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 www.cisco.com/go/cfn. 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

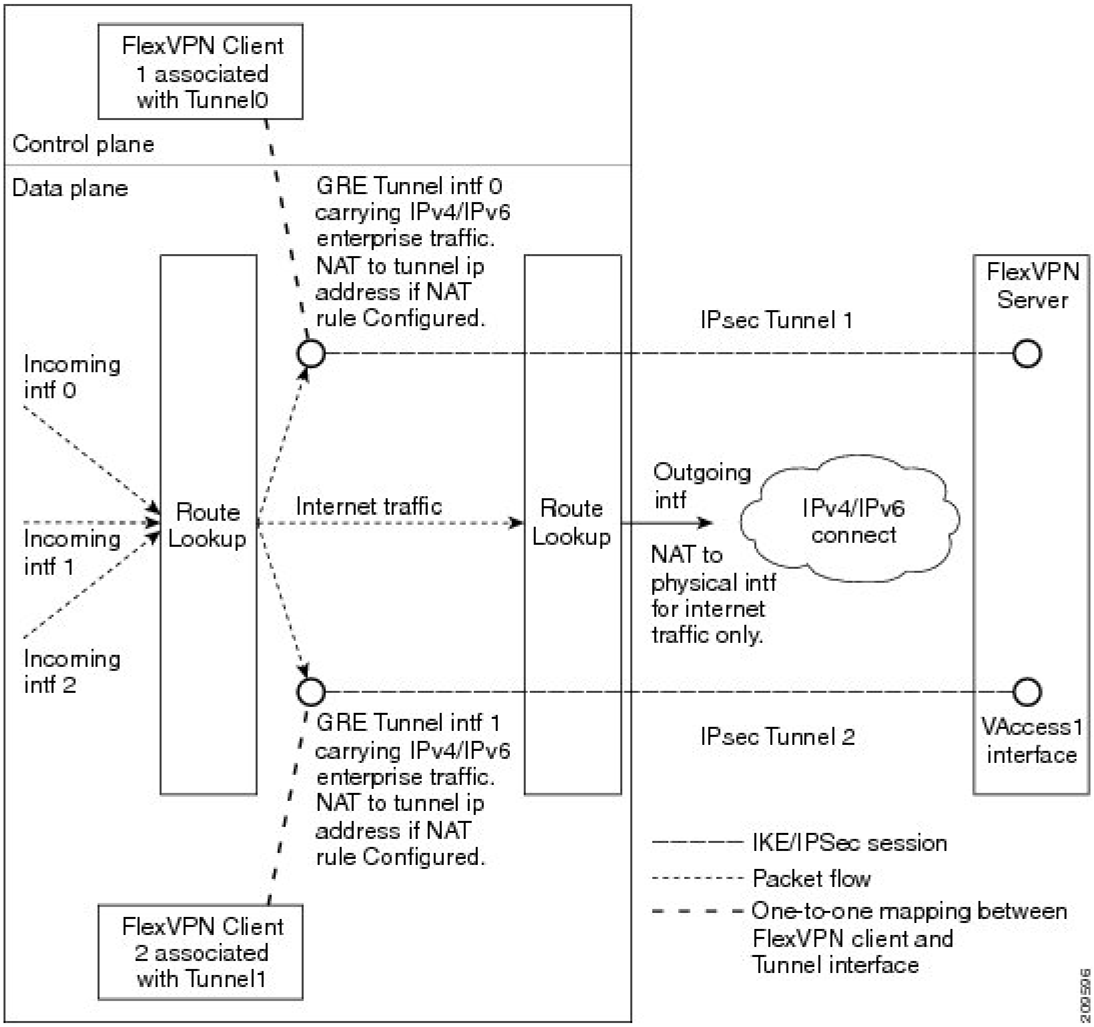
IKEv2 FlexVPN Client

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 1: Association of the FlexVPN Client and the Tunnel Interface


FlexVPN Client 1 associated with Tunnel0
Control plane
Data plane
GRE Tunnel intf 0 carrying IPv4/IPv6 enterprise traffic. NAT to tunnel ip address if NAT rule Configured.
FlexVPN Server
IPsec Tunnel 1
Incoming intf 0
Route Lookup
Internet traffic
Route Lookup
Outgoing intf
IPv4/IPv6 connect
Incoming intf 1
NAT to physical intf for internet traffic only.
Incoming intf 2
GRE Tunnel intf 1 carrying IPv4/IPv6 enterprise traffic. NAT to tunnel ip address if NAT rule Configured.
IPsec Tunnel 2
VAccess1 interface
IKE/IPSec session
Packet flow
One-to-one mapping between FlexVPN client and Tunnel interface
FlexVPN Client 2 associated with Tunnel1
209596

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 with FlexVPN site-to-site 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.

IKEv2 Dynamic Routing

With IKEv2 static routing, route information is exchanged during initial session bring up. The IKEv2 Dynamic Routing Support feature enables exchange of route information even after a session is established. Changes in routing information such as new routes, addition or deletion of routes can be propagated from FlexVPN client to FlexVPN server. The route information is included in the IKEv2 information exchange messages.

This feature is supported only on FlexVPN client and can be configured using the **route redistribute** command in the IKEv2 authorization policy configuration mode. This feature is supported on FlexVPN client on Cisco IOS Release 15.6(3)M2 and later, and the software release on FlexVPN server must be Cisco IOS XE Everest 16.5.1 and later.

The IKEv2 Dynamic Routing feature supports the following:

- IPv6 connected routes only.
- The following route map scenarios are supported:
    - Route matching is performed through the **match ipv6-address** command only.
    - Route map support with local and external AAA attributes.
    - Modifying route-map configurations while the crypto sessions are active can cause route leaks for redistributed routes.
- The redistribute protocol can be pushed via AAA also. The AAA string would be as follows: **ipsec:route-redistribute=***protocol-name* [**route-map** *route-map-name*]. The **route-map** command must be configured on the device already.
- Route dampening can be configured or set to the default value.
- Route injection in the same VRF to which the virtual access interface belongs.
- Upto 100 routes can be configured per session using the **crypto ikev2 route redistribute update-per-msg** command. If the number of routes exceeds 100, multiple information exchange messages are sent, with each message having 100 or less routes.

The following restrictions apply to the IKEv2 Dynamic Routing feature:

- Available on FlexVPN client only. This feature should not be used with Static Virtual Tunnel Interface (SVTI) or dynamic Virtual Tunnel Interface (DVTI), and should not be configured on FlexVPN server.
- The LAN side must be IPv6 address, while the WAN side can be IPv4 or IPv6 address.
- **set** commands in **route-map** are not supported.
- VRF support for IKEv2 Dynamic routing from FlexVPN client is not supported.
- **route accept prefix** command on FlexVPN server and FlexVPN client is not supported.
- Route properties, such as, distance, tags are not propogated with the routes.

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)and FlexVPN Site-to-Site* 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.

	Command or Action	Purpose
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.

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:** `Device(config)# crypto ikev2 client flexvpn client1`	Defines an IKEv2 FlexVPN client profile and enters IKEv2 FlexVPN client profile configuration mode.
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**]}	Connects the FlexVPN tunnel.

	Command or Action	Purpose
	Example: `Device(config-ikev2-flexvpn)# connect track 10 up`	**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.
		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.

Configuring IKEv2 Dynamic Routing

Perform this task to configure IKEv2 dynamic routing on FlexVPN client.

Note IKEv2 dynamic routing can be configured on FlexvPN client only. This feature should not be used with Static Virtual Tunnel Interface (SVTI) or dynamic Virtual Tunnel Interface (DVTI), and should not be configured on FlexVPN server.

SUMMARY STEPS

1. **enable**
2. **configure terminal**
3. **crypto ikev2 route redistribute** {**update-interval** *seconds* | **update-per-msg** *number*}
4. **crypto ikev2 authorization policy** *policy-name*
5. **aaa attribute list** *list-name*
6. **route redistribute** *protocol* [**route-map** *map-name*]
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	**crypto ikev2 route redistribute** {**update-interval** *seconds* | **update-per-msg** *number*}
	Example:
	`Device(config)# crypto ikev2 route redistribute update-interval 5`
		Configures the update interval and update per message (IKEv2 Information Exchange message) for IKEv2 dynamic routing.
		• **update-interval** *seconds*—Duration between the route redistribution, in seconds. The default is two seconds. Range: 1 to 60.
		• **update-per-msg** *number*—Number of routes permitted in a message. Range: 1 to 100.

Step 4	**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 5	**aaa attribute list** *list-name*	Specifies an AAA attribute list.

	Command or Action	Purpose
	Example: `Device(config-ikev2-author-policy)# aaa attribute list list1`	**Note** The AAA attribute list referred to in this command should be defined in global configuration mode.
Step 6	**route redistribute** *protocol* [**route-map** *map-name*] **Example:** `Device(config-ikev2-author-policy)# route redistribute connected`	Enables route redistribution for the configured protocol. • *protocol*—Source protocol from which routes are redistributed. Only **connected** is supported. • **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 7	**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 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
```

Example: Configuring the IKEv2 FlexVPN Client Profile with Dynamic Routing

The following example shows how to configure the IKEv2 FlexVPN client profile for IKEv2 Dynamic Routing Support feature:

```
crypto ikev2 client flexvpn flex
  peer 1 2001:DB8::1
  connect manual
  client connect Tunnel0
!
crypto ikev2 authorization policy flex
  route set interface
  route accept any
  route redistribute connected
!
crypto ikev2 keyring key
peer dvti
  address ::/0
  pre-shared-key cisco
!
crypto ikev2 profile prof
  match identity remote address ::/0
  authentication local pre-share
  authentication remote pre-share
  keyring local key
  aaa authorization group psk list local-group-author-list flex
!
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 gre ipv6
  tunnel protection ipsec-profile ipsecprof
!
interface Ethernet0/0
  ip address 172.16.0.1 255.240.0.0
  ipv6 address 2001:B8:123:2::1/64
  ipv6 enable
!
ip route 0.0.0.0 0.0.0.0 172.16.0.2
!
ipv6 unicast-routing
ipv6 route ::/0 2001:B8:123:2::2
```

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 1: Feature Information for Configuring FlexVPN Client

Feature Name	Releases	Feature Information
IKEv2 Remote Access Hardware Client	Cisco IOS XE Release 3.7S	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.**

Feature Name	Releases	Feature Information
IKEv2 Dynamic Routing Support	15.6(3)M2	With IKEv2 static routing, route information is exchanged during initial session bring up. The IKEv2 Dynamic Routing Support feature enables exchange of route information even after a session is established. Changes in routing information such as new routes, addition or deletion of routes can be propagated from FlexVPN client to FlexVPN server. The route information is included in the IKEv2 information exchange messages. The following commands were introduced or modified: **crypto ikev2 route redistribute**, **route redistribute**, **show crypto ikev2 sa**, **show crypto session**.
IPv6 Remote Access for IPsec VPN	Cisco IOS XE Release 3.8S	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.**