AV Pair
Format	String
Local config	N/A
Radius config	cisco-avpair="ipsec:route-set=prefix *prefix*/*length*"
Example	ipsec:route-set=prefix 192.168.1.0/24
Description	Specifies a subnet protected by FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. **Note** This AV pair was introduced in Cisco IOS Release 15.2(2)T.

Attribute	ipsec:route-set=interface
Type	Cisco AV Pair
Format	String
Local config	route set interface
Radius config	cisco-avpair="ipsec:route-set=interface"

Description	This attribute is used locally and enables sending of VPN interface IP address to the peer via IKEv2 standard config attribute INTERNAL_IP4_SUBNET. This allows running routing protocols such as BGP over VPN. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the "ipsec:route-set-interface" AV pair.

Attribute	ipsec:route-accept
Type	Cisco AV Pair
Format	String
Local config	route accept any [tag *tag-id*] [distance *distance*]
Radius config	cisco-avpair="ipsec:route-accept=any [tag:*tag*] [distance:*distance*]"
Example	ipsec:route-accept=any tag=100
Description	This attribute is used locally and specifies the filter for the subnets received from the peer via IKEv2 standard config attribute INTERNAL_IP4_SUBNET. The attribute also specifies the tag and distance for the routes added by IKEv2 for the filtered subnets. **Note** In Cisco IOS Release 15.2(2)T, the AV pair "ipsec:route-accept=any" replaced "ipsec:route-accept=accept acl:any" and the AV pair "ipsec:route-accept=none" replaced "ipsec:route-accept=deny".

Attribute	ipsec:ipsec-flow-limit
Type	Cisco AV Pair
Format	Integer
Local config	ipsec flow-limit *limit*
Radius config	cisco-avpair="ipsec:ipsec-flow-limit=*limit*"
Description	This attribute is used by FlexVPN server and specifies the maximum number of IPsec SAs that an IPSec dVTI session can have. There is no limit by default. This parameter is similar to the **crypto ipsec profile** and **set security-policy limit** commands.

Attribute	ip:interface-config
Type	Cisco AV Pair
Format	String
Local config	aaa attribute list *list* attribute type interface-config *string*
Radius config	cisco-avpair="ip:interface-config=interface cmd string"
Example	ip:interface-config=ip vrf forwarding red

Description	This attribute is used locally and specifies an interface configuration mode command string that is applied on the virtual access interface for the session. For local configuration, the IKEv2 authorization policy points to an AAA attribute list that must have interface-config attribute.

Attribute	Tunnel-Type
Type	IETF
Format	Integer
Attribute ID	64
Radius config	Tunnel-Type=type
Description	This attribute specifies the tunnel type (ESP, AH, GRE, etc.) and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	Tunnel-Medium-Type
Type	IETF
Format	Integer
Attribute ID	65,
Radius config	Tunnel-Medium-Type=type
Description	This attribute specifies the tunnel transport type (IPv4, IPv6, etc.) and is received when FlexVPN server fetches preshared key for the session from the RADIUS server.

Attribute	Tunnel-Password
Type	IETF
Format	String
Attribute ID	69
Radius config	Tunnel-Password=string
Description	This attribute specifies the symmetric preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	ipsec:ikev2-password-local
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:ikev2-password-local=*string*"
Description	This attribute specifies the local preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	ipsec:ikev2-password-remote
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:ikev2-password-remote=*string*"
Description	This attribute specifies the remote preshared key and is received when FlexVPN server fetches preshared key for the session from RADIUS server.

Attribute	Framed-IP-Address
Type	IETF
Format	IPaddr
Attribute ID	8
Radius config	Framed-IP-Address=*ipaddr*
Description	Specifies IPv4 address assigned to the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_ADDRESS.

Attribute	Framed-IP-Netmask
Type	IETF
Format	IPaddr
Attribute ID	9
Local config	netmask *mask*
Radius config	Framed-IP-Netmask=*mask*
Description	Specifies the subnet mask of the IPv4 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP4_NETMASK.

Attribute	ipsec:dns-servers
Type	Cisco AV Pair
Format	String
Local config	dns *primary* [*secondary*]
Radius config	cisco-avpair="ipsec:dns-servers=*primary secondary*"
Description	Specifies the primary and secondary IPv4 DNS servers for the client. This is pushed to the client via IKEv2 standard config attribute INTERNAL_IP4_DNS.

Attribute	ipsec:wins-servers

Type	Cisco AV Pair
Format	String
Local config	wins *primary* [*secondary*]
Radius config	cisco-avpair="ipsec:wins-servers=*primary secondary*"
Description	Specifies the primary and secondary IPv4 WINS servers for the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_NBNS.

Attribute	ipsec:route-set=access-list
Type	Cisco AV Pair
Format	String
Local config	route set access-list {*acl-name* | *acl-number*}
Radius config	cisco-avpair="ipsec:route-set=access-list {*acl-name* | *acl-number*}"
Description	Specifies the IPv4 subnets protected by FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the "ipsec:inacl" AV pair.

Attribute	ipsec:addrv6
Type	Cisco AV Pair
Format	String
Radius config	cisco-avpair="ipsec:addrv6=*ipv6-addr*"
Description	Specifies the IPv6 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP6_ADDRESS in the first 16 bytes.

Attribute	ipsec:prefix-len
Type	Cisco AV Pair
Format	Integer
Local config	N/A
Radius config	cisco-avpair="ipsec:prefix-len=*value*"
Example	ipsec:prefix-len=24
Description	Specifies the prefix length of the IPv6 address assigned to the client. This is pushed to client via IKEv2 standard configuration attribute INTERNAL_IP6_ADDRESS in the last (17^{th}) byte.

Attribute	ipsec:ipv6-dns-servers-addr

Type	Cisco AV Pair
Format	String
Local config	ipv6 dns *primary* [*secondary*]
Radius config	cisco-avpair="ipsec: ipv6-dns-servers-addr=ipaddr1 *ipaddr2"
Description	Specifies the primary and secondary IPv6 DNS servers for the client. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP6_DNS.

Attribute	ipsec:route-set=access-list ipv6
Type	Cisco AV Pair
Format	String
Local config	route set access-list ipv6 acl-name
Radius config	cisco-avpair="ipsec:route-set=access-list ipv6 *acl-name*"
Description	Specifies IPv6 subnets protected by the FlexVPN server. This is pushed to the client via IKEv2 standard configuration attribute INTERNAL_IP6_SUBNET. **Note** In Cisco IOS Release 15.2(2)T, this AV pair replaced the " ipsec:ipv6-subnet-acl" AV pair.

Attribute	ipsec:banner
Type	Cisco AV Pair
Format	String
Local config	banner *text*
Radius config	cisco-avpair="ipsec:banner=*text*"
Description	Specifies the banner text. This is pushed to the client via Cisco Unity attribute MODECFG_BANNER.

Attribute	ipsec:default-domain
Type	Cisco AV Pair
Format	String
Local config	def-domain *name*
Radius config	cisco-avpair="ipsec:default-domain=*name*"
Description	Specifies the default domain. This is pushed to the client via Cisco Unity attribute MODECFG_DEFDOMAIN.

Attribute	ipsec:split-dns

Type	Cisco AV Pair
Format	String
Local config	split-dns name
Radius config	cisco-avpair="ipsec:split-dns=name"
Description	Specifies the split DNS name. This is pushed to the client via Cisco Unity attribute MODECFG_SPLITDNS_NAME. You can configure up to 10 split DNS names.

Attribute	ipsec:ipsec-backup-gateway
Type	Cisco AV Pair
Format	String
Local config	backup-gateway *name*
Radius config	cisco-avpair="ipsec:ipsec-backup-gateway=*name*"
Description	Specifies the backup gateway. This is pushed to the client via Cisco Unity attribute MODECFG_BACKUPSERVERS. You can configure up to 10 backup gateways.

Attribute	ipsec:pfs
Type	Cisco AV Pair
Format	Integer
Local config	pfs
Radius config	cisco-avpair="ipsec:pfs=*value*"
Description	Specifies IPsec PFS (Perfect Forward Secrecy) enable/disable. This is pushed to the client via Cisco Unity attribute MODECFG_PFS. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:include-local-lan
Type	Cisco AV Pair
Format	Integer
Local config	include-local-lan
Radius config	cisco-avpair="ipsec:include-local-lan=*value*"
Description	Enables or disables include local LAN. This is pushed to the client via Cisco Unity attribute MODECFG_INCLUDE_LOCAL_LAN. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:smartcard-removal-disconnect
Type	Cisco AV Pair

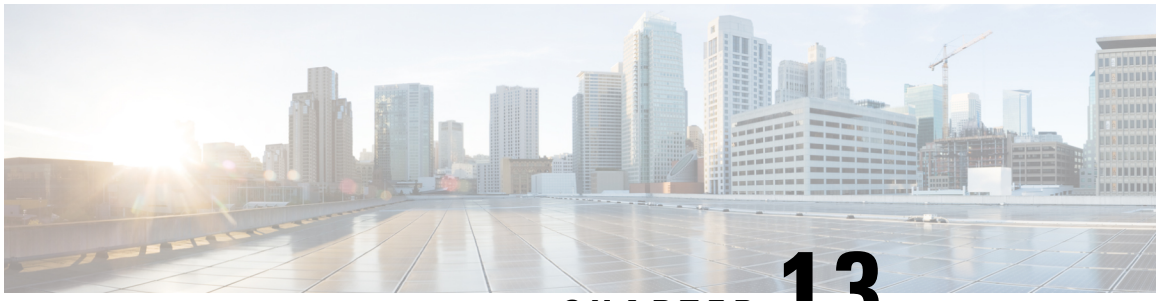
Format	Integer
Local config	smartcard-removal-disconnect
Radius config	cisco-avpair="ipsec:smartcard-removal-disconnect =*value*"
Description	Enables or disables smartcard removal disconnect. This is pushed to the client via Cisco Unity attribute MODECFG_SMARTCARD_REMOVAL_DISCONNECT. The value must be 0 to disable and 1 to enable.

Attribute	ipsec:configuration-url
Type	Cisco AV Pair
Format	String
Local config	configuration url *url*
Radius config	cisco-avpair="ipsec:configuration-url=*url*"
Description	Specifies the URL for configuration download. This is pushed to the client via Cisco FlexVPN attribute MODECFG_CONFIG_URL.

Attribute	ipsec:configuration-version
Type	Cisco AV Pair
Format	Integer
Local config	configuration version *version*
Radius config	cisco-avpair="ipsec:configuration-version=*version*"
Description	Specifies the version of the configuration to download. This is pushed to the client via Cisco FlexVPN attribute MODECFG_CONFIG_VERSION.

Attribute	Route-set remote
Type	Cisco AV Pair
Format	String
Local config	route set remote {ipv4 ip-address mask | ipv6 ip-address/mask}
Radius config	cisco-avpair="ipsec:route-set=remote {ipv4 network subnet_mask | ipv6 network/subnet_mask}"
Description	Specifies a subnet protected by FlexVPN server. This is pushed to the client through IKEv2 standard configuration attribute INTERNAL_IP4_SUBNET. While route-set prefix is working with a subnet mask represented in decimal fashion [e.g. /24], route-set remote requires the standard subnet mask representation. [e.g. 255.255.255.0] **Note** This AV pair was introduced in Cisco IOS Release 3.10.0S

Attribute	Route-set local
Type	Cisco AV Pair
Format	String
Local config	route set local {ipv4 ip-address mask | ipv6 ip-address/mask}
Radius config	cisco-avpair="ipsec:route-set=local {ipv4 network subnet_mask | ipv6 network/subnet_mask}”
Description	This attribute is useful in an extranet scenario where you do not necessary trust the routing information that you receive from the remote device. In other words, remote routes can be denied and route addition can be locally controlled by using this AV pair. **Note** This AV pair was introduced in Cisco IOS Release 3.10.0S.



CHAPTER 13

Appendix: IKEv2 and Legacy VPNs

This module provides examples on how to configure IKEv2 on crypto map based configurations.

Note Crypto maps are considered a legacy configuration construct. It is recommended that you migrate existing crypto map based setups to use tunnel protection and virtual interfaces.

Example: Configuring Crypto-Map-Based IKEv2 Peers Using Preshared Key Authentication Method

The following example shows how to configure crypto-map-based IKEv2 peers using the preshared key authentication method between a static crypto-map IKEv2 initiator and a dynamic crypto-map IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.231 255.255.255.224
  pre-shared-key abc
 !
!
!
```

```
crypto ikev2 profile prof
 match fvrf any
 match identity remote fqdn dmap-responder
 identity local fqdn smap-initiator
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.227 255.255.255.224
 crypto map cmap
!
ip route 209.165.200.229 255.255.255.224 209.165.200.225
!
ip access-list extended ikev2list
 permit ip any any
!
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.228
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote fqdn smap-initiator
 identity local fqdn dmap-responder
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
 ivrf global
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto dynamic-map dmap 1
 set transform-set trans
 set reverse-route tag 222
 set ikev2-profile prof
```

```
 match address ikev2list
!
crypto map cmap 1 ipsec-isakmp dynamic dmap
!
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
 crypto map cmap
!
ip route 209.165.200.233 255.255.255.224 209.165.200.228
!
ip access-list extended ikev2list
 permit ip any any
!
```

To initiate the connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

To display the session details, enter the following **show** commands:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 209.165.200.225 port 500
  IKEv2 SA: local 209.165.200.228/500 remote 209.165.200.231/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detail
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   209.165.200.228/500  209.165.200.231/500           (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK

      Life/Active Time: 86400/21 sec
      CE id: 1002, Session-id: 2
      Status Description: Negotiation done
      Local spi: 687752902752A6FD       Remote spi: C9DCCFC65493D14F
      Local id: smap-initiator
      Remote id: dmap-responder
      Local req msg id:  2              Remote req msg id:  0
      Local next msg id: 2              Remote next msg id: 0
      Local req queued:  2              Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring Crypto Map-Based IKEv2 Peers Using Certification Authentication Method

The following example shows how to configure crypto-map-based IKEv2 peers using the certificate authentication method between a static crypto-map IKEv2 initiator, a dynamic crypto-map IKEv2 responder, and a CA server. The initiator configuration is as follows:

```
crypto pki trustpoint ca-server
 enrollment url http://10.1.1.3:80
 revocation-check none
!
crypto pki certificate map cmap-1 1
 subject-name eq hostname = responder
!
!
crypto pki certificate chain ca-server
 certificate 02
  308201AF 30820118 A0030201 02020102 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32353132 355A170D 31313033 31303132 35313235 5A301A31 18301606 092A8648
  86F70D01 09021609 494E4954 4941544F 52305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100A47E 8C58BA89 8CCDC5A4 5A63BD29 C331A2A5 393F4616
  6B43FD2E 5ED4C81A 913E3B13 33A9B2DC CFC30391 24BB0DC8 B28FD6F1 C008D101
  34C10062 30F88CF7 9D630203 010001A3 4F304D30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 144871D9 002C66DF D85FACB8 45D1D25F EA357455
  91301D06 03551D0E 04160414 E77C74E7 183AB530 83DC531B 1DE3DA1D 914A925D
  300D0609 2A864886 F70D0101 04050003 81810042 21934B77 7E485E6F EE717D75
  6407B361 45190CEF E1A29CF2 6FA29E9A 5ECC1CEE B273533D 1453F6CE 1FDDA747
  7E701B4B 2A2AE53F D67C2345 952325BA 30950435 0706C5EE A7A8B414 CFEEB7A2
  9CD46F8F 3F663268 A20C4CCF E75D61EF 03FBA85D EDD6B26E 63653F09 F97DAFA6
  6C76E44E C9CA3FDC 6CD85D30 169A1D9E 4E870B
        quit
 certificate ca 01
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32343933 385A170D 31333033 30393132 34393338 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 DA4ECE09 B998F670 598F32C1 7E9FA920 1D217AC4 293B842E
  7563CE11 B2F0F822 23077930 636C8293 00F6CFDD F6C9B0F5 8348BE58 6478F631
  7D44152F 494AEBCC A507FA6B 408D6BBB FAAB0A7A 2E7546A8 CA70F9A6 0F7F6824
  554BD833 060D657D ABDF406C 69EEF449 7A4F9AFE 6F0852E7 05DEDAC1 D433191E
  712868C2 A94E642B 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 16801448
  71D9002C 66DFD85F ACB845D1 D25FEA35 74559130 1D060355 1D0E0416 04144871
  D9002C66 DFD85FAC B845D1D2 5FEA3574 5591300D 06092A86 4886F70D 01010405
  00038181 00AFC36B 8A917284 06BD51CB 83BDC4E8 9457A361 6CAAF416 3BBEF691
  04215AC5 EDBC5730 C071C2FB 8A6C90CF D6AB39C2 3BC2147F D35553D9 028B2155
  802E50DB 48CDE067 B3857447 89A1C733 D81EFEF7 1115480F 70ED2F22 F27E35A1
  F3BB597C 7C8F717B FAAD79D3 0F469702 DE9190E4 B1B0808E 46A118EB 887CEAEB
  DFE2900E D2
        quit
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
```

```
!
crypto ikev2 profile prof
 match fvrf any
 match certificate cmap-1
 identity local dn
 authentication local rsa-sig
 authentication remote pre-share
 authentication remote rsa-sig
 pki trustpoint ca-server
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 209.165.200.225
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.227 255.255.255.224
 crypto map cmap
!
interface Ethernet1/0
 ip address 209.165.200.228 255.255.255.224
!
ip route 209.165.200.229 255.255.255.224 209.265.200.231
!
ip access-list extended ikev2list
 permit ip any any
!
```

The responder configuration is as follows:

```
crypto pki trustpoint ca-server
 enrollment url http://10.1.1.3:80
 revocation-check none
!
!
!
crypto pki certificate map cmap-2 1
 subject-name eq hostname = initiator
!
crypto pki certificate chain ca-server
 certificate 03
  308201AF 30820118 A0030201 02020103 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32353231 325A170D 31313033 31303132 35323132 5A301A31 18301606 092A8648
  86F70D01 09021609 52455350 4F4E4445 52305C30 0D06092A 864886F7 0D010101
  0500034B 00304802 4100B517 EB8E64E1 B58CB014 07B3A6AF E6B69577 87486367
  9471B1DA BC66B847 DFA5073A 82121332 E787EA2D 3C433514 39033074 4095E7C7
  67A387A1 EBD24692 A76F0203 010001A3 4F304D30 0B060355 1D0F0404 030205A0
  301F0603 551D2304 18301680 144871D9 002C66DF D85FACB8 45D1D25F EA357455
  91301D06 03551D0E 04160414 DFF2401C 53276D96 89DE8C0A 786CCA71 C9EA792B
  300D0609 2A864886 F70D0101 04050003 8181002C 6E334273 CB832A95 3DDC6293
  669E416C A134D543 20952BC3 14A5C0B0 03AE011C 963AF523 C7C5C935 4FE9B2A5
  F24B3161 4D0D723A FA428BD1 85ADF172 B4007067 43C27D8A 1F74ED3D DEBE9F73
  1F515355 E77E766C AEACC303 39457991 29AB090C 99E21B5B 60DCB2C8 780B4479
  3EB3D46B B66C8C26 15311A7A B7A4ED97 32727C
        quit
 certificate ca 01
```

```
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D313030 33313031
  32343933 385A170D 31333033 30393132 34393338 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 DA4ECE09 B998F670 598F32C1 7E9FA920 1D217AC4 293B842E
  7563CE11 B2F0F822 23077930 636C8293 00F6CFDD F6C9B0F5 8348BE58 6478F631
  7D44152F 494AEBCC A507FA6B 408D6BBB FAAB0A7A 2E7546A8 CA70F9A6 0F7F6824
  554BD833 060D657D ABDF406C 69EEF449 7A4F9AFE 6F0852E7 05DEDAC1 D433191E
  712868C2 A94E642B 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 16801448
  71D9002C 66DFD85F ACB845D1 D25FEA35 74559130 1D060355 1D0E0416 04144871
  D9002C66 DFD85FAC B845D1D2 5FEA3574 5591300D 06092A86 4886F70D 01010405
  00038181 00AFC36B 8A917284 06BD51CB 83BDC4E8 9457A361 6CAAF416 3BBEF691
  04215AC5 EDBC5730 C071C2FB 8A6C90CF D6AB39C2 3BC2147F D35553D9 028B2155
  802E50DB 48CDE067 B3857447 89A1C733 D81EFEF7 1115480F 70ED2F22 F27E35A1
  F3BB597C 7C8F717B FAAD79D3 0F469702 DE9190E4 B1B0808E 46A118EB 887CEAEB
  DFE2900E D2
        quit
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
!
crypto ikev2 profile prof
 match fvrf any
 match certificate cmap-2
 identity local dn
 authentication local rsa-sig
 authentication remote pre-share
 authentication remote rsa-sig
 pki trustpoint ca-server
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto dynamic-map dmap 1
 set transform-set trans
 set ikev2-profile prof
!
!
crypto map cmap 1 ipsec-isakmp dynamic dmap
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
 crypto map cmap
!
interface Ethernet1/0
 ip address 209.165.200.232 255.255.255.224
!
ip route 209.165.200.233 255.255.255.224 209.165.200.228
!
ip access-list extended ikev2list
 permit ip host 209.165.200.231 host 209.165.200.228
```

The CA server configuration is as follows:

```
crypto pki server ca-server
```

```
 grant auto
!
crypto pki trustpoint ca-server
 revocation-check crl
 rsakeypair ca-server
!
!
crypto pki certificate chain ca-server
 certificate ca 01
  30820201 3082016A A0030201 02020101 300D0609 2A864886 F70D0101 04050030
  14311230 10060355 04031309 63612D73 65727665 72301E17 0D303930 33303831
  36333335 395A170D 31323033 30373136 33333539 5A301431 12301006 03550403
  13096361 2D736572 76657230 819F300D 06092A86 4886F70D 01010105 0003818D
  00308189 02818100 99750598 EF4AF8B4 823DEF66 2F3BBA31 81C2DC5F D9B4040B
  99FB6020 22243CD6 B9F24C84 A543D7DB DD0B3018 2E36208C D0FD4015 EAF0DA69
  C1B0302B 87CEC34B 8646593F 0185AF02 0B86A3F3 5E5C3880 A992CD4A 79F13403
  411CC61F 07CEB4D9 0E967CB2 FAE0A899 5A3B6C87 73111F06 128465DA A45291F8
  F828C5DC 657487E7 02030100 01A36330 61300F06 03551D13 0101FF04 05300301
  01FF300E 0603551D 0F0101FF 04040302 0186301F 0603551D 23041830 1680147B
  D032BFB7 B3F70F1A 597B7C1E 1B42E472 5CCD6030 1D060355 1D0E0416 04147BD0
  32BFB7B3 F70F1A59 7B7C1E1B 42E4725C CD60300D 06092A86 4886F70D 01010405
  00038181 003838FA 628804EF E9FF69D9 3D5E299C 29074B2C AE33A563 8AF75976
  78FB68D4 5EF1E27B 04936FDF 78A09432 5348849D F79E17F5 70B233C9 2C1535D0
  506F0C35 99335012 84BBA3DC 050FD3C9 6E7B1D63 41ACC2B5 2B02432D BA2CC2CF
  E379DEA0 A9C208AC 0BEBB2D8 E6488815 EB12F1E0 19072D55 D5D11A49 739144D8
  271A842E ED
        quit
!
interface Ethernet1/0
 ip address 209.165.200.232 255.255.255.224
!
ip http server
```

To obtain the CA and device certificates, enter the **crypto pki authenticate ca-server** and **crypto pki enroll ca-server** commands. To initiate a connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
```

The output of the command is as follows:

```
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** commands in the responder's CLI to display the session details:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 1.1.1.1 port 500
  IKEv2 SA: local 209.165.200.231/500 remote 209.165.200.227/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 host 209.165.200.226
        Active SAs: 2, origin: dynamic crypto map
```

```
show crypto ikev2 sa detailed
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1         209.165.200.231/500 209.165.200.227/500  (none)/(none)                  READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: RSA, Auth verify: RSA

      Life/Active Time: 86400/846 sec
      CE id: 1001, Session-id: 1
      Status Description: Negotiation done
      Local spi: F79756E978ED41C7       Remote spi: 188FB9A119516D34
      Local id: hostname=RESPONDER
      Remote id: hostname=INITIATOR
      Local req msg id:  0              Remote req msg id:  2
      Local next msg id: 0              Remote next msg id: 2
      Local req queued:  0              Remote req queued:  2
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring Crypto Map- and dVTI-Based IKEv2 Peers

The following example shows how to configure crypto map-and dVTI-based IKEv2 peers using the preshared key authentication method between a static crypto map IKEv2 initiator and a dVTI-based IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 0.0.0.0 0.0.0.0
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto map cmap 1 ipsec-isakmp
 set peer 206.165.200.235
 set transform-set trans
 set ikev2-profile prof
 match address ikev2list
!
interface Loopback0
 ip address 206.165.200.226 255.255.255.224
!
interface Ethernet0/0
 ip address 206.165.200.227 255.255.255.224
```

```
 crypto map cmap
!
ip route 206.165.200.229 255.255.255.224 206.165.200.235
!
ip access-list extended ikev2list
 permit ip host 206.165.200.227 host 206.165.200.235
 permit ip 206.165.200.233 255.255.255.224 206.165.200.229 255.255.255.224
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer cisco
  address 0.0.0.0 0.0.0.0
  pre-shared-key cisco
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 0.0.0.0
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
 virtual-template 1
!
crypto ipsec transform-set set esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile vi
 set transform-set set
 set ikev2-profile prof
!
interface Loopback0
 ip address 206.165.200.230 255.255.255.224
!
interface Ethernet0/0
 ip address 206.165.200.235 255.255.255.224
!
interface Virtual-Template1 type tunnel
 ip unnumbered Ethernet0/0
 ip mtu 1000
 tunnel source Ethernet0/0
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile vi
!
```

To initiate a connection between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 206.165.200.230 source 206.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 206.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 206.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 206.165.200.226-206.165.200.226 Protocol: 1 Port Range: 0-65535;
```

```
 remote traffic selector = Address Range: 206.165.200.230-206.165.200.230 Protocol: 1 Port
 Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** command in an Easy VPN server to display the session details:

```
show crypto session
Crypto session current status
Interface: Virtual-Access2
Session status: UP-ACTIVE
Peer: 206.165.200.227 port 500
  IKEv2 SA: local 206.165.200.235/500 remote 206.165.200.227/500 Active
  IPSEC FLOW: permit ip 206.165.200.229/255.255.255.224 206.165.200.233/255.255.255.224
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detail
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   206.165.200.235/500      206.165.200.227/500       (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp: 14, Auth sign: PSK, Auth verify:
PSK
      Life/Active Time: 86400/8 sec
      CE id: 1001, Session-id: 1
      Status Description: Negotiation done
      Local spi: 305F610F57428834       Remote spi: D9D183B5689AEDCD
      Local id: 206.165.200.235
      Remote id: 206.165.200.227
      Local req msg id:  0              Remote req msg id:  2
      Local next msg id: 0              Remote next msg id: 2
      Local req queued:  0              Remote req queued:  2
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
show crypto route
VPN Routing Table: Shows RRI and VTI created routes
Codes: RRI - Reverse-Route, VTI- Virtual Tunnel Interface
        S - Static Map ACLs
Routes created in table GLOBAL DEFAULT
206.165.200.233/255.255.255.224 [1/0] via 206.165.200.227 tag 0
                                on Virtual-Access2 RRI
```

Example: Configuring IPsec Using sVTI-Based IKEv2 Peers

The following example shows how to configure IPsec using the preshared key authentication method between an sVTI IKEv2 initiator and an sVTI IKEv2 responder. The initiator configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.225
  pre-shared-key abc
 !
```

```
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 209.165.200.231 255.255.255.224
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set ikev2-profile prof
!
interface Loopback0
 ip address 209.165.200.226 255.255.255.224
!
interface Tunnel0
 ip address 10.0.0.1 255.255.255.0
 tunnel source 209.165.200.231
 tunnel mode ipsec ipv4
 tunnel destination 209.165.200.225
 tunnel protection ipsec profile ipsecprof
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
!
ip route 209.165.200.229 255.255.255.224 Tunnel0
!
```

The responder configuration is as follows:

```
crypto ikev2 proposal prop-1
 encryption aes-cbc-128
 integrity sha1
 group 14
!
crypto ikev2 policy pol-1
 match fvrf any
 proposal prop-1
!
crypto ikev2 keyring v2-kr1
 peer abc
  address 209.165.200.231
  pre-shared-key abc
 !
!
!
crypto ikev2 profile prof
 match fvrf any
 match identity remote address 209.165.200.231 255.255.255.224
 authentication local pre-share
 authentication remote pre-share
 keyring v2-kr1
!
!
crypto ipsec transform-set trans esp-aes-cbc-128 esp-sha-hmac
!
crypto ipsec profile ipsecprof
 set transform-set trans
 set ikev2-profile prof
!
```

```
crypto map cmap 1 ipsec-isakmp dynamic dmap
!
interface Loopback0
 ip address 209.165.200.230 255.255.255.224
!
interface Tunnel0
 ip address 10.0.0.2 255.255.255.0
 tunnel source 209.165.200.225
 tunnel mode ipsec ipv4
 tunnel destination 209.165.200.231
 tunnel protection ipsec profile ipsecprof
!
interface Ethernet0/0
 ip address 209.165.200.231 255.255.255.224
!
ip route 209.165.200.233 255.255.255.224 Tunnel0
```

With sVTI on IKEv2 peers, the session is initiated only when the sVTI interfaces are enabled. In other words, network traffic is not required to initiate the session. To verify the traffic between the initiator and the responder, enter the following command at the initiator's CLI:

```
ping 209.165.200.230 source 209.165.200.226
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 209.165.200.230, timeout is 2 seconds:
Packet sent with a source address of 209.165.200.226
%IKEV2-5-OSAL_INITIATE_TUNNEL: Received request to establish an IPsec tunnel; local traffic
 selector = Address Range: 209.165.200.226-209.165.200.226 Protocol: 1 Port Range: 0-65535;
 remote traffic selector = Address Range: 209.165.200.230-209.165.200.23 Protocol: 1 Port
Range: 0-65535
%IKEV2-5-SA_UP: SA UP
.!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/11/12 ms
```

Enter the following **show** command in the initiator's CLI to display the session details:

```
show crypto session
Crypto session current status
Interface: Ethernet0/0
Session status: UP-ACTIVE
Peer: 209.165.200.225 port 500
  IKEv2 SA: local 209.165.200.231/500 remote 209.165.200.225/500 Active
  IPSEC FLOW: permit ip 0.0.0.0/0.0.0.0 0.0.0.0/0.0.0.0
        Active SAs: 2, origin: crypto map
show crypto ikev2 sa detailed
Tunnel-id Local                 Remote                fvrf/ivrf            Status
1   209.165.200.231/500  209.165.200.225/500           (none)/(none)        READY
      Encr: AES-CBC, Keysize: 128, Hash: SHA96, DH Grp: 14, Auth sign: PSK, Auth verify:
PSK
      Life/Active Time: 86400/21 sec
      CE id: 1002, Session-id: 2
      Status Description: Negotiation done
      Local spi: 687752902752A6FD       Remote spi: C9DCCFC65493D14F
      Local id: smap-initiator
      Remote id: dmap-responder
      Local req msg id:  2              Remote req msg id:  0
      Local next msg id: 2              Remote next msg id: 0
      Local req queued:  2              Remote req queued:  0
      Local window:      5              Remote window:      5
      DPD configured for 0 seconds, retry 0
      NAT-T is not detected
```

Example: Configuring IKEv2 on DMVPN Networks

DMVPN uses a tunnel protection CLI that is identical between IKEv1 and IKEv2. The IPsec profile applied on a DMVPN tunnel only refers to an IKEv2 profile. The DMVPN Hub configuration is as follows:

```
crypto ikev2 keyring cisco-ikev2-keyring
 peer dmvpn-node
 description symmetric pre-shared key for the hub/spoke
 address 0.0.0.0 0.0.0.0
 pre-shared-key cisco123
crypto ikev2 profile cisco-ikev2-profile
 keyring cisco-ikev2-keyring
 authentication pre-shared
 match local address 0.0.0.0
crypto ipsec profile cisco-ipsec-ikev2
 set transform-set cisco-ts
 set ikev2-profile cisco-ikev2-profile
! interface Tunnel 0
 description This is the Legacy IKEv1 facing tunnel on the hub
 ip address 1.1.1.99 255.255.255.0
 no ip redirects
 ip nhrp map multicast dynamic
 ip nhrp network-id 99
 ip nhrp redirect
 no ip split-horizon eigrp 1
 tunnel source Ethernet0/0
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec
 !
interface Tunnel1
 description This would be the new IKEv2 facing tunnel on the hub
 ip address 2.2.2.99 255.255.255.0
 no ip redirects
 ip nhrp map multicast dynamic
 ip nhrp network-id 100
 no ip split-horizon eigrp 1
 tunnel source Ethernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec-ikev2
```

The IKEv2 configuration is as follows:

```
crypto ikev2 profile cisco-ikev2-profile
 keyring cisco-ikev2-keyring
 authentication pre-shared
 match local address 0.0.0.0
crypto ipsec profile cisco-ipsec-ikev2
 set transform-set cisco-ts
 set ikev2-profile cisco-ikev2-profile
interface Tunnel1
 ip address 2.2.2.11 255.255.255.0
 no ip redirects
 ip nhrp map 2.2.2.99 22.22.22.99
 ip nhrp map multicast 22.22.22.99
 ip nhrp network-id 100 ? Keep this same for all IKEv2 spokes for clarity
 ip nhrp nhs 2.2.2.99 ? This points to the hub's IKEv2 facing interface
 tunnel source Ethernet0/1
 tunnel mode gre multipoint
 tunnel protection ipsec profile cisco-ipsec-ikev2
```