

Remote Access IPsec VPNs

- Overview of Remote Access IPsec VPNs, on page 1
- Licensing Requirements for AnyConnect VPN Module of Cisco Secure Client, on page 3
- Restrictions for Remote Access IPsec VPN, on page 3
- Configure Remote Access IPsec VPNs, on page 3
- VPN Authentication Using Post-Quantum Pre-Shared Keys, on page 10
- Configure Portal Access Rules, on page 15
- Configure VPN Access of Remote Users based on Geolocation, on page 16
- Configuration Examples for Remote Access IPsec VPNs, on page 18
- Configuration Examples for Standards-Based IPSec IKEv2 Remote Access VPN in Multicontext Mode, on page 19
- Configuration Examples for Secure Client IPSec IKEv2 Remote Access VPN in Multicontext Mode, on page 20
- Feature History for Remote Access VPNs, on page 22

Overview of Remote Access IPsec VPNs

Remote access VPNs allow users to connect to a central site through a secure connection over a TCP/IP network. The Internet Security Association and Key Management Protocol, also called IKE, is the negotiation protocol that lets the IPsec client on the remote PC and the ASA agree on how to build an IPsec Security Association. Each ISAKMP negotiation is divided into two sections called Phase1 and Phase2.

Phase 1 creates the first tunnel to protect later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data travelling across the secure connection.

To set the terms of the ISAKMP negotiations, you create an ISAKMP policy. It includes the following:

- An authentication method, to ensure the identity of the peers.
- An encryption method, to protect the data and ensure privacy.
- A Hashed Message Authentication Codes (HMAC) method to ensure the identity of the sender and to ensure that the message has not been modified in transit.
- A Diffie-Hellman group to set the size of the encryption key.
- A time limit for how long the ASA uses an encryption key before replacing it.

A transform set combines an encryption method and an authentication method. During the IPsec security association negotiation with ISAKMP, the peers agree to use a particular transform set to protect a particular data flow. The transform set must be the same for both peers.

A transform set protects the data flows for the ACL specified in the associated crypto map entry. You can create transform sets in the ASA configuration, and then specify a maximum of 11 of them in a crypto map or dynamic crypto map entry. For more overview information, including a table that lists valid encryption and authentication methods, see Create an IKEv1 Transform Set or IKEv2 Proposal, on page 6.

You can configure the ASA to assign an IPv4 address, an IPv6 address, or both an IPv4 and an IPv6 address to the Secure Client by creating internal pools of addresses on the ASA or by assigning a dedicated address to a local user on the ASA.

The endpoint must have the dual-stack protocol implemented in its operating system to be assigned both types of addresses. In both scenarios, when no IPv6 address pools are left but IPv4 addresses are available or when no IPv4 address pools are left but IPv6 addresses are available, connection still occurs. The client is not notified; however, so the administrator must look through the ASA logs for the details.

Assigning an IPv6 address to the client is supported for the SSL protocol.

About Mobike and Remote Access VPNs

Mobile IKEv2 (mobike) extends ASA RA VPNs to support mobile device roaming. This support means the end-point IP address for a mobile device's IKE/IPSEC security association (SA) can be updated rather than deleted when the device moves from its current connection point to another.

Mobike is available by default on ASAs since version 9.8(1), meaning Mobike is "always on." Mobike is enabled for each SA only when the client proposes it and the ASA accepts it. This negotiation occurs as part of the IKE_AUTH exchange.

After the SA is established with mobike support as enabled, client can change its address anytime and notify the ASA using the INFORMATIONAL exchange with UPDATE_SA_ADDRESS payload indicating the new address. The ASA will process this message and update the SA with the new client IP address.



Note You can use the `show crypto ikev2 sa detail` command to determine whether mobike is enabled for all current SAs.

The current Mobike implementation supports the following:

- IPv4 addresses only
- Changes in NAT mappings
- Path connectivity and outage detection, by means of optional Return Routability checking
- Active/standby failover
- VPN load balancing

If the Return Routability Check (RRC) feature is enabled, an RRC message is sent to the mobile client to confirm the new IP address before the SA is updated.

Licensing Requirements for AnyConnect VPN Module of Cisco Secure Client

Note This feature is not available in No Payload Encryption models.

If you want to deploy Cisco Secure Client (including AnyConnect) from a Secure Firewall ASA headend and use the VPN and Secure Firewall Posture or HostScan modules, an Advantage or Premier license is required. Trial licenses are available. See the Cisco Secure Client Ordering Guide. See Cisco ASA Series Feature Licenses for maximum values per model.

Restrictions for Remote Access IPsec VPN

- Firewall Mode Guidelines-Supported only in routed firewall mode. Transparent mode is not supported.
- Failover Guidelines IPsec-VPN sessions are replicated in Active/Standby failover configurations only. Active/Active failover configurations are not supported.
- Configuration changes are blocked during HA synchronization. If a user attempts to log in during that time, the DACL rule installation in the firewall may fail. After the completion of the HA synchronization, the user can successfully log in.
- ASA does not accept remote access VPN sessions if the third-party client sends a null user agent.
- Using Fully Qualified Domain Name (FQDN) Access Control Lists (ACLs) for a domain that resolves to multiple, frequently changing IP addresses can impact the resolution of DHCP addresses in a remote access VPN environment. This issue may arise if an external DHCP server is configured and a transactional commit for Network Address Translation (NAT) is enabled.
- Posture assessment using Advanced Endpoint Assessment may result in SSL connection syslog messages and are not associated with a VPN logon or logoff event.
- Local authentication is not possible because ASA does not terminate any EAP methods.

  ASA supports EAP only as a pass-through and requires certificate authentication for VPN clients for EAP authentication of the clients. When you configure EAP as a remote authentication method, ensure that you configure certificate authentication for VPN clients. Errors are displayed even if multiple remote authentication methods such as EAP, PSK, or certificates are configured along with EAP.
- In a dual-stack configuration where IPv4 addresses are assigned by an external DHCPv4 server and IPv6 addresses are sourced from a local pool, if the DHCPv4 server fails to renew the IPv4 address, Secure Client disconnects and the IPv6 address is released.

Configure Remote Access IPsec VPNs

This section describes how to configure remote access VPNs.

Configure an Interface

An ASA has at least two interfaces, referred to here as outside and inside. Typically, the outside interface is connected to the public Internet, while the inside interface is connected to a private network and is protected from public access.

To begin, configure and enable two interfaces on the ASA. Then assign a name, IP address and subnet mask. Optionally, configure its security level, speed and duplex operation on the security appliance.

Procedure

Step 1 Enter interface configuration mode from global configuration mode:

interface {*interface*}

Example:

```
hostname(config)# interface ethernet0
hostname(config-if)#
```

Step 2 Set the IP address and subnet mask for the interface:

ip address *ip_address* [*mask*] [*standby ip_address*]

Example:

```
hostname(config)# interface ethernet0
hostname(config-if)# ip address 10.10.4.200 255.255.0.0
```

Step 3 Specify a name for the interface (maximum of 48 characters). You cannot change this name after you set it.

nameif *name*

Example:

```
hostname(config-if)# nameif outside
hostname(config-if)#
```

Step 4 Enable the interface. By default, interfaces are disabled.

Example:

```
hostname(config-if)# no shutdown
hostname(config-if)#
```

Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface

Procedure

Step 1 Specify the authentication method and the set of parameters to use during IKEv1 negotiation.

Priority uniquely identifies the Internet Key Exchange (IKE) policy and assigns a priority to the policy. Use an integer from 1 to 65,534, with 1 being the highest priority and 65,534 the lowest.

In the steps that follow, we set the priority to 1.

Step 2 Specify the encryption method to use within an IKE policy:

crypto ikev1 policy *priority* **encryption** {**aes-192** | **aes-256** | | }

Example:

Step 3 Specify the hash algorithm for an IKE policy (also called the HMAC variant):

crypto ikev1 policy *priority* **hash** { | **sha**}

Example:

```
hostname(config)# crypto ikev1 policy 1 hash sha
hostname(config)#
```

Step 4 Specify the Diffie-Hellman group for the IKE policy—the crypto protocol that allows the IPsec client and the ASA to establish a shared secret key:

crypto ikev1 policy *priority* **group** {**14** | | | **19** | **20** | 21}

Example:

```
hostname(config)#crypto ikev1 policy 1 group 14
hostname(config)#
```

Step 5 Specify the encryption key lifetime—the number of seconds each security association should exist before expiring:

crypto ikev1 policy *priority* **lifetime** {*seconds*}

The range for a finite lifetime is 120 to 2147483647 seconds. Use 0 seconds for an infinite lifetime.

Example:

```
hostname(config)# crypto ikev1 policy 1 lifetime 43200
hostname(config)#
```

Step 6 Enable ISAKMP on the interface named outside:

crypto ikev1 enable *interface-name*

Example:

```
hostname(config)# crypto ikev1 enable outside
hostname(config)#
```

Step 7 Save the changes to the configuration:

write memory

Configure an Address Pool

The ASA requires a method for assigning IP addresses to users. This section uses address pools as an example.

Procedure

Create an address pool with a range of IP addresses, from which the ASA assigns addresses to the clients.

ip local pool *poolname first-address—last-address* [**mask** *mask*]

The address mask is optional. However, You must supply the mask value when the IP addresses assigned to VPN clients belong to a non-standard network and the data could be routed incorrectly if you use the default mask. A typical example is when the IP local pool contains 10.10.10.0/255.255.255.0 addresses, since this is a Class A network by default. This could cause routing issues when the VPN client needs to access different subnets within the 10 network over different interfaces.

Example:

```
hostname(config)# ip local pool testpool 192.168.0.10-192.168.0.15
hostname(config)#
```

Add a User

Procedure

Create a user and password and privilege level for the user:

username *name* {**nopassword** | **password** *password* [**mschap** | **encrypted** | **nt-encrypted**]} [**privilege** *priv_level*]

Example:

```
Hostname(config)# username testuser password 12345678
```

Create an IKEv1 Transform Set or IKEv2 Proposal

This section shows how to configure a transform set (IKEv1) or proposal (IKEv2), which combines an encryption method and an authentication method.

The following steps show how to create both an IKEv1 and an IKEv2 proposal.

Procedure

Step 1 Configure an IKEv1 transform set that specifies the IPsec IKEv1 encryption and hash algorithms to be used to ensure data integrity.

crypto ipsec ikev1 transform-set *transform-set-name encryption-method* [*authentication*]

Use one of the following values for encryption:

- esp-aes to use AES with a 128-bit key.
- esp-aes-192 to use AES with a 192-bit key.
- esp-aes-256 to use AES with a 256-bit key.
- esp-null to not use encryption.

Use one of the following values for authentication:

- esp-md5-hmac to use the MD5/HMAC-128 as the hash algorithm.
- esp-sha-hmac to use the SHA/HMAC-160 as the hash algorithm.
- esp-none to not use HMAC authentication.

Example:

To Configure an IKEv1 transform set using AES:

```
hostname(config)# crypto ipsec transform set FirstSet esp-aes esp-sha-hmac
```

Step 2 Configure an IKEv2 proposal set that specifies the IPsec IKEv2 protocol, encryption, and integrity algorithms to be used.

esp specifies the Encapsulating Security Payload (ESP) IPsec protocol (currently the only supported protocol for IPsec).

crypto ipsec ikev2 ipsec-proposal *proposal_name*

protocol {**esp**} {**encryption** { | | **aes** | **aes-192** | **aes-256** | } | **integrity** { | **sha-1**}

Use one of the following values for encryption:

- aes to use AES (default) with a 128-bit key encryption for ESP.
- aes-192 to use AES with a 192-bit key encryption for ESP.
- aes-256 to use AES with a 256-bit key encryption for ESP.

Use one of the following values for integrity:

- sha-1 (default) specifies the Secure Hash Algorithm (SHA) SHA-1, defined in the U.S. Federal Information Processing Standard (FIPS), for ESP integrity protection.

To configure an IKEv2 proposal:

```
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure_proposal

hostname(config-ipsec-proposal)# protocol esp encryption aes intergrity sha-1
```

Define a Tunnel Group

A tunnel group is a collection of tunnel connection policies. You configure a tunnel group to identify AAA servers, specify connection parameters, and define a default group policy. The ASA stores tunnel groups internally.

There are two default tunnel groups in the ASA system: DefaultRAGroup, which is the default remote-access tunnel group, and DefaultL2Lgroup, which is the default LAN-to-LAN tunnel group. You can change these groups, but do not delete them. The ASA uses these groups to configure default tunnel parameters for remote access and LAN-to-LAN tunnel groups when there is no specific tunnel group identified during tunnel negotiation.

Procedure

Step 1 Create an IPsec remote access tunnel-group (also called connection profile):

tunnel-group *name* **type** *type*

Example:

```
hostname(config)# tunnel-group testgroup type ipsec-ra
hostname(config)#
```

Step 2 Enter tunnel group general attributes mode where you can enter an authentication method:

tunnel-group *name* **general-attributes**

Example:

```
hostname(config)# tunnel-group testgroup general-attributes
hostname(config-tunnel-general)#
```

Step 3 Specify an address pool to use for the tunnel group:

address-pool [*(interface name)*] *address_pool1* [*...address_pool6*]

Example:

```
hostname(config-general)# address-pool testpool
```

Step 4 Enter tunnel group IPsec attributes mode where you can enter IPsec-specific attributes for IKEv1 connections:

tunnel-group *name* **ipsec-attributes**

Example:

```
hostname(config)# tunnel-group testgroup ipsec-attributes
hostname(config-tunnel-ipsec)#
```

Step 5 (Optional) Configure a pre-shared key (IKEv1 only). The key can be an alphanumeric string from 1-128 characters.

The keys for the adaptive security appliance and the client must be identical. If a Cisco VPN Client with a different preshared key size tries to connect, the client logs an error message indicating it failed to authenticate the peer.

ikev1 pre-shared-key *key*

Example:

```
hostname(config-tunnel-ipsec)# pre-shared-key 44kkaol59636jnfx
```

Create a Dynamic Crypto Map

Dynamic crypto maps define policy templates in which not all the parameters are configured. This lets the ASA receive connections from peers that have unknown IP addresses, such as remote access clients.

Dynamic crypto map entries identify the transform set for the connection. You can also enable reverse routing, which lets the ASA learn routing information for connected clients, and advertise it via RIP or OSPF.

Procedure

Step 1 Create a dynamic crypto map and specifies an IKEv1 transform set or IKEv2 proposal for the map:

- For IKEv1, use this command:

  crypto dynamic-map *dynamic-map-name seq-num* **set ikev1 transform-set** *transform-set-name*

- For IKEv2, use this command:

  crypto dynamic-map *dynamic-map-name seq-num* **set ikev2 ipsec-proposal** *proposal-name*

Example:

```
hostname(config)# crypto dynamic-map dyn1 1 set ikev1 transform-set FirstSet
hostname(config)#
```

```
hostname(config)# crypto dynamic-map dyn1 1 set ikev2 ipsec-proposal secure_proposal
hostname(config)#
```

Step 2 (Optional) Enable Reverse Route Injection for any connection based on this crypto map entry:

crypto dynamic-map *dynamic-map-name dynamic-seq-num* **set reverse-route**

Example:

```
hostname(config)# crypto dynamic-map dyn1 1 set reverse route
hostname(config)#
```

Create a Crypto Map Entry to Use the Dynamic Crypto Map

Create a crypto map entry that lets the ASA use the dynamic crypto map to set the parameters of IPsec security associations.

In the following examples for this command, the name of the crypto map is mymap, the sequence number is 1, and the name of the dynamic crypto map is dyn1, which you created in the Create a Dynamic Crypto Map topic.

Procedure

Step 1 Create a crypto map entry that uses a dynamic crypto map:

crypto map *map-name seq-num* **ipsec-isakmp dynamic** *dynamic-map-name*

Example:

```
hostname(config)# crypto map mymap 1 ipsec-isakmp dynamic dyn1
```

Step 2 Apply the crypto map to the outside interface:

crypto map *map-name* **interface** *interface-name*

Example:

```
hostname(config)# crypto map mymap interface outside
```

Step 3 Save the changes to the configuration:

write memory

Configuring IPSec IKEv2 Remote Access VPN in Multicontext Mode

For more information about configuring Remote Access IPsec VPNs, see the following sections:

- Configure an Interface, on page 4
- Configure an Address Pool, on page 5
- Add a User, on page 6
- Create an IKEv1 Transform Set or IKEv2 Proposal, on page 6
- Define a Tunnel Group, on page 7
- Create a Dynamic Crypto Map, on page 9
- Create a Crypto Map Entry to Use the Dynamic Crypto Map, on page 9

VPN Authentication Using Post-Quantum Pre-Shared Keys

You can configure IKEv2 with a new key, a post-quantum pre-shared key (PPK), along with pre-shared keys (PSKs) to secure the IPsec communication between Secure Client and an ASA from quantum computer attacks. You must configure matching sets of PPKs and PSKs on the client and the ASA for a secure IPsec connection. Secure Client and ASA use the PPK and PSKs to derive encryption and decryption keys for the network traffic.

PPKs are generated cryptographically in the binary format. For ASA and Secure Client configurations, you must convert the binary PPK to a 256 bit 64 character hexadecimal string.

Prerequisites for Using Post-Quantum Pre-Shared Keys for VPN Authentication

- License: ASA must have a Strong Encryption license.
- Supported Versions
  - ASA Version 9.18.1 and later.
  - Secure Client Version 5.1.8.x and later.

- Configure all other parameters on the ASA for the remote access IPsec/IKEv2 VPN connection such as address pool, IKEv2 proposal, and crypto map.
- Generate the binary PPK.
- Convert the binary PPK to a 256-bit 64 character hexadecimal string.
- Configure both the PPK and the two PSKs for Secure Client in the Windows Credential Manager (WCM) of your client machine. See, Configure Post-Quantum Pre-Shared Keys and Pre-Shared Keys on Windows Credential Manager, on page 12.
- Configure PPK attributes in the VPN profile of the Secure Client. See, Configure the VPN Profile for Secure Client with Post-Quantum Pre-Shared Key Attributes, on page 13.
- Ensure that the PPK and the PPK ID values are the same on the ASA and the Secure Client.

Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication

Guideline

- The administrator must ensure the generation, quality, and distribution of the PPKs and PSKs to each client device.

Limitations

- Supports only IKEv2 with PSK and PPK.
- Supports only Windows for Secure Client.
- Client can store credentials only for one ASA in WCM.

Workflow for Using Post-Quantum Pre-Shared Keys for VPN Authentication

Table 1: Workflow for Using Post-Quantum Pre-Shared Keys for VPN Authentication

Step	Action	More Information
1	Generate the binary PPK and convert it to a 256-bit 64 character hexadecimal string.	-
2	Configure the PPK and PSK in the Windows Credential Manager (WCM).	Configure Post-Quantum Pre-Shared Keys and Pre-Shared Keys on Windows Credential Manager, on page 12
3	Configure the Secure Client VPN profile with the PPK parameters.	Configure the VPN Profile for Secure Client with Post-Quantum Pre-Shared Key Attributes, on page 13

Step	Action	More Information
4	Configure the ASA tunnel group.	Configure VPN Authentication on ASA Using Post-Quantum Pre-Shared Keys, on page 13
5	User logs into Secure Client to connect to ASA.	-
6	Secure Client uses the PPK_ID in the VPN profile to retrieve the PPK and two PSKs from WCM.	-
7	Secure Client verifies the PPK and PSK parameters in the WCM with the ASA tunnel group parameters.	-
8	Secure Client establishes a VPN connection with ASA if the PPK and PSKs of Secure Client and ASA match. The VPN connection with ASA fails if the PPK and PSKs do not match.	-

Configure Post-Quantum Pre-Shared Keys and Pre-Shared Keys on Windows Credential Manager

You must configure separate credential entries for a PPK, a local PSK, and a remote PSK.

Before you begin

Ensure that you review Prerequisites for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 10 and Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 11.

Procedure

Step 1 In your Windows client machine, choose **Control Panel > User Accounts > Credential Manager**.

Step 2 Click the **Windows Credentials** tab.

Step 3 Click **Add a Generic Credential**.

Step 4 In the **Internet or network address** field, specify one of the following values:

- For a PPK, specify the value as **AC/PPK/<HostAddress**: A post-quantum pre-shared key. It is stored as 64 hexadecimal characters in WCM, and the client converts it to binary, and then includes the key in the derivation of encryption and decryption keys in IKEv2..
- For a local PSK, specify the value as **AC/PSK_Local/<HostAddress>**: Client's PSK.
- For a remote PSK, specify the value as **AC/PSK_Remote/<HostAddress**: ASA's PSK.

Step 5 In the **User name** field, specify the value as **n/a**, as it is not used by Secure Client.

Step 6 In the **Password** field, specify one of the following values:

- For a PPK, specify a 256-bit 64-character hexadecimal string.
- For a local and remote PSK, specify a string that specifies the tunnel group alias.

Step 7 Click **OK**.

When the client and ASA are properly configured, the client uses the PPK_ID in the VPN profile to retrieve the PPK and two PSKs from WCM. Secure Client uses the above PPK and PSK values, converts the PPK to binary, matches PPK and PSK values with the ASA configuration, and performs the VPN authentication. No other inputs are required for the VPN connection to be established as these three keys are the authentication credentials.

Configure the VPN Profile for Secure Client with Post-Quantum Pre-Shared Key Attributes

The **HostEntry** parameter in the VPN profile has the following new fields for configuring the PPK parameters for the Secure Client:

- **IKEIdentity**—Specify a string to identify the peer ASA. This string must match the tunnel group name in ASA.
- **PPK_ID**—Specify a unique string to identify the PPK. This value must match the PPK ID in ASA.
- **PPK_mandatory**—Specify the value as true if PPK is mandatory for the VPN connection. If you do not configure this value, then the PPK configuration will be optional.

Example

An example for a HostEntry in the VPN profile is given below:

```
<HostEntry>
<HostName> ASAv_PPK</HostName>
<HostAddress>192.168.1.2</HostAddress>
<UserGroup>IPSec_Profile</UserGroup>
<PrimaryProtocol>IPsec
<StandardAuthenticationOnly>true</StandardAuthenticationOnly>
 <IKEIdentity>secure_client_PPK</IKEIdentity>
 <PPK_ID>PPKID_test</PPK_ID>
</PrimaryProtocol>
</HostEntry>
```

Configure VPN Authentication on ASA Using Post-Quantum Pre-Shared Keys

Tunnel groups in ASA identify the group policy for a VPN connection. You can configure the tunnel group policy to enable VPN authentication using PPK and PSK.

Before you begin

Ensure that you review Prerequisites for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 10 and Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 11.

Procedure

Step 1 Configure the IPsec attributes of the tunnel group.

tunnel-group *name* **ipsec-attributes**

Example:

```
hostname(config)# tunnel-group secure_client_PPK ipsec-attributes
hostname(config-tunnel-ipsec)#
```

Step 2 Configure the client's PSK.

ikev2 remote-authentication pre-shared-key *key*

Example:

```
hostname(config-tunnel-ipsec)#ikev2 remote-authentication pre-shared-key *****
```

Step 3 Configure the ASA's PSK.

ikev2 local-authentication pre-shared-key *key*

Example:

```
hostname(config-tunnel-ipsec)#ikev2 local-authentication pre-shared-key *****
```

Step 4 Configure the client's PPK.

ikev2 remote-authentication post-quantum-key *key* **identifier** *id* mandatory

- key: Specify the PPK key.
- ID: Specify the unique string to identify the PPK. This value must match the PPK ID in Secure Client's VPN profile.
- mandatory: Specify if PPK is mandatory for the VPN connection. If you do not specify mandatory, then the PPK configuration will be optional.

Example:

```
hostname(config-tunnel-ipsec)#ikev2 remote-authentication post-quantum-key  *****
 identifier PPKID_test mandatory
```

The following example shows a snippet of the ASA tunnel group configuration using PPK and PSK:

Example

```
tunnel-group secure_client_PPK ipsec-attributes
       ikev2 remote-authentication pre-shared-key *****
       ikev2 local-authentication pre-shared-key *****
       ikev2 remote-authentication post-quantum-key ***** identity PPKID_test mandatory
```

Note the following:

- The tunnel group name must match the IKEIdentity string of the VPN profile.
- The PPK ID in the tunnel group configuration must match the PPK_ID of the VPN profile.

Additional References

- RFC 8784
- Cisco Secure Client (including AnyConnect) Administrator Guide, Release 5

Configure Portal Access Rules

You can configure a global clientless SSL VPN access policy to permit or deny clientless SSL VPN sessions based on the data present in the HTTP header. The ASA evaluates this access policy before authenticating the endpoint. If ASA denies a clientless SSL VPN session according to the access policy, it returns an error code to the endpoint immediately.

Before you begin

Log in to the ASA and enter the global configuration mode.

Procedure

Step 1 Enter the clientless SSL VPN configuration mode using the following command:

webvpn

Step 2 Permit or deny a clientless SSL VPN session based on an HTTP header code or a string in the HTTP header using the following command:

portal-access-rule *priority* {**permit** | **deny** [**code** *code*]} {**any** | **user-agent match** *string*}

Table 2: portal-access-rule Command Keywords and Variables

Parameter	Description
priority	Specifies the priority of the rule.
permit	Permits a clientless SSL VPN connection based on this setting.
deny	Denies a clientless SSL VPN connection based on this setting.
code *code*	Specifies the HTTP message code. The range is from 200 to 599.

Parameter	Description
user-agent match *string*	Specifies a string that identifies the application, operating system, vendor, and or version of the requesting user agent.

- To specify a string, start and end the string with wildcards (*). Without wildcards, the rule may not match any strings or it may match many fewer strings than you expect. For example, *Thunderbird*.
- To specify a string with a space, start and end the string with wildcards (*) and then add quotes (“ ”) to the beginning and end. In the second example, `my agent` is the string.

Example:

```
hostname(config-webvpn)# portal-access-rule 1 deny code 403 user-agent match *Thunderbird*
hostname(config-webvpn)# portal-access-rule 1 deny code 403 user-agent match "*my agent*"
```

Configure VPN Access of Remote Users based on Geolocation

You can manage the remote access VPN connections of users based on their geolocations. By configuring rules such that they allow or deny VPN connections from specific countries or regions, you can meet compliance requirements and enhance security. Connections that don't meet these location-based criteria are blocked before authentication and the details are logged in the ASA device.

You can apply geolocation policies to Web VPN portals too.

Procedure

Step 1 Create a geolocation object group to define locations in global configuration mode:

object-group geolocation {*object-group-name*}

location {*country-name*}

Example:

```
hostname(config)# object-group geolocation South_Asia
    location "Bangladesh"
    location "Bhutan"
    location "India"
    location "Nepal"
    location "Maldives"
    location "Sri Lanka"
```

Step 2 Define a service access policy to allow geolocation-based access control for your remote clients.

service-access {**permit**| **deny**} [**interface** [*interface-name*] {**geolocation** *object-group-name*| **any**}

To permit remote access SSL users from South Asia on the outside interface, run this command:

Example:

```
hostname(config)# service-access permit interface outside ra-ssl-client geolocation South_Asia
```

To deny all remote access sessions from any location, run this command:

Example:

```
hostname(config)# service-access deny ra-ikev2 ra-ssl-client geolocation any
```

What to do next

Monitor and Troubleshoot Geolocation-based Access Control Policies

Verify Service Access Policies

From the ASA device CLI, run the following commands:

- **show running-config service-access**: Displays the user-defined service access policies.

```
firepower#show running-config service-access
   service-access deny geolocation OBJGRP_Asia1
   service-access permit interface outside ra-ssl-client geolocation OBJGRP_India
   service-access deny ra-ikev2 geolocation any
```

- **show service-access**: Displays details of the user-defined service access policies.

```
firepower# show service-access
1 outside         : ra-ikev2 ra-ssl-client (permit) hits = 8288
  Last hit time   : 10:58:10.038 IST Tue Jul 16 2024
  object-group    : FMC_INTERNAL_XXY
2 any             : ra-ikev2 ra-ssl-client (deny) hits = 123
  Last hit time   : 11:23:12.032 IST Tue Jul 17 2024
  object-group    : any


firepower# show service-access detail
1 outside         : ra-ikev2 ra-ssl-client (permit) hits = 8288
  Last hit time   : 10:58:10.038 IST Tue Jul 16 2024
  object-group    : FMC_INTERNAL_XXY
  geolocation     : Egypt(818) Jordan(400)
                    Iran (Islamic Republic of)(364)
                    Saudi Arabia(682)
```

- **show geodb**: Displays details of the geolocation table.

  show geodb { **ipv4** | **ipv6** | **counters** | **context** } [**location** *country_name* | **lookup** *ip_address*] [**detail**]

  - **show geodb** { **ipv4** | **ipv6** }: Displays the total number of IPv4 or IPv6 address mappings.

```
firepower# show geodb ipv4
 Geolocation Table - IPv4
 Total number of mappings available: 532507
 Last geolocation data read time:  17:02:13.000 IST Thu Jul 18 2024
 Running geolocation update version: 2024-02-15-019
```

  - **show geodb** { **ipv4** | **ipv6** } **location** *country_name* **detail**: Displays the details of the IPv4 or IPv6 address mappings.

```
firepower# show geodb ipv4 location Antarctica detail
Geolocation Table - IPv4
id=0x00007fff82c284e0, geo_id=10, hits=0
        range_lower=77.70.176.176, range_upper=77.70.176.183
id=0x00007fff82cca360, geo_id:10, hits=0
     range_lower=79.110.169.69, range_upper=79.110.169.69
Total number of mappings available: 28
```

- **show geodb counters**: Displays the details of active, permitted, and denied sessions.

```
firepower# show geodb counters
current   - ongoing sessions
permitted - cumulative permitted sessions
denied    - cumulative denied sessions
Location                current          permitted      denied
Egypt                     0                  0             5
India                     45                 1345          45
```

- **show geodb** { **ipv4** | **ipv6lookup** *ip_address*: Displays the geolocation of a specific IPv4 or IPv6 address.

```
firepower# show geodb ipv4 lookup 223.223.128.24
Geolocation of 223.223.128.24 is "India" (356) with id=0x000015114d0aa330
Matching network range: 223.223.128.0 - 223.223.159.255
```

Troubleshoot Service Access Policies

- Use the **show running-config service-access**, and **show service-access** commands to view details of user-defined service access policies.
- Use the **show geodb** command to view details of the geolocation table.
- Use the **debug geolocation <debug-level>** command to capture debug logs related to geolocation. The debug levels can be 1 (Error), 2 (Warning), 3 and 4 (Info), 5 (Debug), or 255 (Debug all).
- Use the **clear geodb counters** command to clear the geolocation table counters such as the hit counts of the service access policies. However, you cannot clear the actual permitted and denied counters for locations using this command; you can clear these counters only after you reboot the device.

Configuration Examples for Remote Access IPsec VPNs

The following example shows how to configure a remote access IPsec/IKEv1 VPN:

```
hostname(config)# crypto ikev1 policy 10
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)# encryption aes-256
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)# group 2
hostname(config)# crypto ikev1 enable outside
hostname(config)# ip local pool POOL 192.168.0.10-192.168.0.15
hostname(config)# username testuser password 12345678
hostname(config)# crypto ipsec ikev1 transform set AES256-SHA
esp-aes-256 esp-sha-hmac
hostname(config)# tunnel-group RAVPN type remote-access
hostname(config)# tunnel-group RAVPN general-attributes
```

```
hostname(config-general)# address-pool POOL
hostname(config)# tunnel-group RAVPN ipsec-attributes
hostname(config-ipsec)# ikev1 pre-shared-key ravpnkey
hostname(config)# crypto dynamic-map DYNMAP 1 set ikev1
transform-set AES256-SHA
hostname(config)# crypto dynamic-map DYNMAP 1 set reverse-route
hostname(config)# crypto map CMAP 1 ipsec-isakmp dynamic DYNMAP
hostname(config)# crypto map CMAP interface outside
```

The following example shows how to configure a remote access IPsec/IKEv2 VPN:

```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)# group 2
hostname(config-ikev2-policy)# integrity sha512
hostname(config-ikev2-policy)# prf sha512
hostname(config)# crypto ikev2 enable outside
hostname(config)# ip local pool POOL 192.168.0.10-192.168.0.15
hostname(config)# username testuser password 12345678
hostname(config)# crypto ipsec ikev2 ipsec-proposal  AES256-SHA512
hostname(config-ipsec-proposal)# protocol esp encryption  aes-256
hostname(config-ipsec-proposal)# protocol esp integrity sha-512
hostname(config)# tunnel-group RAVPN type remote-access
hostname(config)# tunnel-group RAVPN general-attributes
hostname(config-general)# address-pool POOL
hostname(config)# tunnel-group RAVPN ipsec-attributes
hostname(config-tunnel-ipsec)# ikev2 local-authentication
pre-shared-key localravpnkey
hostname(config-tunnel-ipsec)# ikev2 remote-authentication
pre-shared-key remoteravpnkey
hostname(config)# crypto dynamic-map DYNMAP 1 set ikev2
ipsec-proposal AES256-SHA512
hostname(config)# crypto dynamic-map DYNMAP 1 set reverse-route
hostname(config)# crypto map CMAP 1 ipsec-isakmp dynamic DYNMAP
hostname(config)# crypto map CMAP interface outside
```

Configuration Examples for Standards-Based IPSec IKEv2 Remote Access VPN in Multicontext Mode

The following examples show how to configure ASA for Standards-based remote access IPsec/IKEv2 VPN in multicontext mode. The examples provide information for the System Context and User Context configurations respectively.

For the System Context configuration:

```
class default
  limit-resource All 0
  limit-resource Mac-addresses 65536
  limit-resource ASDM 5
  limit-resource SSH 5
  limit-resource Telnet 5
  limit-resource VPN AnyConnect 4.0%
```

```
hostname(config)#context CTX2
hostname(config-ctx)#member default   ===============> License allotment for contexts using
 class
```

```
hostname(config-ctx)#allocate-interface Ethernet1/1.200
hostname(config-ctx)#allocate-interface Ethernet1/3.100
hostname(config-ctx)#config-url disk0:/CTX2.cfg
```

For the User Context configuration:

```
hostname/CTX2(config)#ip local pool CTX2-pool 1.1.2.1-1.1.2.250 mask 255.255.255.0
hostname/CTX2(config)#aaa-server ISE protocol radius
hostname/CTX2(config)#aaa-server ISE (inside) host 10.10.190.100
hostname/CTX2(config-aaa-server-host)#key *****
hostname/CTX2(config-aaa-server-host)#exit
hostname/CTX2(config)#
```

```
hostname/CTX2(config)#group-policy GroupPolicy_CTX2-IKEv2 internal
hostname/CTX2(config)#group-policy GroupPolicy_CTX2-IKEv2 attributes
hostname/CTX2(config-group-policy)#vpn-tunnel-protocol ikev2
hostname/CTX2(config-group-policy)#exit
hostname/CTX2(config)#
```

```
hostname/CTX2(config)#crypto dynamic-map SYSTEM_DEFAULT_CRYPTO_MAP 65535 set ikev2
ipsec-proposal AES256 AES192 AES 3DES DES
hostname/CTX2(config)#crypto map outside_map 65535 ipsec-isakmp dynamic
SYSTEM_DEFAULT_CRYPTO_MAP
hostname/CTX2(config)#crypto map outside_map interface outside
```

IPSec/IKEv2 Remote Access connections from standard-based clients by default fall on tunnel group DefaultRAGroup:

```
hostname/CTX2(config)#tunnel-group DefaultRAGroup type remote-access
hostname/CTX2(config)#tunnel-group DefaultRAGroup general-attributes
hostname/CTX2(config-tunnel-general)#default-group-policy GroupPolicy_CTX2-IKEv2
hostname/CTX2(config-tunnel-general)#address-pool CTX2-pool
hostname/CTX2(config-tunnel-general)#authentication-server-group ISE
hostname/CTX2(config-tunnel-general)#exit
hostname/CTX2(config)#
```

```
hostname/CTX2(config)#tunnel-group DefaultRAGroup ipsec-attributes
hostname/CTX2(config-tunnel-ipsec)#ikev2 remote-authentication eap query-identity
hostname/CTX2(config-tunnel-ipsec)#ikev2 local-authentication certificate ASDM_TrustPoint0
hostname/CTX2(config-tunnel-ipsec)#exit
hostname/CTX2(config)#
```

Configuration Examples for Secure Client IPSec IKEv2 Remote Access VPN in Multicontext Mode

The following examples show how to configure ASA for Secure Client remote access IPsec/IKEv2 VPN in multicontext mode. The examples provide information for the System Context and User Context configurations respectively.

For the System Context configuration:

```
class default
    limit-resource All 0
    limit-resource Mac-addresses 65536
    limit-resource ASDM 5
```

```
    limit-resource SSH 5
    limit-resource Telnet 5
    limit-resource VPN AnyConnect 4.0%

hostname(config)#context CTX3
hostname(config-ctx)#member default   ===============> License allotment for contexts using
 class
hostname(config-ctx)#allocate-interface Ethernet1/1.200
hostname(config-ctx)#allocate-interface Ethernet1/3.100
hostname(config-ctx)#config-url disk0:/CTX3.cfg
```

Virtual File System creation for each context can have Secure Client files like Image and profile.

```
hostname(config-ctx)#storage-url shared disk0:/shared disk0
```

For the User Context configuration:

```
hostname/CTX3(config)#ip local pool ctx3-pool 1.1.3.1-1.1.3.250 mask 255.255.255.0
hostname/CTX3(config)#webvpn
hostname/CTX3(config-webvpn)#enable outside
hostname/CTX3(config-webvpn)# anyconnect image
disk0:/anyconnect-win-4.6.00010-webdeploy-k9.pkg 1
hostname/CTX3(config-webvpn)#anyconnect profiles IKEv2-ctx1 disk0:/ikev2-ctx1.xml
hostname/CTX3(config-webvpn)#anyconnect enable
hostname/CTX3(config-webvpn)#tunnel-group-list enable


hostname/CTX3(config)#username cisco password *****
hostname/CTX3(config)#ssl trust-point ASDM_TrustPoint0 outside
hostname/CTX3(config)#group-policy GroupPolicy_CTX3-IKEv2 internal
hostname/CTX3(config)#group-policy GroupPolicy_CTX3-IKEv2 attributes


hostname/CTX3(config-group-policy)#vpn-tunnel-protocol ikev2 ssl-client
hostname/CTX3(config-group-policy)#dns-server value 10.3.5.6
hostname/CTX3(config-group-policy)#wins-server none
hostname/CTX3(config-group-policy)#default-domain none
hostname/CTX3(config-group-policy)#webvpn
hostname/CTX3(config-group-webvpn)#anyconnect profiles value IKEv2-ctx1 type user
```

In the example below, to enable Client Services, use the **crypto ikev2 enable outside client-services** command.

The Client Services Server provides HTTPS (SSL) access to allow the Secure Client Downloader to receive software upgrades, profiles, localization and customization files, CSD, SCEP, and other file downloads required by the client. If you select this option, specify the client services port number. If you do not enable the Client Services Server, users will not be able to download any of these files that the Secure Client might need.



Note You can use the same port that you use for SSL VPN running on the same device. Even if you have an SSL VPN configured, you must select this option to enable file downloads over SSL for IPsec-IKEv2 clients.

```
hostname/CTX3(config)#crypto ikev2 enable outside client-services port 443
hostname/CTX3(config)#crypto ikev2 remote-access trustpoint ASDM_TrustPoint0
hostname/CTX3(config)#crypto dynamic-map SYSTEM_DEFAULT_CRYPTO_MAP 65535 set ikev2
```

```
ipsec-proposal AES256 AES192 AES 3DES DES
hostname/CTX3(config)#crypto map outside_map 65535 ipsec-isakmp dynamic
SYSTEM_DEFAULT_CRYPTO_MAP
hostname/CTX3(config)#crypto map outside_map interface outside
```

```
hostname/CTX3(config)#tunnel-group CTX3-IKEv2 type remote-access
hostname/CTX3(config)#tunnel-group CTX3-IKEv2 general-attributes
hostname/CTX3(config-tunnel-general)#default-group-policy GroupPolicy_CTX3-IKEv2
hostname/CTX3(config-tunnel-general)#address-pool ctx3-pool
hostname/CTX3(config)#tunnel-group CTX3-IKEv2 webvpn-attributes
hostname/CTX3(config-tunnel-webvpn)#group-alias CTX3-IKEv2 enable
```

Feature History for Remote Access VPNs

Feature Name	Releases	Feature Information
Remote access VPNs for IPsec IKEv1 and SSL.	7.0	Remote access VPNs allow users to connect to a central site through a secure connection over a TCP/IP network such as the Internet.
Remote access VPNs for IPsec IKEv2.	8.4(1)	Added IPsec IKEv2 support for the Secure Client.
Automatic mobike support for remote access VPNs.	9.8(1)	Added Mobile IKE (mobike) support for IPsec IKEv2 RA VPNs. Mobike is always on. Added `ikev2 mobike-rrc` command to enable return routability checking during mobike communications for IKEv2 RA VPN connections.
Remote access VPNs for IPsec IKEv2 in Multi-Context mode	9.9(2)	Support for configuring ASA to allow Secure Client and third party Standards-based IPSec IKEv2 VPN clients to establish Remote Access VPN sessions to ASA operating in multi-context mode. Added the `ikev2 rsa-sig-hash sha1` command to sign the authentication payload.
RSA with SHA-1 hash algorithm for signing the authentication payload	9.12(1)	Support for signing authentication payload with SHA-1 hash algorithm while using a third party Standards-based IPSec IKEv2 VPN clients to establish Remote Access VPN sessions to ASA.

Feature Name	Releases	Feature Information
Deprecations of IKE/IPsec encryption and integrity/PRF ciphers DH group 14 support for IKEv1	9.13(1)	The following encryption/integrity/PRF ciphers are deprecated and will be removed in the later release - 9.14(1): • 3DES encryption • DES encryption • MD5 integrity Added DH group 14 (default) support for IKEv1. The group 2 and group 5 command options was deprecated and will be removed in the later release- 9.14(1).