

# CLI Book 3: Cisco Secure Firewall ASA VPN CLI Configuration Guide, 9.22

**First Published:** 2024-03-01

**Last Modified:** 2026-09-16

**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

THE SPECIFICATIONS AND INFORMATION REGARDING THE PRODUCTS IN THIS MANUAL ARE SUBJECT TO CHANGE WITHOUT NOTICE. ALL STATEMENTS, INFORMATION, AND RECOMMENDATIONS IN THIS MANUAL ARE BELIEVED TO BE ACCURATE BUT ARE PRESENTED WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. USERS MUST TAKE FULL RESPONSIBILITY FOR THEIR APPLICATION OF ANY PRODUCTS.

THE SOFTWARE LICENSE AND LIMITED WARRANTY FOR THE ACCOMPANYING PRODUCT ARE SET FORTH IN THE INFORMATION PACKET THAT SHIPPED WITH THE PRODUCT AND ARE INCORPORATED HEREIN BY THIS REFERENCE. IF YOU ARE UNABLE TO LOCATE THE SOFTWARE LICENSE OR LIMITED WARRANTY, CONTACT YOUR CISCO REPRESENTATIVE FOR A COPY.

The Cisco implementation of TCP header compression is an adaptation of a program developed by the University of California, Berkeley (UCB) as part of UCB's public domain version of the UNIX operating system. All rights reserved. Copyright © 1981, Regents of the University of California.

NOTWITHSTANDING ANY OTHER WARRANTY HEREIN, ALL DOCUMENT FILES AND SOFTWARE OF THESE SUPPLIERS ARE PROVIDED “AS IS" WITH ALL FAULTS. CISCO AND THE ABOVE-NAMED SUPPLIERS DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION, THOSE OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT OR ARISING FROM A COURSE OF DEALING, USAGE, OR TRADE PRACTICE.

IN NO EVENT SHALL CISCO OR ITS SUPPLIERS BE LIABLE FOR ANY INDIRECT, SPECIAL, CONSEQUENTIAL, OR INCIDENTAL DAMAGES, INCLUDING, WITHOUT LIMITATION, LOST PROFITS OR LOSS OR DAMAGE TO DATA ARISING OUT OF THE USE OR INABILITY TO USE THIS MANUAL, EVEN IF CISCO OR ITS SUPPLIERS HAVE BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Any Internet Protocol (IP) addresses and phone numbers used in this document are not intended to be actual addresses and phone numbers. Any examples, command display output, network topology diagrams, and other figures included in the document are shown for illustrative purposes only. Any use of actual IP addresses or phone numbers in illustrative content is unintentional and coincidental.

All printed copies and duplicate soft copies of this document are considered uncontrolled. See the current online version for the latest version.

Cisco has more than 200 offices worldwide. Addresses and phone numbers are listed on the Cisco website at www.cisco.com/go/offices.

Cisco and the Cisco logo are trademarks or registered trademarks of Cisco and/or its affiliates in the U.S. and other countries. To view a list of Cisco trademarks, go to this URL: https://www.cisco.com/c/en/us/about/legal/trademarks.html. Third-party trademarks mentioned are the property of their respective owners. The use of the word partner does not imply a partnership relationship between Cisco and any other company. (1721R)

© 2026 Cisco Systems, Inc. All rights reserved.



# CONTENTS


PREFACE

**About This Guide** xiii

Document Objectives xiii

Related Documentation xiii

Document Conventions xiii

Communications, Services, and Additional Information xv

CHAPTER 1

**IPsec and ISAKMP** 1

About Tunneling, IPsec, and ISAKMP 1

IPsec Overview 2

ISAKMP and IKE Overview 2

About IKEv2 Multi-Peer Crypto Map 3

Licensing for IPsec VPNs 6

Guidelines for IPsec VPNs 7

Configure ISAKMP 7

Configure IKEv1 and IKEv2 Policies 7

IKE Policy Keywords and Values 9

Enable IKE on the Outside Interface 11

Enable or Disable IKEv1 Aggressive Mode 11

Configure an ID Method for IKEv1 and IKEv2 ISAKMP Peers 12

INVALID_SELECTORS Notification 13

Configure IKEv2 Pre-shared Key in Hex 13

Enable or Disable Sending of IKE Notification 13

Configure IKEv2 Fragmentation Options 13

AAA Authentication With Authorization 15

Enable IPsec over NAT-T 15

Enable IPsec with IKEv1 over TCP 17

Configure Certificate Group Matching for IKEv1 17
Configure IPsec 19
Define Crypto Maps 19
Example of LAN-to-LAN Crypto Maps 23
Set Public Key Infrastructure (PKI) Keys 27
Apply Crypto Maps to Interfaces 28
Use Interface ACLs 28
Change IPsec SA Lifetimes 30
Change VPN Routing 31
Create Static Crypto Maps 32
Create Dynamic Crypto Maps 36
Provide Site-to-Site Redundancy 39
Managing IPsec VPNs 39
Viewing an IPsec Configuration 39
Wait for Active Sessions to Terminate Before Rebooting 40
Alert Peers Before Disconnecting 40
Clear Security Associations 40
Clear Crypto Map Configurations 41

CHAPTER 2 **L2TP over IPsec 43**
About L2TP over IPsec/IKEv1 VPN 43
IPsec Transport and Tunnel Modes 44
Licensing Requirements for L2TP over IPsec 45
Prerequisites for Configuring L2TP over IPsec 45
Guidelines and Limitations 45
Configuring L2TP over Eclipse with CLI 47
Creating IKE Policies to Respond to Windows 7 Proposals 50
Configuration Example for L2TP over IPsec 51
Feature History for L2TP over IPsec 52

CHAPTER 3 **High Availability Options 53**
High Availability Options 53
VPN and Clustering on the Secure Firewall eXtensible Operating System (FXOS) Chassis 53
VPN Load Balancing 54

Failover 54
VPN Load Balancing 54
About VPN Load Balancing 54
VPN Load-Balancing Algorithm 55
VPN Load-Balancing Group Configurations 56
VPN Load Balancing Director Election 56
Frequently Asked Questions About VPN Load Balancing 57
Licensing for VPN Load Balancing 58
Prerequisites for VPN Load Balancing 59
Guidelines and Limitations for VPN Load Balancing 59
Configuring VPN Load Balancing 61
Configure the Public and Private Interfaces for VPN Load Balancing 61
Configure the VPN Load Balancing Group Attributes 62
Configuration Examples for VPN Load Balancing 65
Viewing VPN Load Balancing Information 66
Feature History for VPN Load Balancing 66

CHAPTER 4 **General VPN Parameters 67**
Guidelines and Limitations 67
Configure IPsec to Bypass ACLs 68
Permitting Intra-Interface Traffic (Hairpinning) 68
NAT Considerations for Intra-Interface Traffic 69
Setting Maximum Active IPsec or SSL VPN Sessions 70
Use Client Update to Ensure Acceptable IPsec Client Revision Levels 70
Implement NAT-Assigned IP to Public IP Connection 72
Displaying VPN NAT Policies 73
Configure VPN Session Limits 74
Show License Resource Allocation 74
Show License Resource Usage 75
Limit VPN Sessions 75
Using an Identify Certificate When Negotiating 75
Configure the Pool of Cryptographic Cores 76
Configure Dynamic Split Tunneling 76
Configure the Management VPN Tunnel 77

Viewing Active VPN Sessions 78
Viewing Active Secure Client Sessions by IP Address Type 78
Viewing Active LAN to LAN VPN Sessions by IP Address Type 79
About ISE Policy Enforcement 79
Configure RADIUS Server Groups for ISE Policy Enforcement 80
Example Configurations for ISE Policy Enforcement 83
Troubleshooting Policy Enforcement 84
Configure Advanced SSL Settings 84
Rate Limit for Preauthenticated SSL Connections 89
Persistent IPsec Tunneled Flows 91
Configure Persistent IPsec Tunneled Flows Using CLI 92
Troubleshooting Persistent IPsec Tunneled Flows 92
Is the Persistent IPsec Tunneled Flows Feature Enabled? 92
Locating Orphaned Flows 93
Troubleshooting Using Crypto Archives 94
Using SSL Counters 95
How to Remove Stuck ASP Table Entries 96
Clear WebVPN Configurations from ASA 97

CHAPTER 5 **Connection Profiles, Group Policies, and Users** 99
Overview of Connection Profiles, Group Policies, and Users 99
Connection Profiles 100
General Connection Profile Connection Parameters 101
IPsec Tunnel-Group Connection Parameters 102
Connection Profile Connection Parameters for SSL VPN Sessions 103
Configure Connection Profiles 104
Maximum Connection Profiles 105
Default IPsec Remote Access Connection Profile Configuration 105
IPsec Tunnel-Group General Attributes 106
Configure Remote-Access Connection Profiles 106
Specify a Name and Type for the Remote Access Connection Profile 107
Configure Remote-Access Connection Profile General Attributes 107
Configure Double Authentication 111
Configure Remote-Access Connection Profile IPsec IKEv1 Attributes 113

Configure IPsec Remote-Access Connection Profile PPP Attributes 115
Configure LAN-to-LAN Connection Profiles 117
Default LAN-to-LAN Connection Profile Configuration 117
Specify a Name and Type for a LAN-to-LAN Connection Profile 117
Configure LAN-to-LAN Connection Profile General Attributes 117
Configure LAN-to-LAN IPsec IKEv1 Attributes 118
About Tunnel Groups for Standards-based IKEv2 Clients 120
Standards-based IKEv2 Attribute Support 121
DAP Support 121
Tunnel Group Selection for Remote Access Clients 121
Authentication Support for Standards-based IKEv2 Clients 122
Add Multiple Certificate Authentication 123
Configure the query-identity Option for Retrieval of EAP Identity 124
Configure Microsoft Active Directory Settings for Password Management 126
Use Active Directory to Force the User to Change Password at Next Logon 126
Use Active Directory to Specify Maximum Password Age 127
Use Active Directory to Enforce Minimum Password Length 127
Use Active Directory to Enforce Password Complexity 127
Configure the Connection Profile for RADIUS/SDI Message Support for the Secure Client 128
Configure the Security Appliance to Support RADIUS/SDI Messages 128
Group Policies 130
Modify the Default Group Policy 131
Configure Group Policies 133
Configure an External Group Policy 133
Create an Internal Group Policy 134
Configure General Internal Group Policy Attributes 135
Group Policy Name 135
Configure the Group Policy Banner Message 135
Specify Address Pools for Remote Access Connections 136
Assign an IPv4 Address Pool to an Internal Group Policy 136
Assign an IPv6 Address Pool to an Internal Group Policy 137
Specify the Tunneling Protocol for the Group Policy 138
Specify a VLAN for Remote Access or Apply a Unified Access Control Rule to the Group Policy 139

Specify VPN Access Hours for a Group Policy 141
Specify Simultaneous VPN Logins for a Group Policy 142
Restrict Access to a Specific Connection Profile 143
Specify the Maximum VPN Connection Time in a Group Policy 143
Specify a VPN Session Idle Timeout for a Group Policy 144
Configure WINS and DNS Servers for a Group Policy 146
Set the Split-Tunneling Policy 147
Specify a Network List for Split-Tunneling 148
Configure Domain Attributes for Split Tunneling 150
Configure DHCP Intercept for Windows XP and Split Tunneling 151
Configure Browser Proxy Settings for use with Remote Access Clients 152
Configure Security Attributes for IPsec (IKEv1) Clients 154
Configure IPsec-UDP Attributes for IKEv1 Clients 156
Configure Attributes for VPN Hardware Clients 157
Configure Group Policy Attributes for Secure Client Connections 160
Configure Backup Server Attributes 163
Configure Network Admission Control Parameters 164
Configure VPN Client Firewall Policies 167
Configure Secure Client Firewall Policies 168
Use of a Zone Labs Integrity Server 169
Set the Firewall Client Type to Zone Labs 171
Set the Client Firewall Parameters 172
Configure Client Access Rules 174
Configure User Attributes 175
View the Username Configuration 176
Configure Attributes for Individual Users 176
Set a User Password and Privilege Level 176
Configure User Attributes 177
Configure VPN User Attributes 177
Best Practices for Configuring and Adjusting VPN Filter ACL 184

CHAPTER 6 **IP Addresses for VPNs 185**
Configure an IP Address Assignment Policy 185
Configure IPv4 Address Assignments 186

Configure IPv6 Address Assignments 186
View Address Assignment Methods 187
Configure Local IP Address Pools 187
Configure Local IPv4 Address Pools 188
Configure Local IPv6 Address Pools 189
Configure AAA Addressing 189
Configure DHCP Addressing 190

CHAPTER 7 **Remote Access IPsec VPNs** 193
Overview of Remote Access IPsec VPNs 193
About Mobike and Remote Access VPNs 194
Licensing Requirements for AnyConnect VPN Module of Cisco Secure Client 195
Restrictions for Remote Access IPsec VPN 195
Configure Remote Access IPsec VPNs 195
Configure an Interface 196
Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface 196
Configure an Address Pool 197
Add a User 198
Create an IKEv1 Transform Set or IKEv2 Proposal 198
Define a Tunnel Group 199
Create a Dynamic Crypto Map 201
Create a Crypto Map Entry to Use the Dynamic Crypto Map 201
Configuring IPSec IKEv2 Remote Access VPN in Multicontext Mode 202
VPN Authentication Using Post-Quantum Pre-Shared Keys 202
Prerequisites for Using Post-Quantum Pre-Shared Keys for VPN Authentication 202
Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication 203
Workflow for Using Post-Quantum Pre-Shared Keys for VPN Authentication 203
Configure Post-Quantum Pre-Shared Keys and Pre-Shared Keys on Windows Credential Manager 204
Configure the VPN Profile for Secure Client with Post-Quantum Pre-Shared Key Attributes 205
Configure VPN Authentication on ASA Using Post-Quantum Pre-Shared Keys 205
Configuration Examples for Remote Access IPsec VPNs 207
Configuration Examples for Standards-Based IPSec IKEv2 Remote Access VPN in Multicontext Mode 208
Configuration Examples for Secure Client IPSec IKEv2 Remote Access VPN in Multicontext Mode 209

Feature History for Remote Access VPNs 210

CHAPTER 8 **LAN-to-LAN IPsec VPNs 213**

Summary of the Configuration 213
Guidelines for Site-to-Site VPNs 214
Configure Site-to-Site VPN in Multi-Context Mode 214
Configure Interfaces 215
Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface 216
Configure ISAKMP Policies for IKEv1 Connections 217
Configure ISAKMP Policies for IKEv2 Connections 218
Multiple Key Exchanges for IKEv2 219
Guidelines and Limitations for IKEv2 Multiple Key Exchanges 220
Configure Multiple Key Exchanges for IKEv2 220
Verify IKEv2 Multiple Key Exchanges Configurations 221
Create an IKEv1 Transform Set 222
Create an IKEv2 Proposal 223
Configure an ACL 224
Define a Tunnel Group 225
Create a Crypto Map and Applying It To an Interface 226
Apply Crypto Maps to Interfaces 228
Overview of Dynamic Site-to-Site VPNs 229
Prerequisites for using Dynamic VPNs with Loopback Interfaces 229
Configure Dynamic Site-to-Site VPN Using a Loopback Interface 230
Verify Dynamic Site-to-Site VPN Configurations 232

CHAPTER 9 **AnyConnect VPN Client Connections 235**

About the Secure Client VPN Client 235
Licensing Requirements for Secure Client 236
Configure Secure Client Connections 236
Configure the ASA to Web-Deploy the Client 236
Enable Permanent Client Installation 239
Configure DTLS 239
Prompt Remote Users 240
Enable Secure Client Profile Downloads 241

Enable Secure Client Deferred Upgrade 242
Enable DSCP Preservation 244
Enable Additional Secure Client Features 245
Enable Start Before Logon 245
Translating Languages for Secure Client User Messages 246
Understand Language Translation 246
Create Translation Tables 246
Remove Translation Tables 248
Configuring Advanced Secure Client SSL Features 249
Enable Rekey 249
Configure Dead Peer Detection 250
Enable Keepalive 252
Use Compression 252
Adjust MTU Size 253
Update Secure Client Images 253
Enable IPv6 VPN Access 254
SAML 2.0 255
Guidelines and Limitations for SAML 2.0 257
Configure a SAML 2.0 Identity Provider (IdP) 258
Configure ASA as a SAML 2.0 Service Provider (SP) 261
Configure Default OS Browser for SAML Authentication 262
Configure Certificate and SAML Authentication 262
Example SAML 2.0 and Onelogin 264
Troubleshooting SAML 2.0 265
Monitor Secure Client Connections 265
Log Off AnyConnect VPN Sessions 266
Feature History for Secure Client Connections 267

CHAPTER 10 **Secure Client HostScan** 269
Prerequisites for HostScan/Secure Firewall Posture 269
Licensing for HostScan 269
HostScan Packaging 270
Install or Upgrade HostScan/Secure Firewall Posture 270
Enable or Disable HostScan 271

View the HostScan/Secure Firewall Posture Version Enabled on the ASA 272
Uninstall HostScan/Secure Firewall Posture 272
Assign Secure Client Feature Modules to Group Policies 273
HostScan/Secure Firewall Posture Related Documentation 274

CHAPTER 11 **Virtual Tunnel Interface 275**
About Virtual Tunnel Interfaces 275
Guidelines for Virtual Tunnel Interfaces 276
Create a VTI Tunnel 279
Add an IPsec Proposal (Transform Sets) 280
Add an IPsec Profile 281
Add a VTI Interface 282
Add a Dynamic VTI Interface 285
Feature History for Virtual Tunnel Interface 288

CHAPTER 12 **Configure an External AAA Server for VPN 291**
About External AAA Servers 291
Understanding Policy Enforcement of Authorization Attributes 291
Guidelines For Using External AAA Servers 292
Configure Multiple Certificate Authentication 292
Configure Multiple Certificate Username 293
Configure LDAP Authorization for VPN 294
Define the ASA LDAP Configuration 295
Supported Cisco Attributes for LDAP Authorization 295
URL Types Supported in ACLs 304
Guidelines for Using Cisco-AV Pairs (ACLs) 305
Cisco AV Pair Attribute Syntax 306
Cisco AV Pairs ACL Examples 307
Active Directory/LDAP VPN Remote Access Authorization Examples 307
Policy Enforcement of User-Based Attributes 308
Enforce Static IP Address Assignment for Secure Client Tunnels 309
Enforce Dial-in Allow or Deny Access 311
Enforce Logon Hours and Time-of-Day Rules 313



# About This Guide

The following topics explain how to use this guide.


- Document Objectives, on page xiii
- Related Documentation, on page xiii
- Document Conventions, on page xiii
- Communications, Services, and Additional Information, on page xv


## Document Objectives

The purpose of this guide is to help you configure VPN on the Secure Firewall ASA using the command-line interface. This guide does not cover every feature, but describes only the most common configuration scenarios.

You can also configure and monitor the ASA by using Adaptive Security Device Manager (ASDM), a web-based GUI application. ASDM includes configuration wizards to guide you through some common configuration scenarios, and online help for less common scenarios.

This guide applies to the ASA series. Throughout this guide, the term "ASA" applies generically to supported models, unless specified otherwise.

## Related Documentation

For more information, see *Navigating the Cisco ASA Series Documentation* at http://www.cisco.com/go/asadocs.

## Document Conventions

This document adheres to the following text, display, and alert conventions.

### Text Conventions

| Convention | Indication |
|---|---|
| **boldface** | Commands, keywords, button labels, field names, and user-entered text appear in **boldface**. For menu-based commands, the full path to the command is shown. |

| Convention | Indication |
|---|---|
| *italic* | Variables, for which you supply values, are presented in an *italic* typeface.<br>Italic type is also used for document titles, and for general emphasis. |
| `monospace` | Terminal sessions and information that the system displays appear in `monospace` type. |
| {x \| y \| z} | Required alternative keywords are grouped in braces and separated by vertical bars. |
| [ ] | Elements in square brackets are optional. |
| [x \| y \| z] | Optional alternative keywords are grouped in square brackets and separated by vertical bars. |
| [ ] | Default responses to system prompts are also in square brackets. |
| < > | Non-printing characters such as passwords are in angle brackets. |
| !, # | An exclamation point (!) or a number sign (#) at the beginning of a line of code indicates a comment line. |

### Reader Alerts

This document uses the following for reader alerts:



**Note** Means *reader take note*. Notes contain helpful suggestions or references to material not covered in the manual.



**Tip** Means *the following information will help you solve a problem.*



**Caution** Means *reader be careful*. In this situation, you might do something that could result in equipment damage or loss of data.



**Timesaver** Means *the described action saves time.* You can save time by performing the action described in the paragraph.



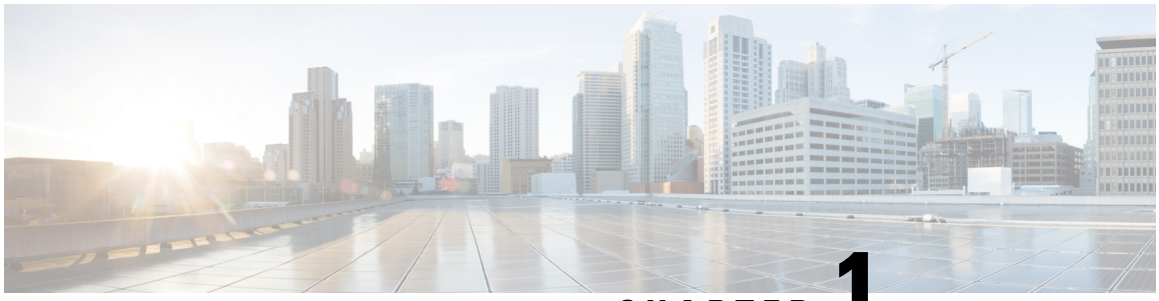
**Warning** **Means *reader be warned.* In this situation, you might perform an action that could result in bodily injury.**

# Communications, Services, and Additional Information

- To receive timely, relevant information from Cisco, sign up at Cisco Profile Manager.
- To get the business results 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.

### Cisco Bug Search Tool

Cisco Bug Search Tool (BST) is a web-based tool that acts as a gateway to the Cisco bug tracking system that maintains a comprehensive list of defects and vulnerabilities in Cisco products and software. BST provides you with detailed defect information about your products and software.



CHAPTER 1

# IPsec and ISAKMP


- About Tunneling, IPsec, and ISAKMP, on page 1
- Licensing for IPsec VPNs, on page 6
- Guidelines for IPsec VPNs, on page 7
- Configure ISAKMP, on page 7
- Configure IPsec, on page 19
- Managing IPsec VPNs, on page 39


## About Tunneling, IPsec, and ISAKMP

This topic describes the Internet Protocol Security (IPsec) and the Internet Security Association and Key Management Protocol (ISAKMP) standards used to build Virtual Private Networks (VPNs).

Tunneling makes it possible to use a public TCP/IP network, such as the Internet, to create secure connections between remote users and a private corporate network. Each secure connection is called a tunnel.

The ASA uses the ISAKMP and IPsec tunneling standards to build and manage tunnels. ISAKMP and IPsec accomplish the following:

- Negotiate tunnel parameters
- Establish tunnels
- Authenticate users and data
- Manage security keys
- Encrypt and decrypt data
- Manage data transfer across the tunnel
- Manage data transfer inbound and outbound as a tunnel endpoint or router

The ASA functions as a bidirectional tunnel endpoint. It can receive plain packets from the private network, encapsulate them, create a tunnel, and send them to the other end of the tunnel where they are unencapsulated and sent to their final destination. It can also receive encapsulated packets from the public network, unencapsulate them, and send them to their final destination on the private network.

# IPsec Overview

The ASA uses IPsec for LAN-to-LAN VPN connections and provides the option of using IPsec for client-to-LAN VPN connections. In IPsec terminology, a *peer* is a remote-access client or another secure gateway. For both connection types, the ASA supports only Cisco peers. Because we adhere to VPN industry standards, ASAs can work with other vendors' peers; however, we do not support them.

During tunnel establishment, the two peers negotiate security associations (SAs) that govern authentication, encryption, encapsulation, and key management. These negotiations involve two phases: first, to establish the tunnel (the IKE SA) and second, to govern traffic within the tunnel (the IPsec SA).

A LAN-to-LAN VPN connects networks in different geographic locations. In IPsec LAN-to-LAN connections, the ASA can function as initiator or responder. In IPsec client-to-LAN connections, the ASA functions only as responder. Initiators propose SAs; responders accept, reject, or make counter-proposals—all in accordance with configured SA parameters. To establish a connection, both entities must agree on the SAs.

### Understanding IPsec Tunnels

IPsec tunnels are sets of SAs that the ASA establishes between peers. The SAs specify the protocols and algorithms to apply to sensitive data and also specify the keying material that the peers use. IPsec SAs control the actual transmission of user traffic. SAs are unidirectional, but are generally established in pairs (inbound and outbound).

The peers negotiate the settings to use for each SA. Each SA consists of the following:

- IKEv1 transform sets or IKEv2 proposals
- Crypto maps
- ACLs
- Tunnel groups
- Prefragmentation policies

# ISAKMP and IKE Overview

ISAKMP is the negotiation protocol that lets two hosts agree on how to build an IPsec security association (SA). It provides a common framework for agreeing on the format of SA attributes. This security association includes negotiating with the peer about the SA and modifying or deleting the SA. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

IKE uses ISAKMP to set up the SA for IPsec to use. IKE creates the cryptographic keys used to authenticate peers.

The ASA supports IKEv1 for connections from the legacy Cisco VPN client, and IKEv2 for the AnyConnect VPN client.

To set the terms of the ISAKMP negotiations, you create an IKE policy, which includes the following:

- The authentication type required of the IKEv1 peer, either RSA signature using certificates or preshared key (PSK).
- 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 determine the strength of the encryption-key-determination algorithm. The ASA uses this algorithm to derive the encryption and hash keys.
- For IKEv2, a separate pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption and so on.
- A limit to the time the ASA uses an encryption key before replacing it.

With IKEv1 policies, you set one value for each parameter. For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This ordering allows you to potentially send a single proposal to convey all the allowed transforms instead of sending each allowed combination as with IKEv1.

The ASA does not support IKEv2 multiple security associations (SAs). The ASA currently accepts inbound IPsec traffic only on the first SA that is found. If IPsec traffic is received on any other SA, it is dropped with reason `vpn-overlap-conflict`. Multiple IPsec SAs can come about from duplicate tunnels between two peers, or from asymmetric tunneling.

### Understanding IKEv1 Transform Sets and IKEv2 Proposals

An IKEv1 transform set or an IKEv2 proposal is a combination of security protocols and algorithms that define how the ASA protects data. During IPsec SA negotiations, the peers must identify a transform set or proposal that is the same at both peers. The ASA then applies the matching transform set or proposal to create an SA that protects data flows in the ACL for that crypto map.

With IKEv1 transform sets, you set one value for each parameter. For IKEv2 proposals, you can configure multiple encryption and authentication types and multiple integrity algorithms for a single proposal. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed combinations instead of the need to send each allowed combination individually as with IKEv1.

The ASA tears down the tunnel if you change the definition of the transform set or proposal used to create its SA. See the Clear Security Associations, on page 40” for further information.

**Note** If you clear or delete the only element in a transform set or proposal, the ASA automatically removes the crypto map references to it.

## About IKEv2 Multi-Peer Crypto Map

Beginning with the 9.14(1) release, ASA IKEv2 supports multi-peer crypto map—when a peer in a tunnel goes down, IKEv2 attempts to establish the tunnel with the next peer in the list. You can configure crypto map with a maximum of 10 peer addresses. This multiple peer support on IKEv2 is useful, especially, when you are migrating from IKEv1 with multi-peer crypto maps.

IKEv2 supports only bi-directional crypto maps. Hence, the multiple peers are also configured on bi-directional crypto maps, and the same is used to accept the request from peers initiating the tunnel.

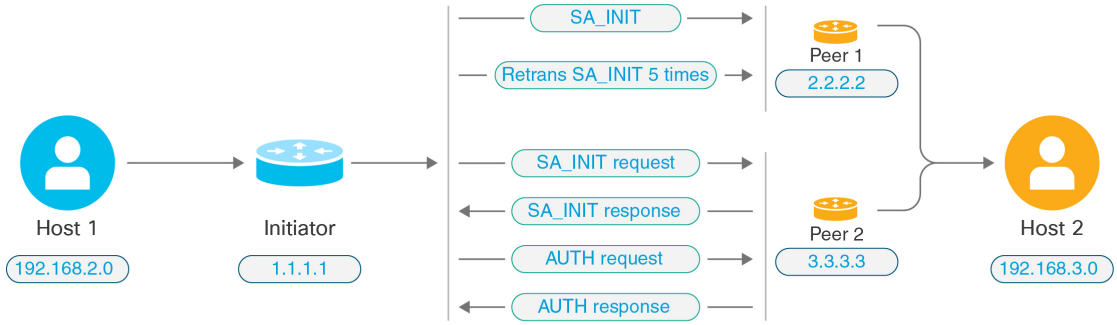
### IKEv2 Initiator Behavior

IKEv2 initiates session with a peer, say Peer1. If Peer1 is unreachable for 5 SA_INIT retransmits, a final retransmit is sent. This activity takes about 2 minutes.

When Peer1 fails, the SA_INIT message is sent to Peer2. If Peer2 is also unreachable, session establishment is initiated with Peer3 after 2 minutes.

After all the peers are exhausted in the peer list of the crypto map, IKEv2 initiates the session again from Peer1 until a SA is established with any of the peers. The following figure depicts this behavior.

***Figure 1: Initiator Process Flow***




**Note** Continuous traffic is required to initiate IKE SA so that each failure attempt would move to the next peer and finally some reachable peer establishes the SA. In cases of disrupted traffic, a manual trigger is needed to initiate the IKE SA with the next peer.

### IKEv2 Responder Behavior

If the responder device of IKE SA is configured with multiple peers in the crypto map, whenever an IKE SA is attempted, the address of the initiator IKE SA is validated with that of the current active peer in the crypto map.

For example, if the current active peer in the crypto map (being used as Responder) is the first peer, then the IKE SA is initiated from Peer1 IP address. Similarly, if the current active peer in the crypto map (being used as Responder) is the second peer, then IKE SA is initiated from Peer2 IP address.

**Note** Peer traversal is not supported on the Responder Side of a IKEv2 multi-peer topology.

### Peer Index Reset Upon Crypto Map Changes

Any change to the crypto map resets the peer index to zero, and the tunnel initiation starts from first peer in the list. Following table provides multiple peer index transition under specific conditions:

*Table 1: Multi-Peer Index Transition before SA*

| Conditions prior to SA | Peer Index Moved<br>Yes/No/Reset |
|---|---|
| Peer not reachable | Yes |
| Phase 1 proposal mismatch | Yes |
| Phase 2 proposal mismatch | Yes |
| DPD ack not received | Yes |
| Traffic selectors mismatch during AUTH phase | Yes |
| Authentication failure | Yes |
| Rekey failure due to peer not reachable | Reset |

*Table 2: Multi-Peer Index Transition after SA*

| Conditions after SA | Peer Index Moved<br>Yes/No/Reset |
|---|---|
| Rekey failure due to proposal mismatch | Reset |
| Traffic selectors mismatch during rekey | Reset |
| Crypto map modification | Reset |
| HA switchover | No |
| Clear crypto IKEv2 SA | Reset |
| Clear ipsec sa | Reset |
| IKEv2 SA timeout | Reset |

## Guidelines for IKEv2 Multi-Peer

### IKEv1 and IKEv2 Protocols

If a crypto map is configured with both the IKE versions and multiple peers, SA attempt is made on each peer with both versions before moving to next peer.

For example, if a crypto map is configured with two peers, say P1 and P2, then the tunnel is initiated to P1 with IKEv2, P1 with IKEv1, P2 with IKEv2, and so on.

### High Availability

A crypto map with multiple peers initiates tunnels to the Responder device that is in HA. It moves to the next Responder device when the first device isn't reachable.

An initiator device initiates tunnels to the Responder device. If the active device goes down, the standby device attempts to establish the tunnel from the Peer1 IP address, irrespective of the crypto map moving to the Peer2 IP address on the active device.

### Centralized Cluster

A crypto map with multiple peers can initiate tunnels to the Responder device that is in a Centralized cluster deployment. If the first device is unreachable, it attempts to move to the next Responder device.

An initiator device initiates tunnels to the Responder device. Every node in the cluster moves to the next Peer2, if Peer1 isn't reachable.

### Distributed Cluster

Distributed clustering isn't supported when an IKEv2 multi-peer crypto map is configured.

### Multiple Context Modes

In multiple context modes, multi-peer behavior is specific to each context.

### Debug Command

If the tunnel establishment fails, enable these commands to further analyse the issue.

- **debug crypto ikev2 platform 255**
- **debug crypto ikev2 protocol 255**
- **debug crypto ike-common 255**

The following example is that of a debug log that is specific to IKEv2 multi-peer, which displays the transition of peers.

```
Sep 13 10:08:58 [IKE COMMON DEBUG]Failed to initiate ikev2 SA with peer 192.168.2.2,
initiate to next peer 192.168.2.3 configured in the multiple peer list of the crypto map.
```

# Licensing for IPsec VPNs

**Note** This feature is not available on No Payload Encryption models.

IPsec remote access VPN using IKEv2 requires an AnyConnect Plus or Apex license, available separately. IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2 uses the Other VPN license that comes with the Essentials license. See Cisco ASA Series Feature Licenses for maximum values per model.

# Guidelines for IPsec VPNs

### Context Mode Guidelines

Supported in single or multiple context mode. Anyconnect Apex license is required for remote-access VPN in multi-context mode. Although ASA does not specifically recognize an AnyConnect Apex license, it enforces licenses characteristics of an Apex license such as AnyConnect Premium licensed to the platform limit, Secure Client for mobile, Secure Client for Cisco VPN phone, and advanced endpoint assessment.

### Firewall Mode Guidelines

Supported in routed firewall mode only. Does not support transparent firewall mode.

### Failover Guidelines

- IPsec VPN sessions are replicated in Active/Standby failover configurations only.
- When a failover occurs, the ESP sequence number increases by 25 million to prevent a false anti-replay.

### Additional Guidelines

When you configure IKE, the system automatically reserves the RADIUS UDP ports 1645 and 1646. This reservation is noted in syslog 713903, where the port numbers are shown as 27910 and 28166. This reservation ensures that the ports do not get used for PAT translations.

# Configure ISAKMP

## Configure IKEv1 and IKEv2 Policies

IKEv1 and IKEv2 each support a maximum of 20 IKE policies, each with a different set of values. Assign a unique priority to each policy that you create. The lower the priority number, the higher the priority.

When IKE negotiations begin, the peer that initiates the negotiation sends all of its policies to the remote peer, and the remote peer tries to find a match. The remote peer checks all of the peer's policies against each of its configured policies in priority order (highest priority first) until it discovers a match.

A match exists when both policies from the two peers contain the same encryption, hash, authentication, and Diffie-Hellman parameter values. For IKEv1, the remote peer policy must also specify a lifetime less than or equal to the lifetime in the policy the initiator sent. If the lifetimes are not identical, the ASA uses the shorter lifetime. For IKEv2 the lifetime is not negotiated but managed locally between each peer, making it possible to configure lifetime independently on each peer. If no acceptable match exists, IKE refuses negotiation and the SA is not established.

There is an implicit trade-off between security and performance when you choose a specific value for each parameter. The level of security the default values provide is adequate for the security requirements of most organizations. If you are interoperating with a peer that supports only one of the values for a parameter, your choice is limited to that value.

You must include the priority in each of the ISAKMP commands. The priority number uniquely identifies the policy and determines the priority of the policy in IKE negotiations.

### Procedure

**Step 1** To create an IKE policy, enter the **crypto** *ikev1* | *ikev2* policy command from global configuration mode in either single or multiple context mode. The prompt displays IKE policy configuration mode.

**Example:**

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

**Note**
New ASA configurations do not have a default IKEv1 or IKEv2 policy.

**Step 2** Specify the encryption algorithm. The default is AES-128.

**encryption [aes | aes-192 | aes-256]**

**Example:**

```
hostname(config-ikev1-policy)#
encryption aes
```

**Step 3** Specify the hash algorithm. The default is SHA-1.

**hash[sha**]

**Example:**

```
hostname(config-ikev1-policy)#
hash sha
```

**Step 4** Specify the authentication method. The default is preshared keys.

**authentication**[**pre-shared**]**rsa-sig**]

**Example:**

```
hostname(config-ikev1-policy)# authentication rsa-sig
```

**Step 5** Specify the Diffie-Hellman group identifier. The default is Group 14.

**group [14]**

**Example:**

```
hostname(config-ikev1-policy)#
group 14
```

**Step 6** Specify the SA lifetime. The default is 86400 seconds (24 hours).

**lifetime** *seconds*

**Example:**

This examples sets a lifetime of 4 hours (14400 seconds):

```
hostname(config-ikev1-policy)# lifetime 14400
```

**Step 7** Specify additional settings using the IKEv1 and IKEv2 policy keywords and their values provided in IKE Policy Keywords and Values, on page 9. If you do not specify a value for a given policy parameter, the default value applies.

## IKE Policy Keywords and Values

| | Keyword | Meaning | Description |
|---|---|---|---|
| **authentication** | **rsa-sig** | A digital certificate with keys generated by the RSA signatures algorithm | Specifies the authentication method the ASA uses to establish the identity of each IPsec peer. |
| | **pre-share** (default) | Preshared keys | Preshared keys do not scale well with a growing network but are easier to set up in a small network. |
| **encryption** | **aes**(default) | AES with a 128-bit key | Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers.<br>The default is 128 -bit key. |
| **hash** | **sha** (default) | SHA-1 (HMAC variant) | Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from and that it has not been modified in transit. |
| **group** | **14** (default) | Group 14 (2048-bit) | Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other.<br>The lower the Diffie-Hellman group number, the less CPU time it requires to execute. The higher the Diffie-Hellman group number, the greater the security.<br>The default group is DH Group 14 |
| **lifetime** | **integer value**<br>(86400 = default) | 120 to 2147483647 seconds | Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the ASA sets up future IPsec SAs more quickly. |

| | Keyword | Meaning | Description |
|---|---|---|---|
| **integrity** | sha (default) | SHA-1 (HMAC variant) | Specifies the hash algorithm used to ensure data integrity. It ensures that a packet comes from where it says it comes from and that it has not been modified in transit. |
| | **sha256** | SHA 2, 256-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 256-bit digest. |

| | Keyword | Meaning | Description |
|---|---|---|---|
| | **sha384** | SHA 2, 384-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 384-bit digest. |
| | **sha512** | SHA 2, 512-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 512-bit digest. |
| | **null** | | When AES-GCM is specified as the encryption algorithm, an administrator can choose null as the IKEv2 integrity algorithm. |
| **encryption** | aes (default) | AES | Specifies the symmetric encryption algorithm that protects data transmitted between two IPsec peers.<br>The default is 128-bit AES. |
| | **aes aes-192 aes-256** | | The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits. |
| | **aes-gcm aes-gcm-192 aes-gcm-256 null** | AES-GCM algorithm options to use for IKEv2 encryption | The Advanced Encryption Standard supports key lengths of 128, 192, 256 bits. |
| **policy_index** | | | Accesses the IKEv2 policy sub-mode. |
| **prf** | sha (default) | SHA-1 (HMAC variant) | Specifies the pseudo random function (PRF)—the algorithm used to generate keying material. |
| | **sha256** | SHA 2, 256-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 256-bit digest. |
| | **sha384** | SHA 2, 384-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 384-bit digest. |
| | **sha512** | SHA 2, 512-bit digest | Specifies the Secure Hash Algorithm SHA 2 with the 512-bit digest. |
| **priority** | | | Extends the policy mode to support the additional IPsec V3 features and makes the AES-GCM and ECDH settings part of the Suite B support. |
| **group** | **14 19 20 21 24** | Group 14 (2048-bit) | Specifies the Diffie-Hellman group identifier, which the two IPsec peers use to derive a shared secret without transmitting it to each other.<br>The lower the Diffie-Hellman group number, the less CPU time it requires to execute. The higher the Diffie-Hellman group number, the greater the security.<br>The default is (DH) Group 14 |

| | Keyword | Meaning | Description |
|---|---|---|---|
| **lifetime** | **integer value**<br>(86400 = default) | 120 to 2147483647 seconds | Specifies the SA lifetime. The default is 86,400 seconds or 24 hours. As a general rule, a shorter lifetime provides more secure ISAKMP negotiations (up to a point). However, with shorter lifetimes, the ASA sets up future IPsec SAs more quickly. |

# Enable IKE on the Outside Interface

You must enable IKE on the interface that terminates the VPN tunnel. Typically this is the outside, or public interface. To enable IKEv1 or IKEv2, use the crypto [**ikev1** | **ikev2**] **enable** *interface-name* command from global configuration mode in either single or multiple context mode.

For example:

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

# Enable or Disable IKEv1 Aggressive Mode

Phase 1 IKEv1 negotiations can use either main mode or aggressive mode. Both provide the same services, but aggressive mode requires only two exchanges between the peers totaling three messages, rather than three exchanges totaling six messages. Aggressive mode is faster, but does not provide identity protection for the communicating parties. Therefore, the peers must exchange identification information before establishing a secure SA. Aggressive mode is enabled by default.

**Note** Disabling aggressive mode prevents Cisco VPN clients from using preshared key authentication to establish tunnels to the ASA. However, they may use certificate-based authentication (that is, ASA or RSA) to establish tunnels.

To enable aggressive mode for phase 1 IKEv1 negotiations, enter the following command in either single or multiple context mode:

```
hostname(config)# crypto map <map-name> seq-num set ikev1 phase1-mode aggressive <group-name>
```

To disable aggressive mode, enter the following command in either single or multiple context mode:

```
hostname(config)# crypto ikev1 am-disable
```

If you have disabled aggressive mode, and want to revert back to it, use the no form of the command. For example:

```
hostname(config)#  no crypto ikev1 am-disable
```

# Configure an ID Method for IKEv1 and IKEv2 ISAKMP Peers

During IKEv1 or IKEv2 ISAKMP Phase I negotiations, the peers must identify themselves to each other. You can choose the identification method from these options.

| | |
|---|---|
| **Address** | Uses the IP addresses of the hosts exchanging ISAKMP identity information. |
| **Automatic**<br>(default) | Determines ISAKMP negotiation by connection type:<br>• IP address for preshared key.<br>• Cert Distinguished Name for certificate authentication. |
| **Hostname** | Uses the fully qualified domain name of the hosts exchanging ISAKMP identity information (default). This name comprises the hostname and the domain name. |
| **Key ID**<br>*key_id_string* | Specifies the string used by the remote peer to look up the preshared key. |

The ASA uses the Phase I IKE ID globally for all IKEv1 and IKEv2 peers. The default ISAKMP identity is **crypto isakmp identity auto**.

To change the peer identification method, enter the following command in either single or multiple context mode:

**crypto isakmp identity** {*address* | *hostname* | **key-id** *id-string* | **auto**}

For example, the following command sets the peer identification method to hostname:

```
hostname(config)# crypto isakmp identity hostname
```

You can define a unique ISAKMP ID for each IKEv2 peer. This option applies to both route-based VPNs and policy-based VPNs.

The table describes the fields used in the commands.

| | |
|---|---|
| *address* | Specifies the IP address of the interface for the identity |
| **auto** | Identity automatically determined by the connection type: IP address for preshared key and Cert DN for Cert based connections |
| *email-id* | Specifies the email-id for the identity |
| *hostname* | Specifies the hostname of the router for the identity |
| *key-id* | Specifies the key-id for the identity |

For a policy based VPN, use this command in global configuration mode to modify the ISAKMP ID for specific peer.

**crypto map** *map-name sequence* **set ikev2 local-identity** {*address* | **auto** | *email-id* | *hostname* | *key-id*}

For a route-based VPN, use this command under **crypto ipsec profile** configuration mode to modify the ISAKMP ID for a specific peer.

**crypto ipsec profile** *profile name* **set ikev2 local-identity** {*address* | **auto** | *email-id* | *hostname* | *key-id*}

# INVALID_SELECTORS Notification

If an IPsec system receives an inbound packet on an SA and the packet's header fields are not consistent with the selectors for the SA, it MUST discard the packet. The audit log entry for this event includes the current date/time, SPI, IPsec protocol(s), source and destination of the packet, any other vector values of the packet that are available, and the selector values from the relevant SA entry. The system generates and sends an IKE notification of INVALID_SELECTORS to the sender (IPsec peer), indicating that the received packet was discarded because of failure to pass selector checks.

The ASA already implements the logging of this event in CTM using the existing syslog shown below:

```
%ASA-4-751027: IKEv2 Received INVALID_SELECTORS Notification from peer: <peer IP>. Peer
received a packet (SPI=<spi>) from <local_IP>. The decapsulated inner packet didn't match
 the negotiated policy in the SA. Packet destination <pkt_daddr>, port <pkt_dest_port>,
source <pkt_saddr>, port <pkt_src_port>, protocol <pkt_prot>
```

An administrator can now enable or disable sending an IKEv2 notification to the peer when an inbound packet is received on an SA that does not match the traffic selectors for that SA. If enabled, the IKEv2 notification messages are rate limited to one notification message per SA every five seconds. The IKEv2 notification is sent in an IKEv2 informational exchange to the peer.

## Configure IKEv2 Pre-shared Key in Hex

You can configure the IKEv2 pre-shared keys in Hex by adding the keyword *hex* to both the local and remote pre-shared key commands.

```
ikev2 local-authentication pre-shared-key [ 0 | 8 | hex ] <string>
ikev2 remote-authentication pre-shared-key [ 0 | 8 | hex ] <string>
```

# Enable or Disable Sending of IKE Notification

An administrator can enable or disable sending an IKE notification to the peer when an inbound packet is received on an IKEv2 IPsec VPN connection that does not match the traffic selectors for that connection. Sending this notification is disabled by default. Sending IKE INVALID_SELECTORS Notifications when Authorization of a username from ASDM certificate is enabled or disabled using the following CLI:

[**no**] **crypto ikev2 notify invalid-selectors**

When certificate authentication is performed, the CN from the certificate is the username, and authorization is performed against the LOCAL server. If “service-type” attribute is retrieved, it is processed as described earlier.

# Configure IKEv2 Fragmentation Options

On the ASA, IKEv2 fragmentation can be enabled or disabled, the MTU (Maximum Transmission Unit) used when fragmenting IKEv2 packets can be specified, and a preferred fragmentation method can be configured by the administrator using the following command:

[**no**] **crypto ikev2 fragmentation** [**mtu** <*mtu-size*>] | [**preferred-method** [**ietf** | **cisco**]]

By default, all methods of IKEv2 fragmentation are enabled, the MTU is 576 for IPv4, or 1280 for IPv6, and the preferred method is the IETF standard RFC-7383.

Specify the [**mtu** <*mtu-size*>] with the following considerations:

- The MTU value used should include the IP(IPv4/IPv6) header + UDP header size.
- If not specified by the administrator the default MTU is 576 for IPv4, or 1280 for IPv6.
- Once specified, the same MTU will be used for both IPv4 and IPv6.
- Valid range is 68-1500.



**Note** You must consider the ESP overhead while configuring the MTU. The packet size increases after encryption due to the ESP overhead that is added to the MTU during the encryption. If you get the "packet too big" error, ensure that you check the MTU size and configure a lower MTU.

One of the following supported fragmentation methods can be configured as the preferred fragmentation method for IKEv2 [**preferred-method** [**ietf** | **cisco**]]:

- IETF RFC-7383 standard based IKEv2 fragmentation.
  - This method will be used when both peers specify support and preference during negotiation.
  - Using this method, encryption is done after fragmentation providing individual protection for each IKEv2 Fragment message.
- Cisco proprietary fragmentation.
  - This method will be used if it is the only method provided by a peer, such as the Secure Client, or if both peers specify support and preference during negotiation.
  - Using this method fragmentation is done after encryption. The receiving peer cannot decrypt or authenticate the message until all fragments are received.
  - This method does not interoperate with non-Cisco peers.

The command **show running-config crypto ikev2** will display the current configuration, and **show crypto ikev2 sa detail** displays the MTU enforced if fragmentation was used for the SA.

### Before you begin

- Path MTU Discovery is not supported, the MTU needs to be manually configured to match the needs of the network.
- This configuration is global and will affect future SAs established after the configuration has been applied. Older SAs will not be affected. Same behavior holds true when fragmentation is disabled.
- A maximum of a 100 fragments can be received.

### Examples

- To disable IKEv2 fragmentation:

```
no crypto ikev2 fragmentation
```

- To reinstate the default operation:

```
crypto ikev2 fragmentation
```

or

```
crypto ikev2 fragmentation mtu 576
preferred-method ietf
```

- To change the MTU value to 600:

```
crypto ikev2 fragmentation mtu 600
```

- To restore the default MTU value:

```
 no crypto ikev2 fragmentation mtu 576
```

- To change the preferred method of fragmentation to Cisco:

```
crypto ikev2 fragmentation preferred-method cisco
```

- To restore the preferred fragmentation method to IETF:

```
no crypto ikev2 fragmentation preferred-method cisco
```

or

```
 crypto ikev2 fragmentation preferred-method ietf
```

# AAA Authentication With Authorization

```
aaa authentication http console LOCAL
aaa authorization http console radius
```

AAA authentication is performed against the LOCAL server using the username/password typed in by the user. Additional authorization is performed against the *radius* server using the same username. *service-type* attribute, if retrieved, is processed as described earlier.

# Enable IPsec over NAT-T

NAT-T lets IPsec peers establish a connection through a NAT device. It does this by encapsulating IPsec traffic in UDP datagrams, using port 4500, which provides NAT devices with port information. NAT-T auto-detects any NAT devices and only encapsulates IPsec traffic when necessary.



**Note** Due to a limitation of the Secure Client, you must enable NAT-T for the Secure Client to successfully connect using IKEv2. This requirement applies even if the client is not behind a NAT-T device.

The ASA can simultaneously support standard IPsec, IPsec over TCP, NAT-T, and IPsec over UDP, depending on the client with which it is exchanging data.

The following breakdown shows the connections with each option enabled.

| Options | Enabled Feature | Client Position | Feature Used |
|---|---|---|---|
| Option 1 | If NAT-T is enabled | and client is behind NAT, then | NAT-T is used |
| | | and no NAT exists, then | Native IPsec (ESP) is used |
| Option 2 | If IPsec over UDP is enabled | and client is behind NAT, then | IPsec over UDP is used |
| | | and no NAT exists, then | IPsec over UDP is used |
| Option 3 | If both NAT-T and IPsec over UDP are enabled | and client is behind NAT, then | NAT-T is used |
| | | and no NAT exists, then | IPsec over UDP is used |

**Note** When IPsec over TCP is enabled, it takes precedence over all other connection methods.

When you enable NAT-T, the ASA automatically opens port 4500 on all IPsec-enabled interfaces.

The ASA supports multiple IPsec peers behind a single NAT/PAT device operating in LAN-to-LAN or remote access networks, but not both. In a mixed environment, the remote access tunnels fail the negotiation because all peers appear to be coming from the same public IP address, address of the NAT device. Also, remote access tunnels fail in a mixed environment because they often use the same name as the LAN-to-LAN tunnel group (that is, the IP address of the NAT device). This match can cause negotiation failures among multiple peers in a mixed LAN-to-LAN and remote access network of peers behind the NAT device.

To use NAT-T, perform the following site-to-site steps in either single or multiple context mode:

### Procedure

**Step 1** Enter the following command to enable IPsec over NAT-T globally on the ASA:

**crypto isakmp nat-traversal** *natkeepalive*

The range for the natkeepalive argument is 10 to 3600 seconds. The default is 20 seconds.

**Example:**

Enter the following command to enable NAT-T and set the keepalive value to one hour:

```
hostname(config)# crypto isakmp nat-traversal 3600
```

**Step 2** Select the before-encryption option for the IPsec fragmentation policy by entering this command:

```
hostname(config)# crypto ipsec fragmentation before-encryption
```

This option lets traffic travel across NAT devices that do not support IP fragmentation. It does not impede the operation of NAT devices that do support IP fragmentation.

# Enable IPsec with IKEv1 over TCP

IPsec over TCP encapsulates both the IKEv1 and IPsec protocols within a TCP-like packet and enables secure tunneling through both NAT and PAT devices and firewalls. This feature is disabled by default. IPsec/IKEv1 over TCP enables a Cisco VPN client to operate in an environment in which standard ESP or IKEv1 cannot function or can function only with modification to existing firewall rules.

**Note** This feature does not work with proxy-based firewalls.

IPsec over TCP works with remote access clients. You enable IPsec over TCP on both the ASA and the client to which it connects. On the ASA, it is enabled globally, working on all IKEv1-enabled interfaces. It does not work for LAN-to-LAN connections.

The ASA can simultaneously support standard IPsec, IPsec over TCP, NAT-Traversal, and IPsec over UDP, depending on the client with which it is exchanging data. IPsec over TCP, if enabled, takes precedence over all other connection methods.

You can enable IPsec over TCP for up to 10 ports that you specify. If you enter a well-known port, for example port 80 (HTTP) or port 443 (HTTPS), the system displays a warning that the protocol associated with that port no longer works on the public interface. The consequence is that you can no longer use a browser to manage the ASA through the public interface. To solve this problem, reconfigure the HTTP/HTTPS management to different ports.

The default port is 10000.

You must configure TCP port(s) on the client as well as on the ASA. The client configuration must include at least one of the ports you set for the ASA.

To enable IPsec over TCP for IKEv1 globally on the ASA, perform the following command in either single or multiple context mode:

**crypto ikev1 ipsec-over-tcp [port** *port 1...port0]*

This example enables IPsec over TCP on port 45:

```
hostname(config)# crypto ikev1 ipsec-over-tcp port 45
```

# Configure Certificate Group Matching for IKEv1

Tunnel groups define user connection terms and permissions. Certificate group matching lets you match a user to a tunnel group using either the Subject DN or Issuer DN of the user certificate.

**Note** Certificate group matching applies to IKEv1 and IKEv2 LAN-to-LAN connections only. IKEv2 remote access connections support the pull-down group selection configured in the webvpn-attributes of the tunnel-group and webvpn configuration mode for certificate-group-map, and so on.

To match users to tunnel groups based on these fields of the certificate, you must first create rules that define a matching criteria, and then associate each rule with the desired tunnel group.

To create a certificate map, **use the crypto ca certificate map** command. To define a tunnel group, use the tunnel-group command.

You must also configure a certificate group matching policy, specifying to match the group from the rules, or from the organizational unit (OU) field, or to use a default group for all certificate users. You can use any or all of these methods.

### Procedure

**Step 1** To configure the policy and rules by which certificate-based ISAKMP sessions map to tunnel groups, and to associate the certificate map entries with tunnel groups, enter the tunnel-group-map command in either single or multiple context mode.

**tunnel-group-map enable** {*rules* | *ou* | *ike-id* | *peer ip*}

**tunnel-group-map** [*rule-index*] **enable** *policy*

| | |
|---|---|
| *policy* | Specifies the policy for deriving the tunnel group name from the certificate. Policy can be one of the following:<br>*ike-id*—Indicates that if a tunnel group is not determined based on a rule lookup or taken from the OU, then the certificate-based ISAKMP sessions are mapped to a tunnel group based on the content of the phase1 ISAKMP ID.<br>*ou*—Indicates that if a tunnel-group is not determined based on a rule lookup, then use the value of the OU in the subject distinguished name (DN).<br>*peer-ip*—Indicates that if a tunnel group is not determined based on a rule lookup or taken from the OU or ike-id methods, then use the peer IP address.<br>*rules*—Indicates that the certificate-based ISAKMP sessions are mapped to a tunnel group based on the certificate map associations configured by this command. |
| *rule index* | (Optional) Refers to parameters specified by the **crypto ca certificate map** command. The values are 1 to 65535. |

Be aware of the following:

- You can invoke this command multiple times as long as each invocation is unique and you do not reference a map index more than once.
- Rules cannot be longer than 255 characters.
- You can assign multiple rules to the same group. To do that, you add the rule priority and group first. Then you define as many criteria statements as you need for each group. When multiple rules are assigned to the same group, a match results for the first rule that tests true.
- By creating a single rule, you can require all criteria to match before assigning a user to a specific tunnel group. Requiring all criteria to match is equivalent to a logical AND operation. Alternatively, create one rule for each criterion if you want to require that only one match before assigning a user to a specific tunnel group. Requiring only one criterion to match is equivalent to a logical OR operation.

**Step 2** Specify a default tunnel group to use when the configuration does not specify a tunnel group.

The syntax is **tunnel-group-map** [*rule-index*] **default-group** *tunnel-group-name* where *rule-index* is the priority for the rule, and tunnel-group name must be for a tunnel group that already exists.

### Examples

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the content of the phase1 ISAKMP ID:

```
hostname(config)# tunnel-group-map enable ike-id
```

The following example enables mapping of certificate-based ISAKMP sessions to a tunnel group based on the IP address of the peer:

```
hostname(config)# tunnel-group-map enable peer-ip
```

The following example enables mapping of certificate-based ISAKMP sessions based on the organizational unit (OU) in the subject distinguished name (DN):

```
hostname(config)# tunnel-group-map enable ou
```

The following example enables mapping of certificate-based ISAKMP sessions based on established rules:

```
hostname(config)# tunnel-group-map enable rules
```

# Configure IPsec

This section describes the procedures required to configure the ASA when using IPsec to implement a VPN.

## Define Crypto Maps

*Crypto maps* define the IPsec policy to be negotiated in the IPsec SA. They include the following:

- ACL to identify the packets that the IPsec connection permits and protects.
- Peer identification.
- Local address for the IPsec traffic. (See Apply Crypto Maps to Interfaces, on page 28 for more details.)
- Up to 11 IKEv1 transform sets or IKEv2 proposals, with which to attempt to match the peer security settings.

A *crypto map set* consists of one or more crypto maps that have the same map name. You create a crypto map set when you create its first crypto map. The following site-to-site task creates or adds to a crypto map in either single or multiple context mode:

**crypto map** *map-name seq-num* **match address** *access-list-name*

Use the access-list-name to specify the ACL ID, as a string or integer up to 241 characters in length.



**Tip** Use all capital letters to more easily identify the ACL ID in your configuration.

You can continue to enter this command to add crypto maps to the crypto map set. In the following example, *mymap* is the name of the crypto map set to which you might want to add crypto maps:

**crypto map mymap 10 match address 101**

The *sequence number* (*seq-num*) shown in the syntax above distinguishes one crypto map from another one with the same name. The sequence number assigned to a crypto map also determines its priority among the other crypto maps within a crypto map set. The lower the sequence number, the higher the priority. After you assign a crypto map set to an interface, the ASA evaluates all IP traffic passing through the interface against the crypto maps in the set, beginning with the crypto map with the lowest sequence number.

**[no] crypto map** *map_name map_index* **set pfs [group14 | group15 | group16 | group19 | group20 | group21 ]**

Specifies the ECDH group used for Perfect Forward Secrecy (PFS) for the cryptography map. Prevents you from configuring group14 and group24 options for a cryptography map (when using an IKEv1 policy).

**[no] crypto map** *map_name seq-num* **set reverse-route** [**dynamic**]

Enables Reverse Route Injection (RRI) for any connection based on this crypto map entry. If dynamic is not specified, RRI is done upon configuration and is considered static, remaining in place until the configuration changes or is removed. Furthermore, whenever an RRI route is configured with same destination for which a static route already exist, the existing static route is discarded and the RRI route is installed. The ASA automatically adds static routes to the routing table and announces these routes to its private network or border routers using OSPF. Do not enable RRI if you specify any source/destination (0.0.0.0/0.0.0.0) as the protected network, because this will impact traffic that uses your default route.



**Note** Use a specific network for RRI because it will not work if you choose the protected network as "any".

If dynamic is specified, routes are created upon the successful establishment of IPsec security associations (SA's) and deleted after the IPsec SA's are deleted.

You cannot configure a dynamic crypto map with the same name as a static crypto map and vice versa, even if one of the crypto maps is not actually in use.



**Note** Dynamic RRI applies to IKEv2 based static crypto maps only.

**[no] crypto map** *name priority* **set validate-icmp-errors**

OR

[**no**]**crypto dynamic-map** *name priority* **set validate-icmp-errors**

Specifies whether incoming ICMP error messages are validated for the cryptography or dynamic cryptography map.

**[no] crypto map <name> <priority> set df-bit [clear-df | copy-df | set-df}**

OR

**[no] crypto map dynamic-map <name> <priority> set df-bit [clear-df | copy-df | set-df]**

Configures the existing do not fragment (DF) policy (at a security association level) for the cryptography or dynamic cryptography map.

- *clear-df*—Ignores the DF bit.
- *copy-df*—Maintains the DF bit.
- *set-df*—Sets and uses the DF bit.

**[no] crypto map <name> <priority> set tfc-packets [burst <length | auto] [payload-size <bytes | auto> [timeout <seconds | auto>**

OR

**[no] crypto dynamic-map <name> <priority> set tfc-packets [burst <length | auto] [payload-size <bytes | auto> [timeout <seconds | auto>**

An administrator can enable dummy Traffic Flow Confidentiality (TFC) packets at random lengths and intervals on an IPsec security association. You must have an IKEv2 IPsec proposal set before enabling TFC.



**Note** Enabling Traffic Flow Confidentiality packets prevents VPN idle timeout.

The ACL assigned to a crypto map consists of all of the ACEs that have the same ACL name, as shown in the following command syntax:

**access-list** *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

You create an ACL when you create its first ACE. The following command syntax creates or adds to an ACL:

**access-list** *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

In the following example, the ASA applies the IPsec protections assigned to the crypto map to all traffic flowing from the 10.0.0.0 subnet to the 10.1.1.0 subnet:

**access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0**

The crypto map that matches the packet determines the security settings used in the SA negotiations. If the local ASA initiates the negotiation, it uses the policy specified in the static crypto map to create the offer to send to the specified peer. If the peer initiates the negotiation, the ASA attempts to match the policy to a static crypto map, and if that fails, then it attempts to match any dynamic crypto maps in the crypto map set, to decide whether to accept or reject the peer offer.

For two peers to succeed in establishing an SA, they must have at least one compatible crypto map. To be compatible, a crypto map must meet the following criteria:

- The crypto map must contain compatible crypto ACLs (for example, mirror image ACLs). If the responding peer uses dynamic crypto maps, so the ASA also must contain compatible crypto ACLs as a requirement to apply IPsec.

- Each crypto map identifies the other peer (unless the responding peer uses dynamic crypto maps).
- The crypto maps have at least one transform set or proposal in common.

You can apply only one crypto map set to a single interface. Create more than one crypto map for a particular interface on the ASA if any of the following conditions exist:

- You want specific peers to handle different data flows.
- You want different IPsec security to apply to different types of traffic.

For example, create a crypto map and assign an ACL to identify traffic between two subnets and assign one IKEv1 transform set or IKEv2 proposal. Create another crypto map with a different ACL to identify traffic between another two subnets and apply a transform set or proposal with different VPN parameters.

If you create more than one crypto map for an interface, specify a sequence number (seq-num) for each map entry to determine its priority within the crypto map set.

Each ACE contains a permit or deny statement. The following table explains the special meanings of permit and deny ACEs in ACLs applied to crypto maps.

| Result of Crypto Map Evaluation | Response |
|---|---|
| Match criterion in an ACE containing a permit statement | Halt further evaluation of the packet against the remaining ACEs in the crypto map set, and evaluate the packet security settings against those in the IKEv1 transform sets or IKEv2 proposals assigned to the crypto map. After matching the security settings to those in a transform set or proposal, the ASA applies the associated IPsec settings. Typically for outbound traffic, this means that it decrypts, authenticates, and routes the packet. |
| Match criterion in an ACE containing a deny statement | Interrupt further evaluation of the packet against the remaining ACEs in the crypto map under evaluation, and resume evaluation against the ACEs in the next crypto map, as determined by the next seq-num assigned to it. |
| Fail to match all tested permit ACEs in the crypto map set | Route the packet without encrypting it. |

ACEs containing deny statements filter out outbound traffic that does not require IPsec protection (for example, routing protocol traffic). Therefore, insert initial deny statements to filter outbound traffic that should not be evaluated against permit statements in a crypto ACL.

For an inbound, encrypted packet, the security appliance uses the source address and ESP SPI to determine the decryption parameters. After the security appliance decrypts the packet, it compares the inner header of the decrypted packet to the permit ACEs in the ACL associated with the packet SA. If the inner header fails to match the proxy, the security appliance drops the packet. It the inner header matches the proxy, the security appliance routes the packet.

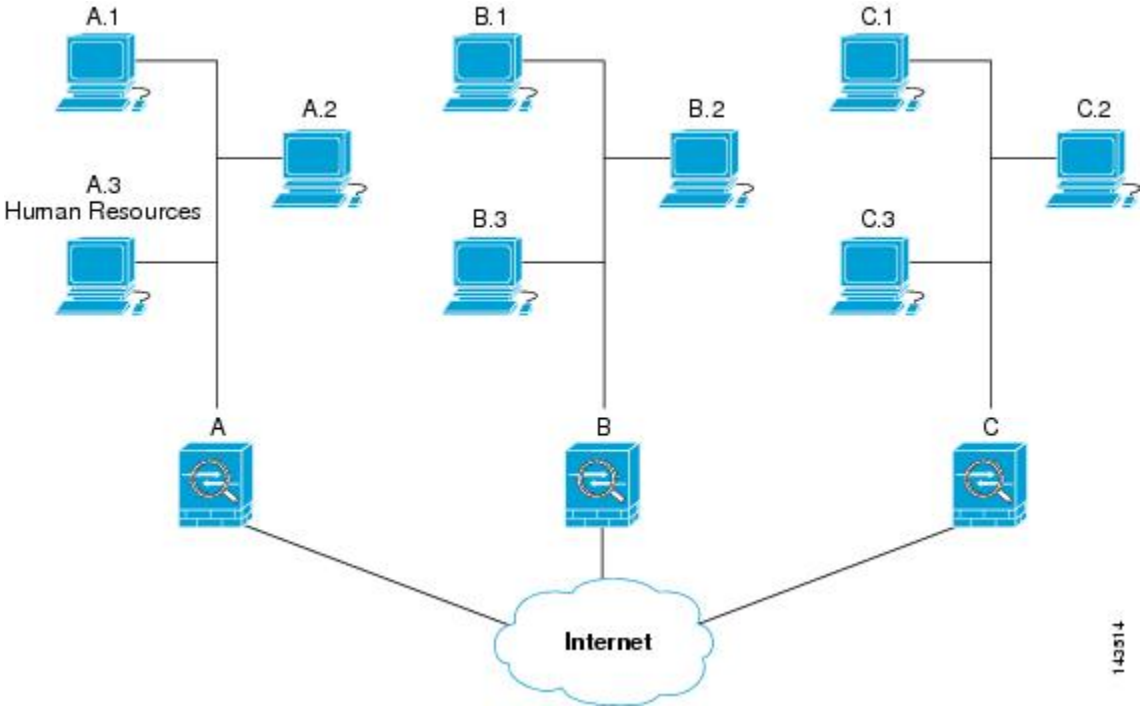
When comparing the inner header of an inbound packet that was not encrypted, the security appliance ignores all deny rules because they would prevent the establishment of a Phase 2 SA.



**Note** To route inbound, unencrypted traffic as clear text, insert deny ACEs before permit ACEs. ASA cannot push more than 28 ACE in split-tunnel access-list.

## Example of LAN-to-LAN Crypto Maps

The objective in configuring Security Appliances A, B, and C in this example LAN-to-LAN network is to permit tunneling of all traffic originating from one of the hosts and destined for one of the other hosts. However, because traffic from Host A.3 contains sensitive data from the Human Resources department, it requires strong encryption and more frequent rekeying than the other traffic. So you will want to assign a special transform set for traffic from Host A.3.




The simple address notation shown in this figure and used in the following explanation is an abstraction. An example with real IP addresses follows the explanation.

To configure Security Appliance A for outbound traffic, you create two crypto maps, one for traffic from Host A.3 and the other for traffic from the other hosts in Network A, as shown in the following example:

```
Crypto Map Seq_No_1
    deny packets from A.3 to B
    deny packets from A.3 to C
    permit packets from A to B
    permit packets from A to C
Crypto Map Seq_No_2
    permit packets from A.3 to B
    permit packets from A.3 to C
```

After creating the ACLs, you assign a transform set to each crypto map to apply the required IPsec to each matching packet.

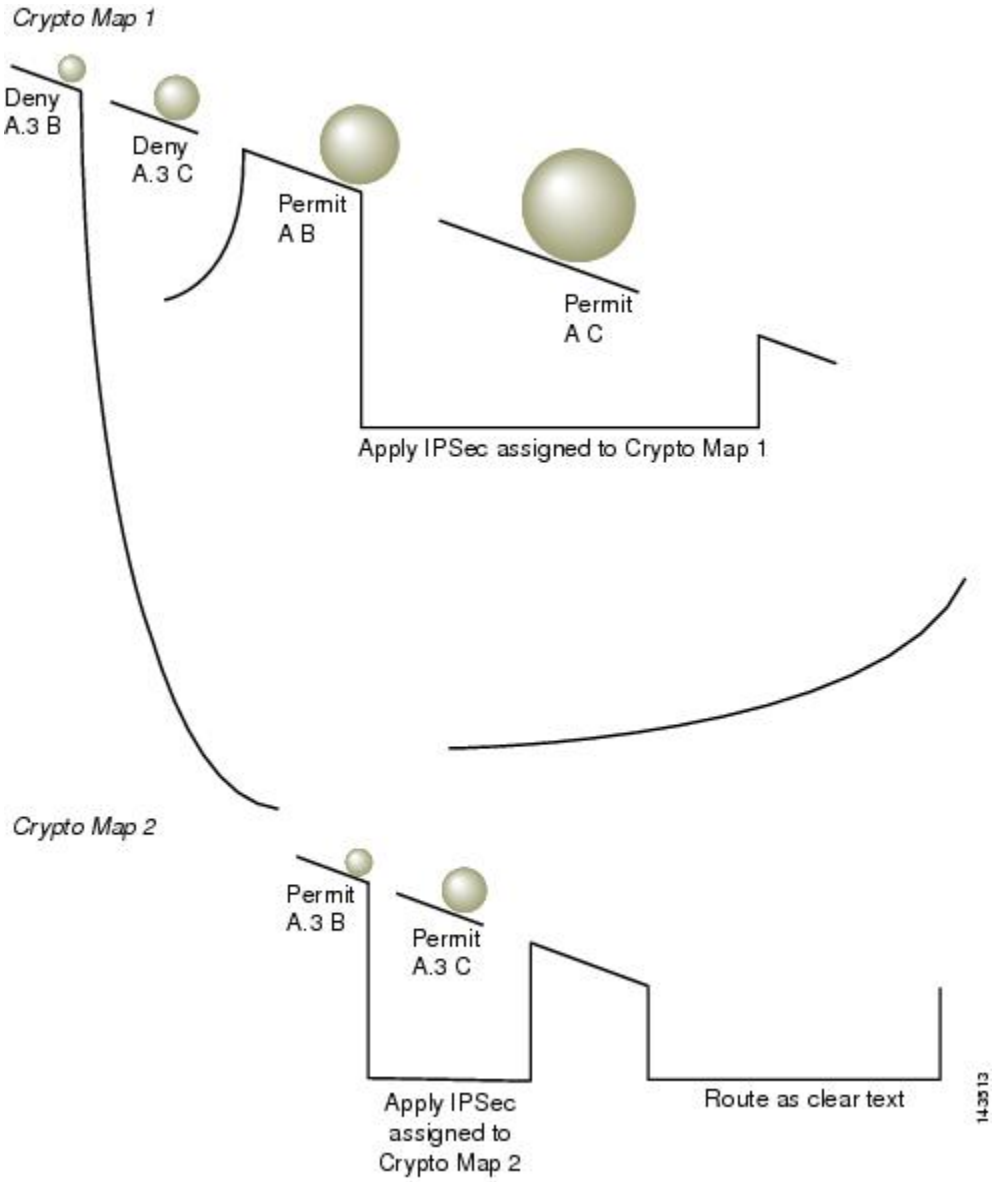
Cascading ACLs involves the insertion of deny ACEs to bypass evaluation against an ACL and resume evaluation against a subsequent ACL in the crypto map set. Because you can associate each crypto map with different IPsec settings, you can use deny ACEs to exclude special traffic from further evaluation in the corresponding crypto map, and match the special traffic to permit statements in another crypto map to provide

or require different security. The sequence number assigned to the crypto ACL determines its position in the evaluation sequence within the crypto map set.

The following illustration shows the cascading ACLs created from the conceptual ACEs in this example. The meaning of each symbol is defined as follows: .

| | |
|---|---|
|  | Crypto map within a crypto map set. |
|  | (Gap in a straight line) Exit from a crypto map when a packet matches an ACE. |
|  | Packet that fits the description of one ACE. Each size ball represents a different packet matching the respective ACE in the figure.<br>The differences in size merely represent differences in the source and destination of each packet. |
|  | Redirection to the next crypto map in the crypto map set. |
|  | Response when a packet either matches an ACE or fails to match all of the permit ACEs in a crypto map set. |

*Figure 2: Cascading ACLs in a Crypto Map Set*



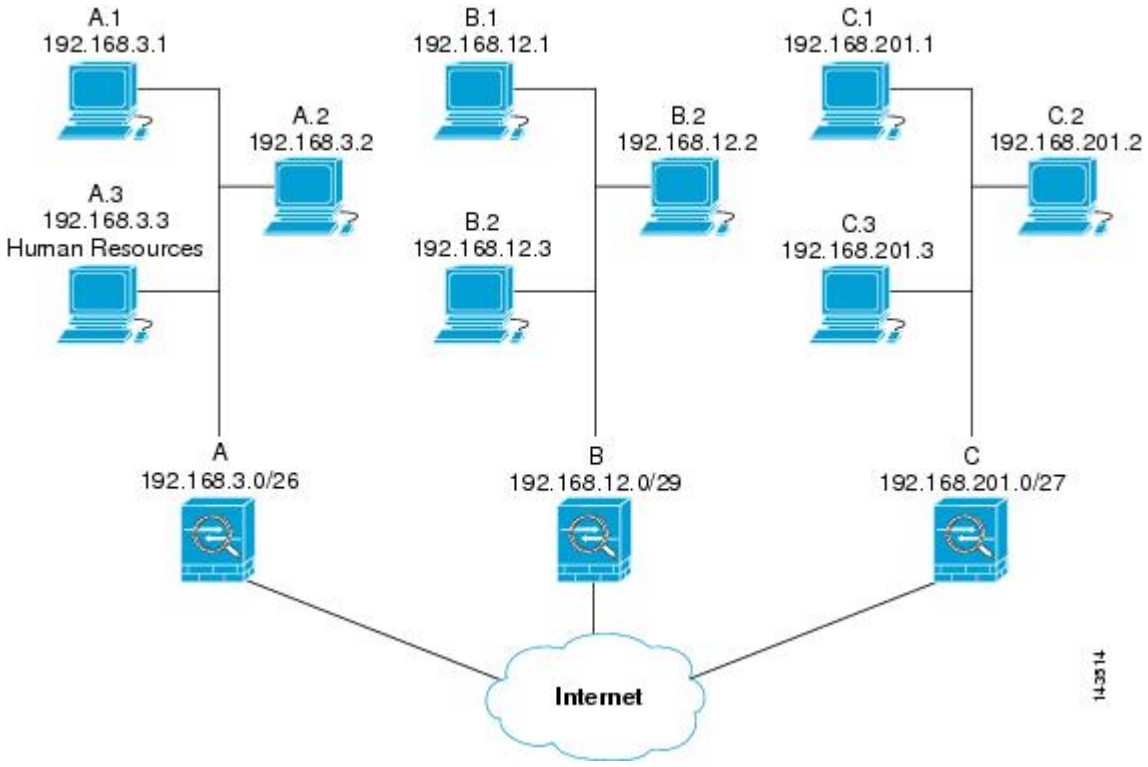

Security Appliance A evaluates a packet originating from Host A.3 until it matches a permit ACE and attempts to assign the IPsec security associated with the crypto map. Whenever the packet matches a deny ACE, the ASA ignores the remaining ACEs in the crypto map and resumes evaluation against the next crypto map, as determined by the sequence number assigned to it. So in the example, if Security Appliance A receives a packet from Host A.3, it matches the packet to a deny ACE in the first crypto map and resumes evaluation of the packet against the next crypto map. When it matches the packet to the permit ACE in that crypto map, it applies the associated IPsec security (strong encryption and frequent rekeying).

To complete the ASA configuration in the example network, we assign mirror crypto maps to ASAs B and C. However, because ASAs ignore deny ACEs when evaluating inbound, encrypted traffic, we can omit the mirror equivalents of the deny A.3 B and deny A.3 C ACEs, and therefore omit the mirror equivalents of Crypto Map 2. So the configuration of cascading ACLs in ASAs B and C is unnecessary.

The following table shows the ACLs assigned to the crypto maps configured for all three ASAs, A, B and C:

| Security Appliance A | | Security Appliance B | | Security Appliance C | |
|---|---|---|---|---|---|
| Crypto Map Sequence No. | ACE Pattern | Crypto Map Sequence No. | ACE Pattern | Crypto Map Sequence No. | ACE Pattern |
| 1 | deny A.3 B | 1 | permit B A | 1 | permit C A |
| | deny A.3 C | | | | |
| | permit A B | | | | |
| | permit A C | | permit B C | | permit C B |
| 2 | permit A.3 B | | | | |
| | permit A.3 C | | | | |

The following illustration maps the conceptual addresses shown previously to real IP addresses.




The real ACEs shown in the following table ensure that all IPsec packets under evaluation within this network receive the proper IPsec settings.

| Security Appliance | Crypto Map Sequence No. | ACE Pattern | Real ACEs |
|---|---|---|---|
| A | 1 | deny A.3 B | deny 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248 |
| | | deny A.3 C | deny 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.224 |
| | | permit A B | permit 192.168.3.0 255.255.255.192 192.168.12.0 255.255.255.248 |
| | | permit A C | permit 192.168.3.0 255.255.255.192 192.168.201.0 255.255.255.22 |
| | 2 | permit A.3 B | permit 192.168.3.3 255.255.255.192 192.168.12.0 255.255.255.248 |
| | | permit A.3 C | permit 192.168.3.3 255.255.255.192 192.168.201.0 255.255.255.22 |
| B | None needed | permit B A | permit 192.168.12.0 255.255.255.248 192.168.3.0 255.255.255.192 |
| | | permit B C | permit 192.168.12.0 255.255.255.248 192.168.201.0 255.255.255.2 |
| C | None needed | permit C A | permit 192.168.201.0 255.255.255.224 192.168.3.0 255.255.255.19 |
| | | permit C B | permit 192.168.201.0 255.255.255.224 192.168.12.0 255.255.255.2 |

You can apply the same reasoning shown in the example network to use cascading ACLs to assign different security settings to different hosts or subnets protected by a ASA.

**Note** By default, the ASA does not support IPsec traffic destined for the same interface from which it enters. Names for this type of traffic include U-turn, hub-and-spoke, and hairpinning. However, you can configure IPsec to support U-turn traffic by inserting an ACE to permit traffic to and from the network. For example, to support U-turn traffic on Security Appliance B, add a conceptual "permit B B" ACE to ACL1. The actual ACE would be as follows: **permit 192.168.12.0 255.255.255.248 192.168.12.0 255.255.255.248**

# Set Public Key Infrastructure (PKI) Keys

You must set public key infrastructure (PKI) in order for an administrator to choose the Suite B ECDSA algorithms when generating or zeroing a keypair:

### Before you begin

If you are configuring a cryptography map to use an RSA or ECDSA trustpoint for authentication, you must first generate the key set. You can then create the trustpoint and reference it in the tunnel group configuration.

### Procedure

**Step 1** Choose the Suite B ECDSA algorithm when generating a keypair:

**crypto key generate [rsa [general-keys | label <name> | modules [512 | 768 | 1024 | 2048 | 4096] | noconfirm | usage-keys] | ecdsa [label <name> | elliptic-curve [256 | 384 | 521] | noconfirm]]**

**Step 2** Choose the Suite B ECDSA algorithm when zeroizing a keypair:

**crypto key zeroize [rsa | ecdsa] [default | label <name> | noconfirm]**

# Apply Crypto Maps to Interfaces

You must assign a crypto map set to each interface through which IPsec traffic flows. The ASA supports IPsec on all interfaces. Assigning the crypto map set to an interface instructs the ASA to evaluate all the traffic against the crypto map set and to use the specified policy during connection or SA negotiation.

Assigning a crypto map to an interface also initializes run-time data structures, such as the SA database and the security policy database. Reassigning a modified crypto map to the interface resynchronizes the run-time data structures with the crypto map configuration. Also, adding new peers through the use of new sequence numbers and reassigning the crypto map does not tear down existing connections.

# Use Interface ACLs

By default, the ASA lets IPsec packets bypass interface ACLs. If you want to apply interface ACLs to IPsec traffic, use the **no** form of the **sysopt connection permit-vpn** command.

The crypto map ACL bound to the outgoing interface either permits or denies IPsec packets through the VPN tunnel. IPsec authenticates and deciphers packets that arrive from an IPsec tunnel, and subjects them to evaluation against the ACL associated with the tunnel.

ACLs define which IP traffic to protect. For example, you can create ACLs to protect all IP traffic between two subnets or two hosts. (These ACLs are similar to ACLs used with the **access-group** command. However, with the **access-group** command, the ACL determines which traffic to forward or block at an interface.)

Before the assignment to crypto maps, the ACLs are not specific to IPsec. Each crypto map references the ACLs and determines the IPsec properties to apply to a packet if it matches a permit in one of the ACLs.

ACLs assigned to IPsec crypto maps have four primary functions:

- Select outbound traffic to be protected by IPsec (permit = protect).
- Trigger an ISAKMP negotiation for data traveling without an established SA.
- Process inbound traffic to filter out and discard traffic that should have been protected by IPsec.
- Determine whether to accept requests for IPsec SAs when processing IKE negotiation from the peer. (Negotiation applies only to **ipsec-isakmp crypto map** entries.) The peer must permit a data flow associated with an **ipsec-isakmp crypto map** command entry to ensure acceptance during negotiation.

**Note** If you delete the only element in an ACL, the ASA also removes the associated crypto map.

If you modify an ACL currently referenced by one or more crypto maps, use the **crypto map interface** command to reinitialize the run-time SA database. See the **crypto map** command for more information.

We recommend that for every crypto ACL specified for a static crypto map that you define at the local peer, you define a "mirror image" crypto ACL at the remote peer. The crypto maps should also support common transforms and refer to the other system as a peer. This ensures correct processing of IPsec by both peers.



**Note** Every static crypto map must define an ACL and an IPsec peer. If either is missing, the crypto map is incomplete and the ASA drops any traffic that it has not already matched to an earlier, complete crypto map. Use the **show conf** command to ensure that every crypto map is complete. To fix an incomplete crypto map, remove the crypto map, add the missing entries, and reapply it.

Crypto ACL does not support duplicate or overlapping entries.

We discourage the use of the **any** keyword to specify source or destination addresses in crypto ACLs because they cause problems. We strongly discourage the **permit any any** command statement because it does the following:

- Protects all outbound traffic, including all protected traffic sent to the peer specified in the corresponding crypto map.
- Requires protection for all inbound traffic.

In this scenario, the ASA silently drops all inbound packets that lack IPsec protection.

Ensure that you define which packets to protect. If you use the **any** keyword in a **permit** statement, preface it with a series of **deny** statements to filter out traffic that would otherwise fall within that **permit** statement that you do not want to protect.



**Note** Decrypted through traffic is permitted from the client despite having an access group on the outside interface, which calls a deny ip any any access-list, while **no sysopt connection permit-vpn** is configured.

Users who want to control access to the protected network via site-to-site or remote access VPN using the **no sysopt permit** command in conjunction with an access control list (ACL) on the outside interface are not successful.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect using SSH to the security appliance. Traffic to hosts on the inside network are blocked correctly by the ACL, but cannot block decrypted through traffic to the inside interface.

The **ssh** and **http** commands are of a higher priority than the ACLs. In other words, to deny SSH, Telnet, or ICMP traffic to the device from the VPN session, use **ssh**, **telnet** and **icmp** commands, which deny the IP local pool should be added.

Regardless of whether the traffic is inbound or outbound, the ASA evaluates traffic against the ACLs assigned to an interface. Follow these steps to assign IPsec to an interface:

### Procedure

**Step 1** Create the ACLs to be used for IPsec.

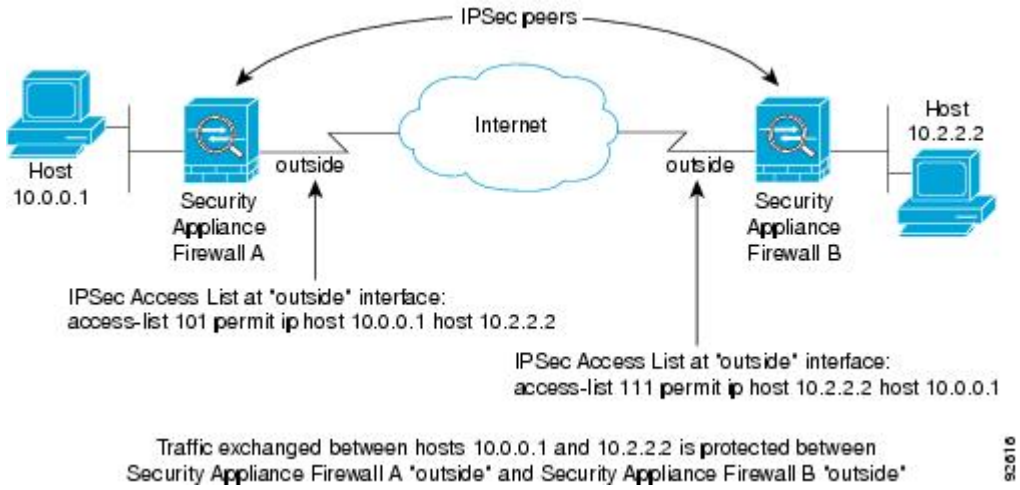
**Step 2** Map the lists to one or more crypto maps, using the same crypto map name.

**Step 3** Map the IKEv1 transform sets or IKEv2 proposals to the crypto maps to apply IPsec to the data flows.

**Step 4** Apply the crypto maps collectively as a crypto map set by assigning the crypto map name they share to the interface.

### Example

In this example, IPsec protection applies to traffic between Host 10.0.0.1 and Host 10.2.2.2 as the data exits the outside interface on ASA A toward Host 10.2.2.2.




ASA A evaluates traffic from Host 10.0.0.1 to Host 10.2.2.2, as follows:

- source = host 10.0.0.1
- dest = host 10.2.2.2

ASA A also evaluates traffic from Host 10.2.2.2 to Host 10.0.0.1, as follows:

- source = host 10.2.2.2
- dest = host 10.0.0.1

The first permit statement that matches the packet under evaluation determines the scope of the IPsec SA.

# Change IPsec SA Lifetimes

You can change the global lifetime values that the ASA uses when negotiating new IPsec SAs. You can override these global lifetime values for a particular crypto map.

IPsec SAs use a derived, shared, secret key. The key is an integral part of the SA; the keys time out together to require the key to refresh. Each SA has two lifetimes: timed and traffic-volume. An SA expires after the respective lifetime and negotiations begin for a new one. The default lifetimes are 28,800 seconds (eight hours) and 4,608,000 kilobytes (10 megabytes per second for one hour).

If you change a global lifetime, the ASA drops the tunnel. It uses the new value in the negotiation of subsequently established SAs.

When a crypto map does not have configured lifetime values and the ASA requests a new SA, it inserts the global lifetime values used in the existing SA into the request sent to the peer. When a peer receives a negotiation request, it uses the smaller of either the lifetime value the peer proposes or the locally configured lifetime value as the lifetime of the new SA.

The peers negotiate a new SA before crossing the lifetime threshold of the existing SA to ensure that a new SA is ready when the existing one expires. The peers negotiate a new SA when about 5 to 15 percent of the lifetime of the existing SA remains.



**Note** We recommend that you configure different security association timers on either side of the site-to-site IKEv2 tunnel to avoid the rekey collision.

# Change VPN Routing

By default, per-packet adjacency lookups are done for the outer ESP packets, lookups are not done for packets sent through the IPsec tunnel.

In some network topologies, when a routing update has altered the inner packet's path, but the local IPsec tunnel is still up, packets through the tunnel may not be routed correctly and fail to reach their destination.

To prevent this, enable per-packet routing lookups for the IPsec inner packets.

### Before you begin

To avoid any performance impact from these lookups, this feature is disabled by default. Enable it only when necessary.

### Procedure

Enable per-packet routing lookups for the IPsec inner packets.

[**no**] [**crypto**] **ipsec inner-routing-lookup**

**Note**
This command, when configured, is only applicable for non-VTI based tunnels.

### Example

```
ciscoasa(config)# crypto ipsec inner-routing-lookup
ciscoasa(config)# show run crypto ipsec
 crypto ipsec ikev2 ipsec-proposal GCM
 protocol esp encryption aes-gcm
 protocol esp integrity null
 crypto ipsec inner-routing-lookup
```

# Create Static Crypto Maps

To create a basic IPsec configuration using a static crypto map, perform the following steps:

### Procedure

**Step 1** To create an ACL to define the traffic to protect, enter the following command:

**access-list** *access-list-name* **{deny | permit} ip** *source source-netmask destination destination-netmask*

The *access-list-name* specifies the ACL ID, as a string or integer up to 241 characters in length. The *destination-netmask* and *source-netmask* specifies an IPv4 network address and subnet mask. In this example, the **permit** keyword causes all traffic that matches the specified conditions to be protected by crypto.

**Example:**

```
hostname(config)# access-list 101 permit ip 10.0.0.0 255.255.255.0 10.1.1.0 255.255.255.0
```

**Step 2** To configure an IKEv1 transform set that defines how to protect the traffic, enter the following command:

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

*Encryption* specifies which encryption method protects IPsec data flows:

- esp-aes—Uses AES with a 128-bit key.
- esp-aes-192—Uses AES with a 192-bit key.
- esp-aes-256—Uses AES with a 256-bit key.\
- esp-null—No encryption.

*Authentication* specifies which encryption method to protect IPsec data flows:

- esp-sha-hmac—Uses the SHA/HMAC-160 as the hash algorithm.
- esp-none—No HMAC authentication.

**Example:**

In this example, myset1 and myset2 and aes_set are the names of the transform sets.

```
hostname(config)# crypto ipsec ikev1 transform-set myset1 esp-aes esp-sha-hmac
hostname(config)#
hostname(config)# crypto ipsec ikev1 transform-set aes_set esp-md5-hmac esp-aes-256
```

**Step 3** To configure an IKEv2 proposal that also defines how to protect the traffic, enter the following command:

**crypto ipsec ikev2 ipsec-proposal** [*proposal tag*]

*proposal tag* is the name of the IKEv2 IPsec proposal, a string from 1 to 64 characters.

Create the proposal and enter the ipsec proposal configuration mode where you can specify multiple encryption and integrity types for the proposal.

**Example:**

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

In this example, secure is the name of the proposal. Enter a protocol and encryption types:

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

**Example:**

This command chooses which AES-GCM or AES-GMAC algorithm to use:

[**no**] **protocol esp encryption** [ **aes**| **aes-192 | aes-256 | aes-gcm**| **aes-gcm-192 | aes-gcm-256**| **null]**

If SHA-2 or null is chosen, you must choose which algorithm to use as an IPsec integrity algorithm. You must choose the null integriy algorithm if AES-GCM/GMAC is configured as the encryption algorithm:

[**no**] **protocol esp integrity** [**sha-1** | **sha-256** | **sha-384** | **sha-512** | **null**]

**Note**
You must choose the null integrity algorithm if AES-GCM/GMAC has been configured as the encryption algorithm. SHA-256 can be used for integrity and PRF to establish IKEv2 tunnels, but it can also be used for ESP integrity protection.

**Step 4** (Optional) An administrator can enable path maximum transfer unit (PMTU) aging and set the interval at which the PMTU value is reset to its original value.

[**no**] **crypto ipsec security-association pmtu-aging** *reset-interval*

**Step 5** To create a crypto map, perform the following site-to-site steps using either single or multiple context mode:

a) Assign an ACL to a crypto map:

**crypto map** *map-name seq-num* **match address** *access-list-name*

A crypto map set is a collection of crypto map entries, each with a different sequence number (*seq-num)* but the same *map name*. Use the *access-list-name* to specify the ACL ID, as a string or integer up to 241 characters in length. In the following example, mymap is the name of the crypto map set. The map set sequence number 10, which is used to rank multiple entries within one crypto map set. The lower the sequence number, the higher the priority.

**Example:**

In this example, the ACL named 101 is assigned to crypto map mymap.

```
crypto map mymap 10 match address 101
```

b) Specify the peer to which the IPsec-protected traffic can be forwarded:

**crypto map** *map_name sequence number***set peer** *ip_address1 [ip_address2] [...]*

**Example:**

```
crypto map mymap 10 set peer 192.168.1.100
```

The ASA sets up an SA with the peer assigned the IP address 192.168.1.100.

**Note**
Beginning with 9.14(1), ASA supports multiple peers in IKEv2 crypto map. You can add a maximum of 10 peers to the list.

c) Specify which IKEv1 transform sets or IKEv2 proposals are allowed for this crypto map. List multiple transform sets or proposals in order of priority (highest priority first). You can specify up to 11 transform sets or proposals in a crypto map using either of these two commands:

**crypto map** *map-name seq-num* **set ikev1 transform-set** *transform-set-name1 [transform-set-name2, …transform-set-name11*]

OR

**crypto map** *map-name seq-num* **set ikev2 ipsec-proposal** *proposal-name1* [*proposal-name2, … proposal-name11*]

*Proposal-name1* and *proposal-name11* specifies one or more names of the IPsec proposals for IKEv2. Each crypto map entry supports up to 11 proposals.

**Example:**

In this example for IKEv1, when traffic matches ACL 101, the SA can use either myset1 (first priority) or myset2 (second priority) depending on which transform set matches the transform set of the peer.

```
crypto map mymap 10 set ikev1 transform-set myset1 myset2
```

d) (Optional) For IKEv2, specify the **mode** for applying ESP encryption and authentication to the tunnel. This determines what part of the original IP packet has ESP applied.

**crypto map** *map-name seq-num* **set ikev2 mode** [**transport | tunnel | transport-require**]

- **Tunnel mode**—(default) Encapsulation mode will be tunnel mode. Tunnel mode applies ESP encryption and authentication to the entire original IP packet (IP header and data), thus hiding the ultimate source and destination addresses.The entire original IP datagram is encrypted, and it becomes the payload in a new IP packet.

  This mode allows a network device, such as a router, to act as an IPsec proxy. That is, the router performs encryption on behalf of the hosts. The source router encrypts packets and forwards them along the IPsec tunnel. The destination router decrypts the original IP datagram and forwards it on to the destination system.

  The major advantage of tunnel mode is that the end systems do not need to be modified to receive the benefits of IPsec. Tunnel mode also protects against traffic analysis; with tunnel mode, an attacker can only determine the tunnel endpoints and not the true source and destination of the tunneled packets, even if they are the same as the tunnel endpoints.

- **Transport mode**— Encapsulation mode will be transport mode with option to fallback on tunnel mode, if peer does not support it. In Transport mode only the IP payload is encrypted, and the original IP headers are left intact.

  This mode has the advantages of adding only a few bytes to each packet and allowing devices on the public network to see the final source and destination of the packet. With transport mode, you can enable special processing (for example, QoS) on the intermediate network based on the information in the IP header. However, the Layer 4 header is encrypted, which limits examination of the packet.

- **Transport Required**— Encapsulation mode will be transport mode only, falling back to tunnel mode is not allowed.

Where **tunnel** encapsulation mode is the default. **transport** encapsulation mode is transport mode with the option to fallback to tunnel mode if the peer does not support it, and **transport-require** encapsulation mode enforces transport mode only.

**Note**
Transport mode is not recommended for Remote Access VPNs.

Examples of negotiation of the encapsulation mode is as follows:

- If the initiator proposes transport mode, and the responder responds with tunnel mode, the initiator will fall back to Tunnel mode.
- If the initiator proposes tunnel mode, and responder responds with transport mode, the responder will fallback to Tunnel mode.
- If the initiator proposes tunnel mode and responder has transport-require mode, then NO PROPOSAL CHOSEN will be sent by the responder.
- Similarly if initiator has transport-require, and responder has tunnel mode, NO PROPOSAL CHOSEN will be sent by the responder.

e) (Optional) Specify an SA lifetime for the crypto map if you want to override the global lifetime.

**crypto map** *map-name seq-num* **set security-association lifetime** {**seconds** *number* | **kilobytes** {*number* | **unlimited**}}

*Map-name* specifies the name of the crypto map set. *Seq-num* specifies the number you assign to the crypto map entry. You can set both lifetimes based on time or on data transmitted. However, the data transmitted lifetime applies to site-to-site VPN only, it does not apply to remote access VPN.

**Example:**

This example shortens the timed lifetime for the crypto map mymap 10 to 2700 seconds (45 minutes). The traffic volume lifetime is not changed.

```
crypto map mymap 10 set security-association lifetime seconds 2700
```

f) (Optional) Specify that IPsec require perfect forward secrecy when requesting new SA for this crypto map, or require PFS in requests received from the peer:

**crypto map** *map_name seq-num* **set pfs** [**group14** | **group15** | **group16** | **group19** | **group20** | **group21**]

**Example:**

This example requires PFS when negotiating a new SA for the crypto map mymap 10. The ASA uses the 2048-bit Diffie-Hellman prime modulus group in the new SA.

```
crypto map mymap 10 set pfs group14
```

g) (Optional) Enable Reverse Route Injection (RRI) for any connection based on this crypto map entry.

**crypto map** *map_name seq-num* **set reverse-route** [**dynamic**]

If dynamic is not specified, RRI is done upon configuration and is considered static, remaining in place until the configuration changes or is removed. The ASA automatically adds static routes to the routing table and announces these routes to its private network or border routers using OSPF. Do not enable RRI if you specify any source/destination (0.0.0.0/0.0.0.0) as the protected network, because this will impact traffic that uses your default route.

If dynamic is specified, routes are created upon the successful establishment of IPsec security associations (SA's) and deleted after the IPsec SA's are deleted.

**Note**
Dynamic RRI applies to IKEv2 based static crypto maps only.

**Example:**

```
crypto map mymap 10 set reverse-route dynamic
```

**Step 6** Apply a crypto map set to an interface for evaluating IPsec traffic:

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

*Map-name* specifies the name of the crypto map set. *Interface-name* specifies the name of the interface on which to enable or disable ISAKMP IKEv1 negotiation.

**Example:**

In this example, the ASA evaluates the traffic going through the outside interface against the crypto map mymap to determine whether it needs to be protected.

```
crypto map mymap interface outside
```

# Create Dynamic Crypto Maps

A dynamic crypto map is a crypto map without all of the parameters configured. It acts as a policy template where the missing parameters are later dynamically learned, as the result of an IPsec negotiation, to match the peer requirements. The ASA applies a dynamic crypto map to let a peer negotiate a tunnel if its IP address is not already identified in a static crypto map. This occurs with the following types of peers:

- Peers with dynamically assigned public IP addresses.

  Both LAN-to-LAN and remote access peers can use DHCP to obtain a public IP address. The ASA uses this address only to initiate the tunnel.

- Peers with dynamically assigned private IP addresses.

  Peers requesting remote access tunnels typically have private IP addresses assigned by the headend. Generally, LAN-to-LAN tunnels have a predetermined set of private networks that are used to configure static maps and therefore used to establish IPsec SAs.

As an administrator configuring static crypto maps, you might not know the IP addresses that are dynamically assigned (via DHCP or some other method), and you might not know the private IP addresses of other clients, regardless of how they were assigned. VPN clients typically do not have static IP addresses; they require a dynamic crypto map to allow IPsec negotiation to occur. For example, the headend assigns the IP address to a Cisco VPN client during IKE negotiation, which the client then uses to negotiate IPsec SAs.

**Note** A dynamic crypto map requires only the **transform-set** parameter.

Dynamic crypto maps can ease IPsec configuration, and we recommend them for use in networks where the peers are not always predetermined. Use dynamic crypto maps for Cisco VPN clients (such as mobile users) and routers that obtain dynamically assigned IP addresses.

**Tip** Use care when using the **any** keyword in **permit** entries in dynamic crypto maps. If the traffic covered by such a **permit** entry could include multicast or broadcast traffic, insert **deny** entries for the appropriate address range into the ACL. Remember to insert **deny** entries for network and subnet broadcast traffic, and for any other traffic that IPsec should not protect.

Dynamic crypto maps work only to negotiate SAs with remote peers that initiate the connection. The ASA cannot use dynamic crypto maps to initiate connections to a remote peer. With a dynamic crypto map, if outbound traffic matches a permit entry in an ACL and the corresponding SA does not yet exist, the ASA drops the traffic.

A crypto map set may include a dynamic crypto map. Dynamic crypto map sets should be the lowest priority crypto maps in the crypto map set (that is, they should have the highest sequence numbers) so that the ASA evaluates other crypto maps first. It examines the dynamic crypto map set only when the other (static) map entries do not match.

Similar to static crypto map sets, a dynamic crypto map set consists of all of the dynamic crypto maps with the same dynamic-map-name. The dynamic-seq-num differentiates the dynamic crypto maps in a set. If you configure a dynamic crypto map, insert a permit ACL to identify the data flow of the IPsec peer for the crypto ACL. Otherwise the ASA accepts any data flow identity the peer proposes.

**Caution** Do not assign module default routes for traffic to be tunneled to a ASA interface configured with a dynamic crypto map set. To identify the traffic that should be tunneled, add the ACLs to the dynamic crypto map. Use care to identify the proper address pools when configuring the ACLs associated with remote access tunnels. Use Reverse Route Injection to install routes only after the tunnel is up.

Create a crypto dynamic map entry using either single or multiple context mode. You can combine static and dynamic map entries within a single crypto map set.

### Procedure

**Step 1** (Optional) Assign an ACL to a dynamic crypto map:

**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **match address** *access-list-name*

This determines which traffic should be protected and not protected. *Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry.

**Example:**

In this example, ACL 101 is assigned to dynamic crypto map dyn1. The map sequence number is 10.

```
crypto dynamic-map dyn1 10 match address 101
```

**Step 2** Specify which IKEv1 transform sets or IKEv2 proposals are allowed for this dynamic crypto map. List multiple transform sets or proposals in order of priority (highest priority first) using the command for IKEv1 transform sets or IKEv2 proposals:

**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set ikev1 transform-set** *transform-set-name1*, [*transform-set-name2*, ...*transform-set-name9*]

**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set ikev2 ipsec-proposal** *proposal-name1* [*proposal-name2*, … *proposal-name11*]

*Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry. The *transform-set-name* is the name of the transform-set being created or modified. The *proposal-name* specifies one or more names of the IPsec proposals for IKEv2.

**Example:**

In this example for IKEv1, when traffic matches ACL 101, the SA can use either myset1 (first priority) or myset2 (second priority), depending on which transform set matches the transform sets of the peer.

```
crypto dynamic-map dyn 10 set ikev1 transform-set myset1 myset2
```

**Step 3** (Optional) Specify the SA lifetime for the crypto dynamic map entry if you want to override the global lifetime value:

**crypto dynamic-map** *dynamic-map-name dynamic-seq-num* **set security-association lifetime** {**seconds** *number* | **kilobytes** {*number* | **unlimited**}}

*Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry. You can set both lifetimes based on time or on data transmitted. However, the data transmitted lifetime applies to site-to-site VPN only, it does not apply to remote access VPN.

**Example:**

This example shortens the timed lifetime for dynamic crypto map dyn1 10 to 2700 seconds (45 minutes). The time volume lifetime is not changed.

```
crypto dynamic-map dyn1 10 set security-association lifetime seconds 2700
```

**Step 4** (Optional) Specify that IPsec ask for PFS when requesting new SAs for this dynamic crypto map, or should demand PFS in requests received from the peer:

**crypto dynamic-map** *dynamic-map-name dynamic-seq-num***set pfs** [**group14** | **group15** | **group16** | **group19** | **group20** | **group21**]

*Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map. *Dynamic-seq-num* specifies the sequence number that corresponds to the dynamic crypto map entry.

**Example:**

```
crypto dynamic-map dyn1 10 set pfs group14
```

**Step 5** Add the dynamic crypto map set into a static crypto map set.

Be sure to set the crypto maps referencing dynamic maps to be the lowest priority entries (highest sequence numbers) in a crypto map set.

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

*Map-name* specifies the name of the crypto map set. *Dynamic-map-name* specifies the name of the crypto map entry that refers to a pre-existing dynamic crypto map.

**Example:**

```
crypto map  mymap 200 ipsec-isakmp dynamic dyn1
```

## Provide Site-to-Site Redundancy

You can define multiple IKEv1 peers by using crypto maps to provide redundancy. This configuration is useful for site-to-site VPNs. This feature is not supported with IKEv2.

If one peer fails, the ASA establishes a tunnel to the next peer associated with the crypto map. It sends data to the peer that it has successfully negotiated with, and that peer becomes the active peer. The active peer is the peer that the ASA keeps trying first for follow-on negotiations until a negotiation fails. At that point the ASA goes on to the next peer. The ASA cycles back to the first peer when all peers associated with the crypto map have failed.

# Managing IPsec VPNs

## Viewing an IPsec Configuration

These are the commands that you can enter in either single or multiple context mode to view information about your IPsec configuration.

***Table 3: Commands to View IPsec Configuration Information***

| | |
|---|---|
| **show running-configuration crypto** | Displays the entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP. |
| **show running-config crypto ipsec** | Displays the complete IPsec configuration. |
| **show running-config crypto isakmp** | Displays the complete ISAKMP configuration. |
| **show running-config crypto map** | Displays the complete crypto map configuration. |
| **show running-config crypto dynamic-map** | Displays the dynamic crypto map configuration. |
| **show all crypto map** | Displays all of the configuration parameters, including those with default values. |
| **show crypto ikev2 sa detail** | Shows the Suite B algorithm support in the Encryption statistics. |
| **show crypto ipsec sa** | Shows the Suite B algorithm support and the ESPv3 IPsec output in either single or multiple context mode. |

| | |
|---|---|
| **show ipsec stats** | Shows information about the IPsec subsystem in either single or multiple context mode. ESPv3 statistics are shown in TFC packets and valid and invalid ICMP errors received. |

# Wait for Active Sessions to Terminate Before Rebooting

You can schedule an ASA reboot to occur only when all active sessions have terminated voluntarily. This feature is disabled by default.

Use the **reload** command to reboot the ASA. If you set the **reload-wait** command, you can use the **reload quick** command to override the **reload-wait** setting. The **reload** and **reload-wait** commands are available in privileged EXEC mode; neither includes the **isakmp** prefix.

### Procedure

To enable waiting for all active sessions to voluntarily terminate before the ASA reboots, perform the following site-to-site task in either single or multiple context mode:

**crypto isakmp reload-wait**

**Example:**

```
hostname(config)# crypto isakmp reload-wait
```

# Alert Peers Before Disconnecting

Remote access or LAN-to-LAN sessions can drop for several reasons, such as an ASA shutdown or reboot, session idle timeout, maximum connection time exceeded, or administrator cut-off.

The ASA can notify qualified peers (in LAN-to-LAN configurations or VPN clients) of sessions that are about to be disconnected. The peer or client receiving the alert decodes the reason and displays it in the event log or in a pop-up pane. This feature is disabled by default.

Qualified clients and peers include the following:

- Security appliances with Alerts enabled
- Cisco VPN clients running Version 4.0 or later software (no configuration required)

To enable disconnect notification to IPsec peers, enter the **crypto isakmp disconnect-notify** command in either single or multiple context mode.

# Clear Security Associations

Certain configuration changes take effect only during the negotiation of subsequent SAs. If you want the new settings to take effect immediately, clear the existing SAs to reestablish them with the changed configuration. If the ASA is actively processing IPsec traffic, clear only the portion of the SA database that the configuration

changes affect. Reserve clearing the full SA database for large-scale changes, or when the ASA is processing a small amount of IPsec traffic.

The following table lists commands you can enter to clear and reinitialize IPsec SAs in either single or multiple context mode.

***Table 4: Commands to Clear and Reinitialize IPsec SAs***

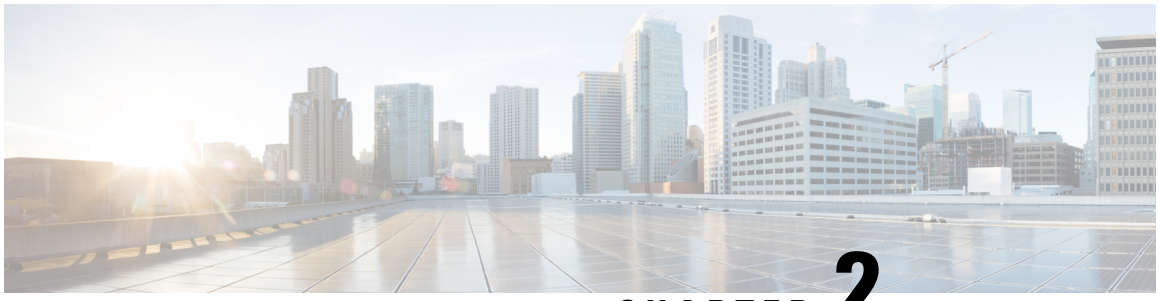
| | |
|---|---|
| **clear configure crypto** | Removes an entire crypto configuration, including IPsec, crypto maps, dynamic crypto maps, and ISAKMP. |
| **clear configure crypto ca trustpoint** | Removes all trustpoints. |
| **clear configure crypto dynamic-map** | Removes all dynamic crypto maps. Includes keywords that let you remove specific dynamic crypto maps. |
| **clear configure crypto map** | Removes all crypto maps. Includes keywords that let you remove specific crypto maps. |
| **clear configure crypto isakmp** | Removes the entire ISAKMP configuration. |
| **clear configure crypto isakmp policy** | Removes all ISAKMP policies or a specific policy. |
| **clear crypto isakmp sa** | Removes the entire ISAKMP SA database. |

# Clear Crypto Map Configurations

The **clear configure crypto** command includes arguments that let you remove elements of the crypto configuration, including IPsec, crypto maps, dynamic crypto maps, CA trustpoints, all certificates, certificate map configurations, and ISAKMP.

Be aware that if you enter the **clear configure crypto** command without arguments, you remove the entire crypto configuration, including all certificates.

For more information, see the **clear configure crypto** command in the *Cisco Secure Firewall ASA Series Command Reference*.



CHAPTER 2

# L2TP over IPsec

This chapter describes how to configure L2TP over IPsec/IKEv1 on the ASA.


- About L2TP over IPsec/IKEv1 VPN, on page 43
- Licensing Requirements for L2TP over IPsec, on page 45
- Prerequisites for Configuring L2TP over IPsec, on page 45
- Guidelines and Limitations, on page 45
- Configuring L2TP over Eclipse with CLI, on page 47
- Feature History for L2TP over IPsec, on page 52


## About L2TP over IPsec/IKEv1 VPN

Layer 2 Tunneling Protocol (L2TP) is a VPN tunneling protocol that allows remote clients to use the public IP network to securely communicate with private corporate network servers. L2TP uses PPP over UDP (port 1701) to tunnel the data.

L2TP protocol is based on the client/server model. The function is divided between the L2TP Network Server (LNS), and the L2TP Access Concentrator (LAC). The LNS typically runs on a network gateway such as a router, while the LAC can be a dial-up Network Access Server (NAS) or an endpoint device with a bundled L2TP client such as Microsoft Windows, Apple iPhone, or Android.

The primary benefit of configuring L2TP with IPsec/IKEv1 in a remote access scenario is that remote users can access a VPN over a public IP network without a gateway or a dedicated line, which enables remote access from virtually anyplace with POTS. An additional benefit is that no additional client software, such as Cisco VPN client software, is required.

**Note** L2TP over IPsec supports only IKEv1. IKEv2 is not supported.

The configuration of L2TP with IPsec/IKEv1 supports certificates using the preshared keys or RSA signature methods, and the use of dynamic (as opposed to static) crypto maps. This summary of tasks assumes completion of IKEv1, as well as pre-shared keys or RSA signature configuration. See Chapter 41, “Digital Certificates,” in the general operations configuration guide for the steps to configure preshared keys, RSA, and dynamic crypto maps.

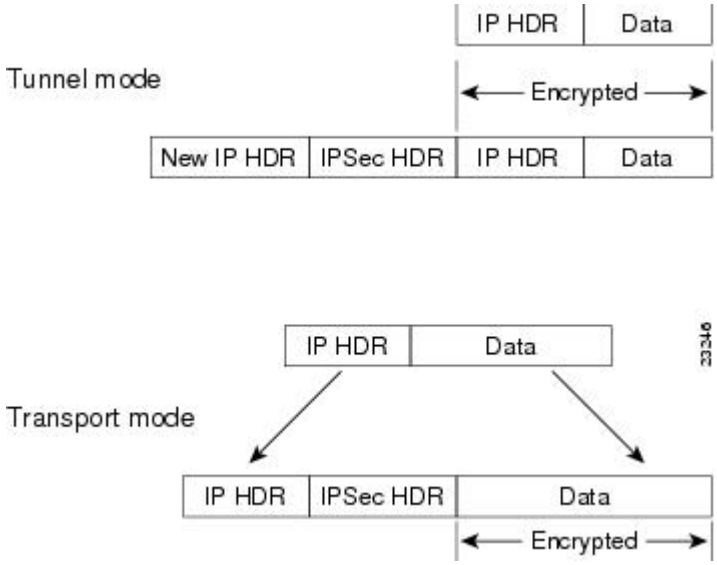
**Note** L2TP with IPsec on the ASA allows the LNS to interoperate with native VPN clients integrated in such operating systems as Windows, MAC OS X, Android, and Cisco IOS. Only L2TP with IPsec is supported, native L2TP itself is not supported on ASA. The minimum IPsec security association lifetime supported by the Windows client is 300 seconds. If the lifetime on the ASA is set to less than 300 seconds, the Windows client ignores it and replaces it with a 300 second lifetime.

# IPsec Transport and Tunnel Modes

By default, the ASA uses IPsec tunnel mode—the entire original IP datagram is encrypted, and it becomes the payload in a new IP packet. This mode allows a network device, such as a router, to act as an IPsec proxy. That is, the router performs encryption on behalf of the hosts. The source router encrypts packets and forwards them along the IPsec tunnel. The destination router decrypts the original IP datagram and forwards it on to the destination system. The major advantage of tunnel mode is that the end systems do not need to be modified to receive the benefits of IPsec. Tunnel mode also protects against traffic analysis; with tunnel mode, an attacker can only determine the tunnel endpoints and not the true source and destination of the tunneled packets, even if they are the same as the tunnel endpoints.

However, the Windows L2TP/IPsec client uses IPsec transport mode—only the IP payload is encrypted, and the original IP headers are left intact. This mode has the advantages of adding only a few bytes to each packet and allowing devices on the public network to see the final source and destination of the packet. The following figure illustrates the differences between IPsec tunnel and transport modes.

***Figure 3: IPsec in Tunnel and Transport Modes***




In order for Windows L2TP and IPsec clients to connect to the ASA, you must configure IPsec transport mode for a transform set using the **crypto ipsec transform-set trans_name mode transport** command. This command is used in the configuration procedure**.**

**Note** ASA cannot push more than 28 ACE in split-tunnel access-list.

With this transport capability, you can enable special processing (for example, QoS) on the intermediate network based on the information in the IP header. However, the Layer 4 header is encrypted, which limits the examination of the packet. Unfortunately, if the IP header is transmitted in clear text, transport mode allows an attacker to perform some traffic analysis.

# Licensing Requirements for L2TP over IPsec

**Note** This feature is not available on No Payload Encryption models.

IPsec remote access VPN using IKEv2 requires an AnyConnect Plus or Apex license, available separately. IPsec remote access VPN using IKEv1 and IPsec site-to-site VPN using IKEv1 or IKEv2 uses the Other VPN license that comes with the Essentials license. See Cisco ASA Series Feature Licenses for maximum values per model.

# Prerequisites for Configuring L2TP over IPsec

Configuring L2TP over IPsec has the following prerequisites:

- Group Policy-You can configure the default group policy (DfltGrpPolicy) or a user-defined group policy for L2TP/IPsec connections. In either case, the group policy must be configured to use the L2TP/IPsec tunneling protocol. If the L2TP/IPsec tunning protocol is not configured for your user-defined group policy, configure the DfltGrpPolicy for the L2TP/IPsec tunning protocol and allow your user-defined group policy to inherit this attribute.
- Connection Profile-You need to configure the default connection proflie (tunnel group), DefaultRAGroup, if you are performing “pre-shared key” authentication. If you are performing certificate-based authentication, you can use a user-defined connection profile that can be chosen based on certificate identifiers.
- IP connectivity needs to be established between the peers. To test connectivity, try to ping the IP address of the ASA from your endpoint and try to ping the IP address of your endpoint from the ASA.
- Make sure that UDP port 1701 is not blocked anywhere along the path of the connection.
- If a Windows 7 endpoint device authenticates using a certificate that specifies a SHA signature type, the signature type must match that of the ASA, either SHA1 or SHA2.

# Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

### Context Mode Guidelines

Supported in single context mode.

### Firewall Mode Guidelines

Supported only in routed firewall mode. Transparent mode is not supported.

### Failover Guidelines

L2TP over IPsec sessions are not supported by stateful failover.

### IPv6 Guidelines

There is no native IPv6 tunnel setup support for L2TP over IPsec.

### Software Limitation on All Platforms

We currently only support 4096 L2TP over IPsec tunnels.

### Authentication Guidelines

The ASA only supports the PPP authentications PAP and Microsoft CHAP, Versions 1 and 2, on the local database. EAP and CHAP are performed by proxy authentication servers. Therefore, if a remote user belongs to a tunnel group configured with the **authentication eap-proxy** or **authentication chap** commands, and the ASA is configured to use the local database, that user will not be able to connect.

### Supported PPP Authentication Types

L2TP over IPsec connections on the ASA support only the PPP authentication types as shown:

*Table 5: AAA Server Support and PPP Authentication Types*

| AAA Server Type | Supported PPP Authentication Types |
|---|---|
| LOCAL | PAP, MSCHAPv1, MSCHAPv2 |
| RADIUS | PAP, CHAP, MSCHAPv1, MSCHAPv2, EAP-Proxy |
| TACACS+ | PAP, CHAP, MSCHAPv1 |
| LDAP | PAP |
| NT | PAP |
| SDI | SDI |

*Table 6: PPP Authentication Type Characteristics*

| Keyword | Authentication Type | Characteristics |
|---|---|---|
| `chap` | CHAP | In response to the server challenge, the client returns the encrypted [challenge plus password] with a cleartext username. This protocol is more secure than the PAP, but it does not encrypt data. |

| Keyword | Authentication Type | Characteristics |
|---|---|---|
| **eap-proxy** | EAP | Enables EAP which permits the security appliance to proxy the PPP authentication process to an external RADIUS authentication server. |
| **ms-chap-v1**<br>**ms-chap-v2** | Microsoft CHAP, Version 1<br>Microsoft CHAP, Version, 2 | Similar to CHAP but more secure in that the server stores and compares only encrypted passwords rather than cleartext passwords as in CHAP. This protocol also generates a key for data encryption by MPPE. |
| **pap** | PAP | Passes cleartext username and password during authentication and is not secure. |

# Configuring L2TP over Eclipse with CLI

You must configure IKEv1 (ISAKMP) policy settings to allow native VPN clients to make a VPN connection to the ASA using the L2TP over Eclipse protocol.

- IKEv1 phase 1— AES encryption with SHA1 hash method.
- Eclipse phase 2 — AES encryption with SHA hash method.
- PPP Authentication—PAP, MS-CHAPv1, or MSCHAPv2 (preferred).
- Pre-shared key (only for iPhone).

### Procedure

**Step 1** Create a transform set with a specific ESP encryption type and authentication type.

**crypto ipsec ike_version transform-set** *transform_name ESP_Encryption_Type ESP_Authentication_Type*

**Example:**

```
crypto ipsec ikev1 transform-set my-transform-set-ikev1 esp-aes esp-sha-hmac
```

**Step 2** Instruct Eclipse to use transport mode rather than tunnel mode.

**crypto ipsec ike_version transform-set** *trans_name* **mode transport**

**Example:**

```
crypto ipsec ikev1 transform-set my-transform-set-ikev1 mode transport
```

**Step 3** Specify L2TP/Eclipse as the vpn tunneling protocol.

**vpn-tunnel-protocol** *tunneling_protocol*

**Example:**

```
hostname(config)# group-policy DfltGrpPolicy attributes
hostname(config-group-policy)# vpn-tunnel-protocol l2tp-ipsec
```

**Step 4** (Optional) Instruct the adaptive security appliance to send DNS server IP addresses to the client for the group policy.

**dns value** [**none** | *IP_Primary* | *IP_Secondary*]

**Example:**

```
hostname(config)# group-policy DfltGrpPolicy attributes
hostname(config-group-policy)# dns value 209.165.201.1 209.165.201.2
```

**Step 5** (Optional) Instruct the adaptive security appliance to send WINS server IP addresses to the client for the group policy.

**wins-server value** [**none** | *IP_primary* [*IP_secondary*]]

**Example:**

```
hostname(config)# group-policy DfltGrpPolicy attributes
hostname (config-group-policy)# wins-server value 209.165.201.3 209.165.201.4
```

**Step 6** (Optional) Create an IP address pool.

**ip local pool** *pool_name starting_address-ending_address* **mask** *subnet_mask*

**Example:**

```
hostname(config)# ip local pool sales_addresses 10.4.5.10-10.4.5.20 mask 255.255.255.0
```

**Step 7** (Optional) Associate the pool of IP addresses with the connection profile (tunnel group).

**address-pool** *pool_name*

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup general-attributes
hostname(config-tunnel-general)# address-pool sales_addresses
```

**Step 8** Link the name of a group policy to the connection profile (tunnel group).

**default-group-policy** name

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup general-attributes
hostname(config-tunnel-general)# default-group-policy DfltGrpPolicy
```

**Step 9** Specify an authentication server to verify users attempting L2TP over the IPsec connections. If you want the authentication to fallback to local authentication when the server is not available, add LOCAL to the end of the command.

**authentication-server-group** *server_group* [**local**]

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup general-attributes
hostname(config-tunnel-general)# authentication-server-group sales_server LOCAL
```

**Step 10** Specify a method to authenticate users attempting L2TP over Eclipse connections, for the connection profile (tunnel group). If you are not using the ASA to perform local authentication, and you want to fallback to local authentication, add LOCAL to the end of the command.

**authentication** *auth_type*

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup ppp-attributes
hostname(config-ppp)# authentication ms-chap-v1
```

**Step 11** Set the pre-shared key for your connection profile (tunnel group).

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

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key cisco123
```

**Step 12** (Optional) Generate a AAA accounting start and stop record for an L2TP session for the connection profile (tunnel group).

**accounting-server-group** *aaa_server_group*

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup general-attributes
hostname(config-tunnel-general)# accounting-server-group sales_aaa_server
```

**Step 13** Configure the interval (in seconds) between hello messages. The range is 10 through 300 seconds. The default interval is 60 seconds.

**l2tp tunnel hello** *seconds*

**Example:**

```
hostname(config)# l2tp tunnel hello 100
```

**Step 14** (Optional) Enable NAT traversal so that ESP packets can pass through one or more NAT devices.

If you expect multiple L2TP clients behind a NAT device to attempt L2TP over Eclipse connections to the adaptive security appliance, you must enable NAT traversal.

**crypto isakmp nat-traversal** *seconds*

To enable NAT traversal globally, check that ISAKMP is enabled (you can enable it with the **crypto isakmp enable** command) in global configuration mode, and then use the **crypto isakmp nat-traversal** command.

**Example:**

```
hostname(config)# crypto ikev1 enable
hostname(config)# crypto isakmp nat-traversal 1500
```

**Step 15** (Optional) Configure tunnel group switching. The goal of tunnel group switching is to give users a better chance at establishing a VPN connection when they authenticate using a proxy authentication server. Tunnel group is synonymous with connection profile.

**strip-group**

**strip-realm**

**Example:**

```
hostname(config)# tunnel-group DefaultRAGroup general-attributes
hostname(config-tunnel-general)# strip-group
hostname(config-tunnel-general)# strip-realm
```

**Step 16** (Optional) Create a user with the username **jdoe**, the password **j!doe1**. The mschap option specifies that the password is converted to Unicode and hashed using MD4 after you enter it.

This step is needed only if you are using a local user database.

**username** *name* **password** *password* **mschap**

**Example:**

```
asa2(config)# username jdoe password j!doe1 mschap
```

**Step 17** Create the IKE Policy for Phase 1 and assign it a number.

**crypto ikev1 policy** *priority*

**group** *Diffie-Hellman Group*

There are several different parameters of the IKE policy that you can configure. You can also specify a Diffie-Hellman Group for the policy. The isakamp policy is used by the ASA to complete the IKE negotiation.

**Example:**

```
hostname(config)# crypto ikev1 policy 14
hostname(config-ikev1-policy)# group14
```

# Creating IKE Policies to Respond to Windows 7 Proposals

Windows 7 L2TP/IPsec clients send several IKE policy proposals to establish a VPN connection with the ASA. Define one of the following IKE policies to facilitate connections from Windows 7 VPN native clients.

Follow the procedure Configuring L2TP over IPsec for ASA. Add the additional steps in this task to configure the IKE policy for Windows 7 native VPN clients.

## Procedure

**Step 1** Display the attributes and the number of any existing IKE policies.

**Example:**

```
hostname(config)# show run crypto ikev1
```

**Step 2** Configure an IKE policy. The number argument specifies the number of the IKE policy you are configuring. This number was listed in the output of the **show run crypto ikev1** command.

**crypto ikev1 policy** *number*

**Step 3** Set the authentication method the ASA uses to establish the identity of each IPsec peer to use preshared keys.

**Example:**

```
hostname(config-ikev1-policy)# authentication pre-share
```

**Step 4** Choose a symmetric encryption method that protects data transmitted between two IPsec peers. For Windows 7, choose **aes** for 128-bit AES, or **aes-256**.

**encryption** {|**aes**|**aes-256**}

**Step 5** Choose the hash algorithm that ensures data integrity. For Windows 7, specify **sha** for the SHA-1 algorithm.

**Example:**

```
hostname(config-ikev1-policy)# hash sha
```

**Step 6** Choose the Diffie-Hellman group identifier. You can specify 14 for aes,aes-256 encryption types.

**Example:**

```
hostname(config-ikev1-policy)# group 14
```

**Step 7** Specify the SA lifetime in seconds. For Windows 7, specify 86400 seconds to represent 24 hours.

**Example:**

```
hostname(config-ikev1-policy)# lifetime 86400
```

# Configuration Example for L2TP over IPsec

The following example shows configuration file commands that ensure ASA compatibility with a native VPN client on any operating system:

```
ip local pool sales_addresses 209.165.202.129-209.165.202.158
group-policy sales_policy internal
group-policy sales_policy attributes
 wins-server value 209.165.201.3 209.165.201.4
 dns-server value 209.165.201.1 209.165.201.2
 vpn-tunnel-protocol l2tp-ipsec
tunnel-group DefaultRAGroup general-attributes
 default-group-policy sales_policy
 address-pool sales_addresses
tunnel-group DefaultRAGroup ipsec-attributes
 pre-shared-key *
tunnel-group DefaultRAGroup ppp-attributes
 no authentication pap
 authentication chap
 authentication ms-chap-v1
 authentication ms-chap-v2

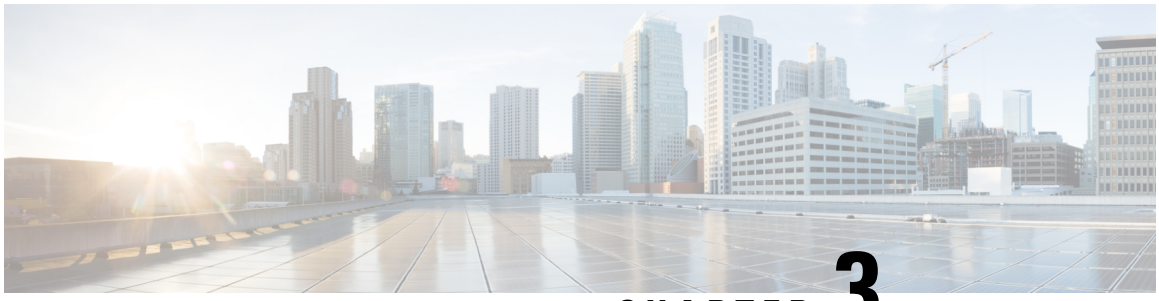

crypto ipsec ikev1 transform-set trans esp-aes esp-sha-hmac
crypto ipsec ikev1 transform-set trans mode transport
crypto dynamic-map dyno 10 set ikev1 transform-set trans
crypto map vpn 20 ipsec-isakmp dynamic dyno
crypto map vpn interface outside
crypto ikev1 enable outside
crypto ikev1 policy 10
 authentication pre-share

 encryption aes
 hash sha

 group 14
 lifetime 86400
```

# Feature History for L2TP over IPsec

| Feature Name | Releases | Feature Information |
|---|---|---|
| L2TP over IPsec | 7.2(1) | L2TP over IPsec provides the capability to deploy and administer an L2<br>solution alongside the IPsec VPN and firewall services in a single platf<br><br>The primary benefit of configuring L2TP over IPsec in a remote access<br>is that remote users can access a VPN over a public IP network without<br>or a dedicated line, which enables remote access from virtually anyplac<br>POTS. An additional benefit is that the only client requirement for VPN<br>the use of Windows with Microsoft Dial-Up Networking (DUN). No ad<br>client software, such as Cisco VPN client software, is required.<br><br>The following commands were introduced or modified: authentication e<br>authentication ms-chap-v1, authentication ms-chap-v2, authentication p<br>tunnel hello, vpn-tunnel-protocol l2tp-ipsec. |
| Deprecations of IKE/IPsec encryption and integrity/PRF ciphers<br><br>DH group 14 support for IKEv1 | 9.13(1) | The following encryption/integrity/PRF ciphers are deprecated and will b<br>in the later release - 9.14(1):<br><br>• 3DES encryption<br>• DES encryption<br>• MD5 integrity<br><br>Added DH group 14 (default) support for IKEv1. The group 2 and group 5<br>options was deprecated and will be removed in the later release- 9.14(1 |



CHAPTER 3

# High Availability Options


- High Availability Options, on page 53
- VPN Load Balancing, on page 54


# High Availability Options

Distributed VPN Clustering, Load balancing and Failover are high-availability features that function differently and have different requirements. In some circumstances you may use multiple capabilities in your deployment. The following sections describe these features. Refer to the appropriate release of the ASA General Operations CLI Configuration Guide for details on Distributed VPN and Failover. Load Balancing details are included here.

## VPN and Clustering on the Secure Firewall eXtensible Operating System (FXOS) Chassis

An ASA FXOS Cluster supports one of two mutually exclusive modes for S2S VPN, centralized or distributed:

- Centralized VPN Mode. The default mode. In centralized mode, VPN connections are established with the control unit of the cluster only.

  VPN functionality is limited to the control unit and does not take advantage of the cluster high availability capabilities. If the control unit fails, all existing VPN connections are lost, and VPN connected users see a disruption in service. When a new control unit is elected, you must reestablish the VPN connections.

  When you connect a VPN tunnel to a Spanned interface address, connections are automatically forwarded to the control unit. VPN-related keys and certificates are replicated to all units.

- Distributed VPN Mode. In this mode, S2S IPsec IKEv2 VPN connections are distributed across members of an ASA cluster providing scalability. Distributing VPN connections across the members of a cluster allows both the capacity and throughput of the cluster to be fully utilized, significantly scaling VPN support beyond Centralized VPN capabilities.



**Note** Centralized VPN clustering mode supports S2S IKEv1 and S2S IKEv2.

Distributed VPN clustering mode supports S2S IKEv2 only.

Distributed VPN clustering mode is supported on the Firepower 9300 only.

Remote access VPN is not supported in centralized or distributed VPN clustering mode.

## VPN Load Balancing

VPN load balancing is a mechanism for equitably distributing remote-access VPN traffic among the devices in a VPN load-balancing group. It is based on simple distribution of traffic without taking into account throughput or other factors. A VPN load-balancing group consists of two or more devices. One device is the director, and the other devices are member devices. Group devices do not need to be of the exact same type, or have identical software versions or configurations.

All active devices in a VPN load-balancing group carry session loads. VPN load balancing directs traffic to the least-loaded device in the group, distributing the load among all devices. It makes efficient use of system resources and provides increased performance and high availability.

## Failover

A failover configuration requires two identical ASAs connected to each other through a dedicated failover link and, optionally, a stateful failover link. The health of the active interfaces and units is monitored to determine when specific failover conditions are met. If those conditions occur, failover occurs. Failover supports both VPN and firewall configurations.

The ASA supports two failover configurations: Active/Active failover and Active/Standby failover.

With Active/Active failover, both units can pass network traffic. This is not true load balancing, although it might appear to have the same effect. When failover occurs, the remaining active unit takes over passing the combined traffic, based on the configured parameters. Therefore, when configuring Active/Active failover, you must make sure that the combined traffic for both units is within the capacity of each unit.

With Active/Standby failover, only one unit passes traffic, while the other unit waits in a standby state and does not pass traffic. Active/Standby failover lets you use a second ASA to take over the functions of a failed unit. When the active unit fails, it changes to the standby state, while the standby unit changes to the active state. The unit that becomes active assumes the IP addresses (or, for transparent firewall, the management IP address) and MAC addresses of the failed unit and begins passing traffic. The unit that is now in standby state takes over the standby IP addresses of the active unit. If an active unit fails, the standby takes over without any interruption to the client VPN tunnel.

# VPN Load Balancing

## About VPN Load Balancing

If you have a remote-client configuration in which you are using two or more ASAs connected to the same network to handle remote sessions, you can configure these devices to share their session load by creating a

VPN load-balancing group. VPN Load balancing directs session traffic to the least loaded device, thus distributing the load among all devices. It makes efficient use of system resources and provides increased performance an availability.

All devices in the VPN load-balancing group carry session loads. One device in the group, the *director*, directs incoming connection requests to the other devices, called *member devices*. The director monitors all devices in the group, keeps track of how busy each is, and distributes the session load accordingly. The role of director is not tied to a physical device; it can shift among devices. For example, if the current director fails, one of the member devices in the group takes over that role and immediately becomes the new director.

The VPN load-balancing group appears to outside clients as a single, virtual IP address. This IP address is not tied to a specific physical device. It belongs to the current director. A VPN client attempting to establish a connection connects first to the virtual IP address. The director then sends back to the client the public IP address of the least-loaded available host in the group. In a second transaction (transparent to the user), the client connects directly to that host. In this way, the VPN load-balancing group director directs traffic evenly and efficiently across resources.

If an ASA in the group fails, the terminated sessions can immediately reconnect to the virtual IP address. The director then directs these connections to another active device in the group. If the director fails, a member device in the group immediately and automatically takes over as the new director. Even if several devices in the group fail, users can continue to connect to the group as long as any one device in the group is up and available.

For each VPN load balancing cluster device, you must configure the public/outside (lbpublic) and private/inside (lbprivate) interfaces.

- Public interface: The device's outside interface used for initial communication to the cluster IP address. This interface is used for the Hello handshake.
- Private interface: The device's inside interface used for messaging between the load balancing cluster members. These messages include keepalives, topology messages, and out-of-service messages related to load balancing.

## VPN Load-Balancing Algorithm

The VPN load-balancing group director maintains a sorted list of group members in ascending IP address order. The load of each member is computed as an integer percentage (the number of active sessions). Secure Client inactive sessions do not count towards the SSL VPN load for VPN load balancing. The director redirects the IPsec and SSL VPN tunnel to the device with the lowest load until it is 1 percent higher than the rest. When all members are 1% higher than the director, the director redirects traffic to itself.

For example, if you have one director and two members, the following cycle applies:



**Note** All nodes start with 0%, and all percentages are rounded half-up.

1. The director takes the connection if all members have a load at 1% higher than the director.
2. If the director does not take the connection, the session is taken by whichever member device has the lowest load percentage.
3. If all members have the same percentage load, the member with the least number of sessions gets the session.

4. If all members have the same percentage load and the same number of sessions, the member with the lowest IP address gets the session.

## VPN Load-Balancing Group Configurations

A VPN load-balancing group can consist of ASAs of the same release or of mixed releases subject to the following restrictions:

- VPN load-balancing groups that consist of both same release ASAs can run VPN load balancing for a mixture of IPsec, Secure Client, and clientless SSL VPN client sessions.
- VPN load-balancing groups that include mixed release ASAs can support IPsec sessions. In such a configuration, however, the ASAs might not reach their full IPsec capacity.

The director of the group assigns session requests to the members of the group. The ASA regards all sessions, SSL VPN or IPsec, as equal, and assigns them accordingly. You can configure the number of IPsec and SSL VPN sessions to allow, up to the maximum allowed by your configuration and license.

We have tested up to 10 nodes in a VPN load-balancing group. Larger groups might work, but we do not officially support such topologies.

## VPN Load Balancing Director Election

### Director Election Process

Each non-master in the virtual cluster maintains a local topology database. This database is updated by the master whenever the topology of the cluster is changed. Each non-master goes into master election state when either no Hello response is received from the master or no Keepalive response is received from the master after maximum retries.

The member performs the following functions during director election:

- Compares the priority of each load balancing unit found in the local topology database.
- If two units with the same priority are found, one with the lower IP address is elected.
- If the member itself is elected, it claims the virtual IP address.
- If one of the other members is elected, the member sends a Hello request to the elected master.
- When two member units try to claim the virtual IP address, the ARP subsystem detects the duplicate IP address condition and sends a notification to ask the member with higher MAC address to give up the director role.

### Hello Handshake

Each member sends a Hello request to the virtual cluster IP address on the outside interface upon startup. If a Hello request is received, the master sends its own Hello request to the member. The non-director member returns a Hello response upon receiving of a Hello request from the director. This concludes the Hello handshake.

Once Hello handshake is completed, the connection is initiated on the inside interface if encryption is configured. If no Hello response is received by the member after maximum retries, the member goes into master election state.

### Keepalive Messages

After a Hello handshake is completed between a member and the director, each member unit sends periodic Keepalive requests to the master with its load information. Keepalive requests are sent by a member unit at one second intervals during normal processing if there is no outstanding keepalive responses from the director. This means that the next keepalive request is sent the next second as long as keepalive responses from the previous request was received. If the member did not receive a keepalive response from the director for the previous keepalive request, no keepalive request are sent the next second. Instead, the member's keepalive timeout logic starts.

The keepalive timeout works as follows:

1. If a member is waiting for an outstanding keepalive response from the director, the member does not send the regular one second interval keepalive request.
2. The member waits for 3 seconds and sends a keepalive request at the 4th second.
3. The member repeats step #2 above five(5) times as long as there is no keepalive response from the director.
4. Then, the member declares the director as gone and starts a new director election cycle.

# Frequently Asked Questions About VPN Load Balancing


- Multiple Context Mode
- IP Address Pool Exhaustion
- Unique IP Address Pools
- Using VPN Load Balancing and Failover on the Same Device
- VPN Load Balancing on Multiple Interfaces
- Maximum Simultaneous Sessions for VPN Load-Balancing Groups

### Multiple Context Mode

**Q.** Is VPN load balancing supported in multiple context mode?

**A.** Neither VPN load balancing nor stateful failover is supported in multiple context mode.

### IP Address Pool Exhaustion

**Q.** Does the ASA consider IP address pool exhaustion as part of its VPN load-balancing method?

**A.** No. If the remote access VPN session is directed to a device that has exhausted its IP address pools, the session does not establish. The load-balancing algorithm is based on load, and is computed as an integer percentage (number of active and maximum sessions) that each member supplies.

### Unique IP Address Pools

**Q.** To implement VPN load balancing, must the IP address pools for Secure Clients or IPsec clients on different ASAs be unique?

**A.** Yes. IP address pools must be unique for each device.

### Using VPN Load Balancing and Failover on the Same Device

**Q.** Can a single device use both VPN load balancing and failover?

**A.** Yes. In this configuration, the client connects to the IP address of the group and is redirected to the least-loaded ASA in the group. If that device fails, the standby unit takes over immediately, and there is no impact to the VPN tunnel.

### VPN Load Balancing on Multiple Interfaces

**Q.** If we enable SSL VPN on multiple interfaces, is it possible to implement VPN load balancing for both of the interfaces?

**A.** You can define only one interface to participate in the VPN load-balancing group as the public interface. The idea is to balance the CPU loads. Multiple interfaces converge on the same CPU, so the concept of VPN load balancing on multiple interfaces does not improve performance.

### Maximum Simultaneous Sessions for VPN Load-Balancing Groups

**Q.** Consider a deployment of two Firepower 1150s, each with a 100-user SSL VPN license. In a VPN load-balancing group, does the maximum total number of users allow 200 simultaneous sessions, or only 100? If we add a third device later with a 100-user license, can we now support 300 simultaneous sessions?

**A.** With VPN load balancing, all devices are active, so the maximum number of sessions that your group can support is the total of the number of sessions for each of the devices in the group, in this case 300.

# Licensing for VPN Load Balancing

VPN load balancing has the following licensing requirements:

- An active 3DES/AES license.

  ASA checks for the existence of this crypto license before enabling VPN load balancing. If it does not detect an active 3DES or AES license, the ASA prevents the enabling of VPN load balancing and also

prevents internal configuration of 3DES by the VPN load-balancing system unless the license permits this usage.

- A valid Security Plus license for this feature activated on your firewall.
- You must have enough of these Security Plus licenses in your Smart account to be compliant.

# Prerequisites for VPN Load Balancing

Also refer to the Guidelines and Limitations for VPN Load Balancing, on page 59.

- VPN load balancing is disabled by default. You must explicitly enable VPN load balancing.
- You must have first configured the public (outside) and private (inside) interfaces. Subsequent references in this section use the names outside and inside.

  You can use the **interface** and **nameif** commands to configure different names for these interfaces.
- You must have previously configured the interface to which the virtual IP address refers. Establish a common virtual IP address, UDP port (if necessary), and IPsec shared secret for the group.
- All devices that participate in a group must share the same cluster-specific values: IP address, encryption settings, encryption key, and port.
- To use VPN load-balancing group encryption, first enable IKEv1 on the inside interface using the **crypto ikev1 enable** command, with the inside interface specified; otherwise you will get an error message when you try to configure VPN load-balancing group encryption.
- The Local CA feature is not supported if you use Active/Active stateful failover or VPN load-balancing. The Local CA cannot be subordinate to another CA; it can act only as the Root CA.

# Guidelines and Limitations for VPN Load Balancing

### Eligible Clients

VPN Load balancing is effective only on remote sessions initiated with the following clients:

- Secure Client (Release 3.0 and later)
- ASA 5505 (when acting as an Easy VPN client)
- Firepower 1010 (when acting as an Easy VPN client)
- IOS EZVPN Client devices supporting IKE-redirect (IOS 831/871)

### Client Considerations

VPN load balancing works with IPsec clients and SSL VPN client sessions. All other VPN connection types (L2TP, PPTP, L2TP/IPsec), including LAN-to-LAN, can connect to an ASA on which VPN load balancing is enabled, but they cannot participate in VPN load balancing.

When multiple ASA nodes are grouped for load balancing, and using Group URLs is desired for Secure Client connections, the individual ASA nodes must:

- Configure each remote access connection profile with a Group URL for each VPN load-balancing virtual address (IPv4 and IPv6).
- Configure a Group URL for this node's VPN load-balancing public address.

### Load Balancing Group

- ASA supports 10 devices per VPN load balancing group.
- UCAPL mode does not support VPN load balancing, even when encryption is disabled. In UCAPL mode, use IKEv2 for secure tunnel establishment.

### Context Mode

VPN load balancing is not supported in multiple context mode.

### FIPS

Cluster encryption not supported with FIPS.

### Certificate Verification

When performing certificate verification for VPN load balancing with Secure Client, and the connection is redirected by an IP address, the client does all of its name checking through this IP address. Make sure the redirection IP address is listed in the certificates common name or the subject alt name. If the IP address is not present in these fields, then the certificate will be deemed untrusted.

Following the guidelines defined in RFC 2818, if a **subject alt name** is included in the certificate, we only use the **subject alt name** for name checks, and we ignore the common name. Make sure that the IP address of the server presenting the certificate is defined in the **subject alt name** of the certificate.

For a standalone ASA, the IP address is the IP of that ASA. In a VPN load-balancing group situation, it depends on the certificate configuration. If the group uses one certificate, then the certificate should have SAN extensions for the virtual IP address and group FQDN and should contain Subject Alternative Name extensions that have each ASA's IP and FQDN. If the group uses multiple certificates, then the certificate for each ASA should have SAN extensions for the virtual IP, group FQDN, and the individual ASA's IP address and FQDN.

### Geographical VPN Load Balancing

In a VPN load balancing environment where the DNS resolutions are being changed at regular intervals, you must carefully consider how to set the time to live (TTL) value. For the DNS load balance configuration to work successfully with Secure Client, the ASA name-to-address mapping must remain the same from the time the ASA is selected until the tunnel is fully established. If too much time passes before the credentials are entered, the lookup restarts and a different IP address may become the resolved address. If the DNS mapping changes to a different ASA before the credentials are entered, the VPN tunnel fails.

Geographical load balancing for VPN often uses a Cisco Global Site Selector (GSS). The GSS uses DNS for the load balancing, and the time to live (TTL) value for DNS resolution is defaulted to 20 seconds. You can significantly decrease the likelihood of connection failures if you increase the TTL value on the GSS. Increasing to a much higher value allows ample time for the authentication phase when the user is entering credentials and establishing the tunnel.

To increase the time for entering credentials, you may also consider disabling Connect on Start Up.

### IKE/IPSec Security Associations

Cluster encryption sessions do not sync to standby in a VPN load balancer topology.

### Upgrade Guidelines

During a software upgrade, the VPN load-balancing cluster does not form if members run different versions. Perform one of these actions:

1. Upgrade all members to the same version. The VPN load-balancing cluster re-forms automatically after all members run the same version.

   VPN load balancing is unavailable during the mixed-version window.

2. For a hitless upgrade, temporarily remove the cluster keys from all members. Keep the interval without cluster keys as short as possible. Upgrade all members and then reconfigure the cluster keys on all members.

   VPN load balancing remains available during the upgrade.



**Note** This requirement applies to ASA Version 9.22.3.26, 9.20.4.49, 9.24.1.26, 9.18.4.94, 9.23.1.47, and 9.16.4.103.

In these ASA versions, if you choose option 2, syslog 718089 is logged:

**%ASA-1-718089: VPN LB cluster is running without authentication. Configure 'cluster key' to secure the cluster.**

Cluster encryption and cluster keys secure authentication between VPN load-balancing members and help protect the group from denial-of-service (DoS) and spoofing attacks.

# Configuring VPN Load Balancing

If you have a remote-client configuration in which you are using two or more ASAs connected to the same network to handle remote sessions, you can configure these devices to share their session load. This feature is called VPN load balancing, which directs session traffic to the least loaded device, thereby distributing the load among all devices. VPN load balancing makes efficient use of system resources and provides increased performance and system availability.

To use VPN load balancing, do the following on each device in the group:

- Configure the VPN load-balancing group by establishing common VPN load-balancing group attributes. This includes a virtual IP address, UDP port (if necessary), and IPsec shared secret for the group. All participants in the group must have an identical group configuration, except for the device priority within the group.
- Configure the participating device by enabling VPN load balancing on the device and defining device-specific properties, such as its public and private addresses. These values vary from device to device.

## Configure the Public and Private Interfaces for VPN Load Balancing

To configure the public (outside) and private (inside) interfaces for the VPN load-balancing group devices, do the following steps.

**Procedure**

**Step 1** Configure the public interface on the ASA by entering the **interface** command with the **lbpublic** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the public interface for VPN load balancing for this device:

**Example:**

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# interface lbpublic outside
hostname(config-load-balancing)#
```

**Step 2** Configure the private interface on the ASA by entering the **interface** command with the **lbprivate** keyword in vpn-load-balancing configuration mode. This command specifies the name or IP address of the private interface for VPN load balancing for this device:

**Example:**

```
hostname(config-load-balancing)# interface lbprivate inside
hostname(config-load-balancing)#
```

**Step 3** Set the priority to assign to this device within the group. The range is from 1 to 10. The priority indicates the likelihood of this device becoming the group director, either at the startup of the device or when an existing director fails. The higher you set the priority (for example, 10), the more likely it is that this device becomes the group director.

**Example:**

For example, to assign this device a priority of 6 within the group, enter the following command:

```
hostname(config-load-balancing)# priority 6
hostname(config-load-balancing)#
```

**Step 4** If you want to apply network address translation for this device, enter the **nat** command with the NAT assigned address for the device. You can define an IPv4 and an IPv6 address or specify the device's hostname.

**Example:**

For example, to assign this device a NAT address of 192.168.30.3 and 2001:DB8::1, enter the following command:

```
hostname(config-load-balancing)# nat 192.168.30.3 2001:DB8::1
hostname(config-load-balancing)#
```

## Configure the VPN Load Balancing Group Attributes

To configure the VPN load-balancing group attributes for each device in the group, do the following steps:

### Procedure

**Step 1** Set up VPN load balancing by entering the **vpn load-balancing** command in global configuration mode:

**Example:**

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)#
```

This enters vpn-load-balancing configuration mode, in which you can configure the remaining load-balancing attributes.

**Step 2** Configure the IP address or the fully qualified domain name of the group to which this device belongs. This command specifies the single IP address or FQDN that represents the entire VPN load-balancing group. Choose an IP address that is within the public subnet address range shared by all the ASAs in the group. You must specify an IPv4 (mandatory). You can optionally provide an IPv6 address.

**Example:**

To configure the virtual IPv4 and IPv6 addresses, enter the following command:

```
hostname(config-load-balancing)# cluster ip address 192.168.10.1 1000::2
hostname(config-load-balancing)#show running-config vpn load-balancing
vpn load-balancing
redirect-fqdn enable
cluster key *****
cluster ip address 192.168.10.1 1000::2
cluster encryption
```

To configure an IPv6 address for a VPN load-balancing cluster, an IPv4 address configuration is mandatory. If you configure only a virtual IPv6 address, an error message will appear.

```
hostname(config-load-balancing)#show running-config vpn load-balancing
vpn load-balancing
redirect-fqdn enable
cluster key *****
cluster encryption
participate
hostname(config-load-balancing)# cluster ip address 1000::2
ERROR: Virtual IPv4 address is not set
```

**Step 3** Configure the group port. This command specifies the UDP port for the VPN load-balancing group in which this device is participating. The default value is 9023. If another application is using this port, enter the UDP destination port number that you want to use for load balancing.

**Example:**

For example, to set the group port to 4444, enter the following command:

```
hostname(config-load-balancing)# cluster port 4444
hostname(config-load-balancing)#
```

**Step 4** (Optional) Enable IPsec encryption for the VPN load-balancing group.

The default is no encryption. This command enables or disables IPsec encryption. If you configure this check attribute, you must first specify and verify a shared secret.The ASAs in the VPN load-balancing group communicate via LAN-to-LAN tunnels using IPsec. To ensure that all load-balancing information communicated between the devices is encrypted, enable this attribute.

**Note**
To use VPN load-balancing group encryption, first enable IKEv1 on the inside interface using the **crypto ikev1 enable** command, with the inside interface specified; otherwise, you will get an error message when you try to configure VPN load-balancing group encryption.

If IKEv1 was enabled when you configured group encryption, but was disabled before you configured the participation of the device in the group, you get an error message when you enter the **participate** command, and encryption is not enabled for the group.

**Example:**

```
hostname(config)# crypto ikev1 enable inside
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# cluster encryption
hostname(config-load-balancing)#
```

**Step 5** If you enable group encryption, you must also specify the IPsec shared secret by entering the **cluster key** command. This command specifies the shared secret between IPsec peers when you have enabled IPsec encryption. The value you enter in the box appears as consecutive asterisk characters. If you need to enter an already encrypted key (for example, you copied it from another configuration), enter the **cluster key 8** *key* command.

**Example:**

For example, to set the shared secret to 123456789, enter the following command:

```
hostname(config-load-balancing)# cluster key 123456789
hostname(config-load-balancing)#
```

**Step 6** Enable this device's participation in the group by entering the **participate** command:

**Example:**

```
hostname(config-load-balancing)# participate
hostname(config-load-balancing)#
```

---

**What to do next**

When multiple ASA nodes are grouped for load balancing, and using Group URLs is desired for Secure Client connections, on the individual ASA nodes you must:

- Configure each remote access connection profile with a Group URL for each load balancing virtual address (IPv4 and IPv6).
- Configure a Group URL for this node's VPN Load Balancing public address.

Use the **tunnel-group**, **general-attributes**, **group-url** command to configure these Group URLs.

### Enable Redirection Using a Fully Qualified Domain Name

By default, the ASA sends only IP addresses in VPN load-balancing redirection to a client. If certificates are in use that are based on DNS names, the certificates will be invalid when redirected to a member device.

As a VPN load-balancing director, this ASA can send a fully qualified domain name (FQDN), using reverse DNS lookup, of a member device (another ASA in the group) instead of its outside IP address when redirecting VPN client connections to that member device.

To enable or disable redirection using a fully qualified domain name in vpn load-balancing mode, use the **redirect-fqdn enable** command in global configuration mode. This behavior is disabled by default.

#### Before you begin

All of the outside and inside network interfaces on the VPN load-balancing devices in a group must be on the same IP network.

#### Procedure

**Step 1** Enable the use of FQDNs for VPN load balancing.

**redirect-fqdn** {**enable** | **disable**}

**Example:**

```
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# redirect-fqdn enable
hostname(config-load-balancing)#
```

**Step 2** Add an entry for each of your ASA outside interfaces into your DNS server if such entries are not already present. Each ASA outside IP address should have a DNS entry associated with it for lookups. These DNS entries must also be enabled for reverse lookup.

**Step 3** Enable DNS lookups on your ASA with the **dns domain-lookup inside** command or whichever interface has a route to your DNS server.

**Step 4** Define your DNS server IP address on the ASA. for example: **dns name-server 10.2.3.4** (IP address of your DNS server).

## Configuration Examples for VPN Load Balancing

### Basic VPN Load Balancing CLI Configuration

The following is an example of a VPN load balancing command sequence that includes an interface command that enables redirection for a fully qualified domain name, specifies the public interface of the group as **test** and the private interface of the group as **foo**”

```
hostname(config)# interface GigabitEthernet 0/1
hostname(config-if)# ip address 209.165.202.159 255.255.255.0
hostname(config)# nameif test
hostname(config)# interface GigabitEthernet 0/2
hostname(config-if)# ip address 209.165.201.30 255.255.255.0
```

```
hostname(config)# nameif foo
hostname(config)# vpn load-balancing
hostname(config-load-balancing)# nat 192.168.10.10
hostname(config-load-balancing)# priority 9
hostname(config-load-balancing)# interface lbpublic test
hostname(config-load-balancing)# interface lbprivate foo
hostname(config-load-balancing)# cluster ip address 209.165.202.224
hostname(config-load-balancing)# cluster key 123456789
hostname(config-load-balancing)# cluster encryption
hostname(config-load-balancing)# cluster port 9023
hostname(config-load-balancing)# redirect-fqdn enable
hostname(config-load-balancing)# participate
```

# Viewing VPN Load Balancing Information

The VPN load-balancing group director receives a periodic message from each ASA in the group with the number of active Secure Client and clientless sessions, as well as the maximum allowed sessions based on the configured or license limits. If an ASA in the group shows 100 percent full capacity, the group director cannot redirect more connections to it. Although the ASA may show as full, some users may be in inactive/wait-to-resume state, wasting the licenses. As a workaround, each ASA provides the total number of sessions minus the sessions in inactive state, instead of the total number of sessions. Refer to the **-sessiondb summary** command in the ASA command reference. In other words, the inactive sessions are not reported to the group director. Even if the ASA is full (with some inactive sessions), the group director still redirects connections to it if necessary. When the ASA receives the new connection, the session that has been inactive the longest is logged off, allowing new connections to take its license.

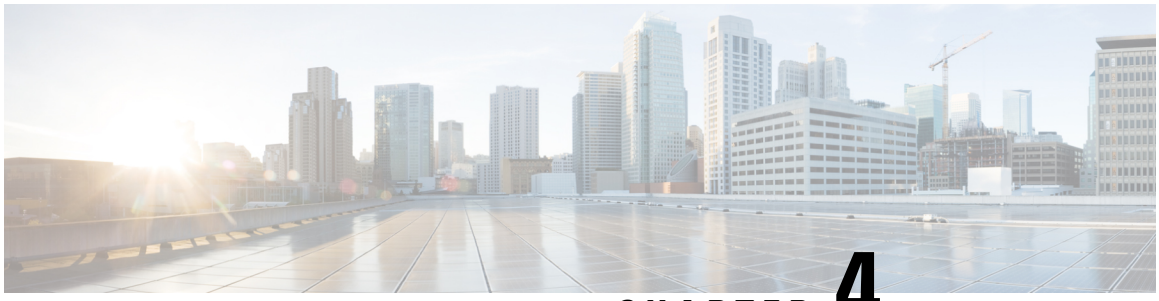
The following example shows 100 SSL sessions (active only) and a 2 percent SSL load. These numbers do not include the inactive sessions. In other words, inactive sessions do not count towards the load for VPN load balancing.

```
hostname# show vpn load-balancing
 Status :    enabled
 Role :    Master
 Failover :    Active
 Encryption :    enabled
 Cluster IP :    192.168.1.100
 Peers :    1

 Load %
Sessions
 Public IP     Role   Pri Model    IPsec SSL IPsec SSL
 192.168.1.9   Master 7   ASA-5540 4     2   216   100
 192.168.1.19  Backup 9   ASA-5520 0     0   0     0
```

# Feature History for VPN Load Balancing

| Feature Name | Releases | Feature Information |
|---|---|---|
| VPN Load balancing with SAML | 9.17(1) | ASA now supports VPN load balancing with SAML authentication. |
| VPN Load balancing | 7.2(1) | This feature was introduced. |



CHAPTER 4

# General VPN Parameters

The ASA implementation of virtual private networking includes useful features that do not fit neatly into categories. This chapter describes some of these features.


- Guidelines and Limitations, on page 67
- Configure IPsec to Bypass ACLs, on page 68
- Permitting Intra-Interface Traffic (Hairpinning), on page 68
- Setting Maximum Active IPsec or SSL VPN Sessions, on page 70
- Use Client Update to Ensure Acceptable IPsec Client Revision Levels, on page 70
- Implement NAT-Assigned IP to Public IP Connection, on page 72
- Configure VPN Session Limits, on page 74
- Using an Identify Certificate When Negotiating, on page 75
- Configure the Pool of Cryptographic Cores, on page 76
- Configure Dynamic Split Tunneling, on page 76
- Configure the Management VPN Tunnel, on page 77
- Viewing Active VPN Sessions, on page 78
- About ISE Policy Enforcement, on page 79
- Configure Advanced SSL Settings, on page 84
- Rate Limit for Preauthenticated SSL Connections, on page 89
- Persistent IPsec Tunneled Flows, on page 91
- Troubleshooting Using Crypto Archives, on page 94
- Using SSL Counters, on page 95
- How to Remove Stuck ASP Table Entries, on page 96
- Clear WebVPN Configurations from ASA, on page 97


# Guidelines and Limitations

This section includes the guidelines and limitations for this feature.

### Context Mode Guidelines

Supported in single and multiple context mode. In the appropriate release of the ASA General Operations CLI Configuration Guide, refer to *Guidelines for Multiple Context Mode* for the list of what is not supported in multiple context mode, and *New Features* which gives the breakdown of what was added throughout the releases.

### Firewall Mode Guidelines

Supported only in routed firewall mode. Transparent mode is not supported.

### Network Address Translation (NAT)

For guidelines and information about NAT configuration, see the *NAT for VPN* section of the *Cisco Secure Firewall ASA Series Firewall CLI Configuration Guide*.

# Configure IPsec to Bypass ACLs

To permit any packets that come from an IPsec tunnel without checking ACLs for the source and destination interfaces, enter the **sysopt connection permit-vpn** command in global configuration mode.

You might want to bypass interface ACLs for IPsec traffic if you use a separate VPN concentrator behind the ASA and want to maximize the ASA performance. Typically, you create an ACL that permits IPsec packets by using the **access-list** command and apply it to the source interface. Using an ACL allows you to specify the exact traffic you want to allow through the ASA.

The following example enables IPsec traffic through the ASA without checking ACLs:

```
hostname(config)# sysopt connection permit-vpn
```



**Note** Decrypted through-traffic is permitted from the client despite having an access group on the outside interface, which calls a **deny ip any any** ACL, while **no sysopt connection permit-vpn** is configured.

Trying to control access to the protected network via site-to-site or remote access VPN using the **no sysopt permit-vpn** command in conjunction with an access control list (ACL) on the outside interface are not successful.

**sysopt connection permit-vpn** will bypass ACLs (both in and out) on interface where crypto map for that interesting traffic is enabled, along with egress (out) ACLs of all other interfaces, but not the ingress (in) ACLs.

In this situation, when management-access inside is enabled, the ACL is not applied, and users can still connect to the ASA using SSH. Traffic to hosts on the inside network is blocked correctly by the ACL, but decrypted through-traffic to the inside interface is not blocked.

The **ssh** and **http** commands are of a higher priority than the ACLs. To deny SSH, Telnet, or ICMP traffic to the box from the VPN session, use **ssh**, **telnet** and **icmp** commands.

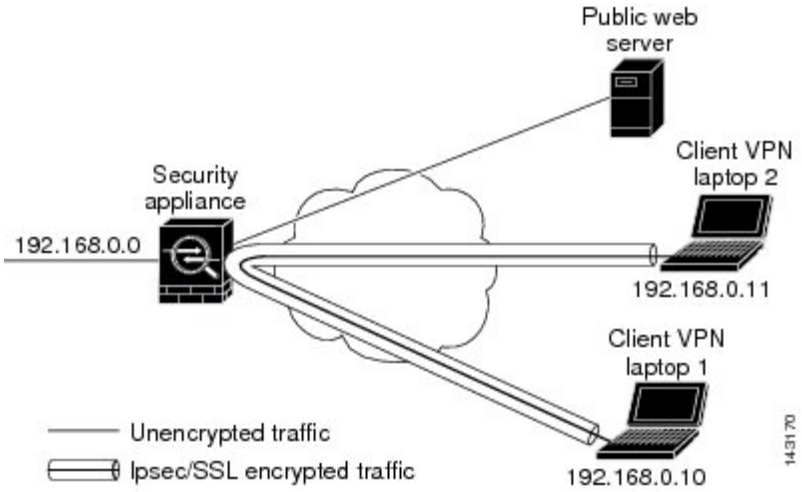
# Permitting Intra-Interface Traffic (Hairpinning)

The ASA includes a feature that lets a VPN client send IPsec-protected traffic to another VPN user by allowing that traffic in and out of the same interface. This is also called “hairpinning”, which can be thought of as VPN spokes (clients) connecting through a VPN hub (the ASA).

Hairpinning can also redirect incoming VPN traffic back out through the same interface as unencrypted traffic. This can be useful, for example, to a VPN client that does not have split tunneling, but needs to both access a VPN and browse the web.

The figure below shows VPN Client 1 sending secure IPsec traffic to VPN Client 2 while also sending unencrypted traffic to a public web server.

***Figure 4: VPN Client Using Intra-Interface Feature for Hairpinning***




To configure this feature, use the **same-security-traffic** command in global configuration mode with its intra-interface argument.

The command syntax is same-security-traffic permit {*inter-interface* | *intra-interface*}.

The following example shows how to enable intra-interface traffic:

```
hostname(config)# same-security-traffic permit intra-interface
hostname(config)#
```

**Note** Use the **same-security-traffic** command with the **inter-interface** argument to permit communication between interfaces with the same security level. This feature is not specific to IPsec connections. For more information, see the "Configuring Interface Parameters" chapter of this guide.

To use hairpinning, you must apply the proper NAT rules to the ASA interface, as described in NAT Considerations for Intra-Interface Traffic.

# NAT Considerations for Intra-Interface Traffic

For the ASA to send unencrypted traffic back out through the interface, you must enable NAT for the interface so that publicly routable addresses replace your private IP addresses (unless you already use public IP addresses in your local IP address pool). The following example applies an interface PAT rule to traffic sourced from the client IP pool:

```
hostname(config)# ip local pool clientpool 192.168.0.10-192.168.0.100
hostname(config)# object network vpn_nat
hostname(config-network-object)# subnet 192.168.0.0 255.255.255.0
hostname(config-network-object)# nat (outside,outside) interface
```

When the ASA sends encrypted VPN traffic back out this same interface, however, NAT is optional. The VPN-to-VPN hairpinning works with or without NAT. To apply NAT to all outgoing traffic, implement only the commands above. To exempt the VPN-to-VPN traffic from NAT, add commands (to the example above) that implement NAT exemption for VPN-to-VPN traffic, such as:

```
hostname(config)# nat (outside,outside) source static vpn_nat vpn_nat destination static
vpn_nat vpn_nat
```

For more information on NAT rules, see the "Applying NAT" chapter of this guide.

# Setting Maximum Active IPsec or SSL VPN Sessions

To limit VPN sessions to a lower value than the ASA allows, enter the **vpn-sessiondb** command in global configuration mode:

**vpn-sessiondb** {**max-anyconnect-premium-or-essentials-limit** <number> | **max-other-vpn-limit** <number>}

The **max-anyconnect-premium-or-essentials-limit** keyword specifies the maximum number of Secure Client sessions, from 1 to the maximum sessions allowed by the license.

**Note** The correct licensing, term, tier, and user count is no longer determined with these commands. Refer to the Secure Client Ordering Guide: http://www.cisco.com/c/dam/en/us/products/collateral/security/anyconnect-og.pdf

The **max-other-vpn-limit** keyword specifies the maximum number of VPN sessions other than the Secure Client sessions, from 1 to the maximum sessions allowed by the license. This includes the Cisco VPN client (IPsec IKEv1) and Lan-to-Lan VPN sessions.

This limit affects the calculated load percentage for VPN Load Balancing.

The following example shows how to set a maximum Anyconnect VPN session limit of 450:

```
hostname(config)# vpn-sessiondb max-anyconnect-premium-or-essentials-limit 450
hostname(config)#
```

# Use Client Update to Ensure Acceptable IPsec Client Revision Levels

**Note** The information in this section applies to IPsec connections only.

The client update feature lets administrators at a central location automatically notify VPN client users that it is time to update the VPN client software.

Remote users might be using outdated VPN software or hardware client versions. You can use the **client-update** command at any time to enable updating client revisions; specify the types and revision numbers of clients to which the update applies; provide a URL or IP address from which to get the update; and, in the case of

Windows clients, optionally notify users that they should update their VPN client version. For Windows clients, you can provide a mechanism for users to accomplish that update. This command applies only to the IPsec remote-access tunnel-group type.

To perform a client update, enter the **client-update** command in either general configuration mode or tunnel-group ipsec-attributes configuration mode. If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. The following procedure explains how to perform a client update:

### Procedure

**Step 1** In global configuration mode, enable client update by entering this command:

```
hostname(config)# client-update enable
hostname(config)#
```

**Step 2** In global configuration mode, specify the parameters for the client update that you want to apply to all clients of a particular type. That is, specify the type of client, the URL or IP address from which to get the updated image, and the acceptable revision number or numbers for that client. You can specify up to four revision numbers, separated by commas.

If the user's client revision number matches one of the specified revision numbers, there is no need to update the client. This command specifies the client update values for all clients of the specified type across the entire ASA.

Use this syntax:

```
hostname(config)# client-update type type url url-string rev-nums rev-numbers
hostname(config)#
```

The available client types are **win9X** (includes Windows 95, Windows 98 and Windows ME platforms), **winnt** (includes Windows NT 4.0, Windows 2000 and Windows XP platforms), **windows** (includes all Windows based platforms).

If the client is already running a software version on the list of revision numbers, it does not need to update its software. If the client is not running a software version on the list, it should update. You can specify up to three of these client update entries. The keyword **windows** covers all of the allowable Windows platforms. If you specify **windows**, do not specify the individual Windows client types.

**Note**
For all Windows clients, you must use the protocol http:// or https:// as the prefix for the URL.

The following example configures client update parameters for the remote access tunnel group. It designates the revision number 4.6.1 and the URL for retrieving the update, which is https://support/updates.

```
hostname(config)# client-update type windows url https://support/updates/ rev-nums 4.6.1
hostname(config)#
```

Alternatively, you can configure client update just for individual tunnel groups, rather than for all clients of a particular type. (See Step 3.)

**Note**

You can have the browser automatically start an application by including the application name at the end of the URL; for example: **https://support/updates/vpnclient.exe.**

**Step 3** Define a set of client-update parameters for a particular ipsec-ra tunnel group.

In tunnel-group ipsec-attributes mode, specify the tunnel group name and its type, the URL or IP address from which to get the updated image, and a revision number. If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client, for example, for a Windows client enter this command:

```
hostname(config)# tunnel-group remotegrp type ipsec-ra
hostname(config)# tunnel-group remotegrp ipsec-attributes
hostname(config-tunnel-ipsec)# client-update type windows url https://support/updates/
rev-nums 4.6.1
hostname(config-tunnel-ipsec)#
```

**Step 4** (Optional) Send a notice to active users with outdated Windows clients that their client needs updating. For these users, a pop-up window appears, offering them the opportunity to launch a browser and download the updated software from the site that you specified in the URL. The only part of this message that you can configure is the URL. (See Step 2 or 3.) Users who are not active get a notification message the next time they log on. You can send this notice to all active clients on all tunnel groups, or you can send it to clients on a particular tunnel group. For example, to notify all active clients on all tunnel groups, enter the following command in privileged EXEC mode:

```
hostname# client-update all
hostname#
```

If the user's client's revision number matches one of the specified revision numbers, there is no need to update the client, and no notification message is sent to the user.

### What to do next

**Note** If you specify the client-update type as **windows** (specifying all Windows-based platforms) and later want to enter a client-update type of **win9x** or **winnt** for the same entity, you must first remove the windows client type with the **no** form of the command, then use new client-update commands to specify the new client types.

# Implement NAT-Assigned IP to Public IP Connection

In rare situations, you might want to use a VPN peer's real IP address on the inside network instead of an assigned local IP address. Normally with VPN, the peer is given an assigned local IP address to access the inside network. However, you might want to translate the local IP address back to the peer-s real public address if, for example, your inside servers and network security is based on the peer's real IP address.

The ASA introduced a way to translate the VPN client's assigned IP address on the internal/protected network to its public (source) IP address. This feature supports the scenario where the target servers/services on the

internal network and network security policy require communication with the VPN client's public/source IP instead of the assigned IP on the internal corporate network.

You can enable this feature on one interface per tunnel group. Object NAT rules are dynamically added and deleted when the VPN session is established or disconnected.

Because of routing issues, we do not recommend using this feature unless you know you need it.

- Only supports legacy (IKEv1) and Secure Clients.
- Return traffic to the public IP addresses must be routed back to the ASA so the NAT policy and VPN policy can be applied.
- Only supports IPv4 assigned and public addresses.
- Multiple peers behind a NAT/PAT device are not supported.
- Does not support load balancing (because of routing issue).
- Does not support roaming.

### Procedure

**Step 1** In global configuration mode, enter **tunnel general**.

**Step 2** Use this syntax to enable the address translation:

```
hostname(config-tunnel-general)# nat-assigned-to-public-ip interface
```

This command dynamically installs NAT policies of the assigned IP address to the public IP address of the source. The *interface* determines where to apply NAT.

**Step 3** Use this syntax to disable the address translation:

```
hostname(config-tunnel-general)# no nat-assigned-to-public-ip
```

# Displaying VPN NAT Policies

Address translation uses the underlying object NAT mechanisms; therefore, the VPN NAT policy displays just like manually configured object NAT policies. This example uses 95.1.226.4 as the assigned IP and 75.1.224.21 as the peer's public IP:

```
hostname# show nat
Auto NAT Policies (Section 2)
1 (outside) to (inside) source static _vpn_nat_95.1.226.4 75.1.224.21
    translate_hits = 315, untranslate_hits = 315

prompt# show nat detail

Auto NAT Policies (Section 2)
1 (outside) to (inside) source static _vpn_nat_95.1.226.4 75.1.224.21
    translate_hits = 315, untranslate_hits = 315
```

```
Source - Origin: 95.1.226.4/32, Translated: 75.1.224.21/32
```

*Outside* is the interface to which the Secure Client connects and *inside* is the interface specific to the new tunnel group.



**Note** Since VPN NAT policies are dynamic and not added to the configuration, the VPN NAT object and NAT policy are hidden from the show run object and show run nat reports.

# Configure VPN Session Limits

You can run as many IPsec and SSL VPN sessions as your platform and ASA license supports. To view the licensing information including maximum sessions for your ASA, enter the **show version** command in global configuration mode and look for the licensing section. The following example shows the command and the licensing information from the output of this command; the other output is redacted for clarity.

```
hostname(config)# show version
...
Licensed features for this platform:
Maximum Physical Interfaces       : Unlimited      perpetual
Maximum VLANs                     : 500            perpetual
Inside Hosts                      : Unlimited      perpetual
Failover                          : Active/Active  perpetual
Encryption-DES                    : Enabled        perpetual
Encryption-3DES-AES               : Enabled        perpetual
Security Contexts                 : 100            perpetual
Carrier                           : Enabled        perpetual
AnyConnect Premium Peers          : 5000           perpetual
AnyConnect Essentials             : 5000           perpetual
Other VPN Peers                   : 5000           perpetual
Total VPN Peers                   : 5000           perpetual
AnyConnect for Mobile             : Enabled        perpetual
AnyConnect for Cisco VPN Phone    : Enabled        perpetual
Advanced Endpoint Assessment      : Enabled        perpetual
Shared License                    : Disabled       perpetual
Total TLS Proxy Sessions          : 3000           perpetual
Botnet Traffic Filter             : Disabled       perpetual
IPS Module                        : Disabled       perpetual
Cluster                           : Enabled        perpetual
Cluster Members                   : 2              perpetual

This platform has an ASA5555 VPN Premium license.
```

# Show License Resource Allocation

Use the following command to show the resource allocation:

```
asa2(config)# sh resource allocation
Resource       Total    % of Avail
Conns[rate]    100(U)      0.00%
Inspects[rate] unlimited
Syslogs[rate]  unlimited
Conns          unlimited
```

```
Hosts           unlimited
IPsec           unlimited
Mac-addresses   unlimited
ASDM            10          5.00%
SSH             10          10.00%
Telnet          10          10.0%
Xlates          unlimited
AnyConnect      1000        10%
AnyConnectBurst 200         2%
OtherVPN        2000        20%
OtherVPNBurst   1000        10%
```

## Show License Resource Usage

Use the following command to show resource usage:



**Note** You can also use the **sh resource usage system controller all 0** command to show system level usage with the limit as the platform limit.

```
ASA(config-ca-trustpoint)# sh resource usage
Resource        Current  Peak  Limit  Denied  Context
Conns           1        16    280000 0       System
Hosts           2        10    N/A    0       System
AnyConnect      2        25    1000   0       cust1
AnyConnectBurst 0        0     200    0       cust1
OtherVPN        1        1     2000   0       cust2
OtherVPNBurst   0        0     1000   0       cust2
```

## Limit VPN Sessions

To limit AnyConnect VPN sessions (either IPsec/IKEv2 or SSL) to a lower value than the ASA allows, use the **vpn-sessiondb** max-anyconnect-premium-or-essentials-limit command in global configuration mode. To remove the session limit, use the **no** version of this command.

If the ASA license allows 500 SSL VPN sessions, and you want to limit the number of AnyConnect VPN sessions to 250, enter the following command:

```
hostname(config)# vpn-sessiondb max-anyconnect-premium-or-essentials-limit 250
hostname(config)#
```

To remove the session limit, use the **no** version of this command.:

```
hostname(config)# no vpn-sessiondb max-anyconnect-premium-or-essentials-limit 250
hostname(config)#
```

# Using an Identify Certificate When Negotiating

The ASA needs to use an identity certificate when negotiating the IKEv2 tunnel with Secure Clients. For ikev2 remote access trustpoint configuration, use the following commands

```
crypto ikev2 remote-access trustpoint <name> [line<number>]
```

Using this command allows the Secure Client to support group selection for the end user. You can configure two trustpoints at the same time: two RSA, two ECDSA, or one of each. The ASA scans the configured trustpoint list and chooses the first one that the client supports. If ECDSA is preferred, you should configure that trustpoint before the RSA trustpoint.

The line number option specifies where in the line number you want the trustpoint inserted. Typically, this option is used to insert a trustpoint at the top without removing and re-adding the other line. If a line is not specified, the ASA adds the trustpoint at the end of the list.

If you try to add a trustpoint that already exists, you receive an error. If you use the *no crypto ikev2 remote-access trustpoint* command without specifying which trustpoint name to remove, all trustpoint configuration is removed.

# Configure the Pool of Cryptographic Cores

You can change the allocation of cryptographic cores on Symmetric Multi-Processing (SMP) platforms to increase the throughput of Secure Client TLS/DTLS traffic. These changes can accelerate the SSL VPN datapath and provide customer-visible performance gains in Secure Client, smart tunnels, and port forwarding. These steps describe configuring the pool of cryptographic cores in either single or multiple context mode.

### Procedure

Specify how to allocate crypto accelerator processors:

**crypto engine accelerator-bias**

- **balanced**—Equally distributes cryptography hardware resources (Admin/SSL and IPsec cores).
- **ipsec**—Allocates cryptography hardware resources to favor IPsec (includes SRTP encrypted voice traffic).
- **ssl**—Allocates cryptography hardware resources to favor Admin/SSL. Use this bias when you support SSL-based Secure Client remote access VPN sessions.

**Example:**

```
hostname(config)# crypto engine accelerator-bias ssl
```

# Configure Dynamic Split Tunneling

With dynamic split tunneling, you can dynamically provision split exclude tunneling after tunnel establishment based on the host DNS domain name. Dynamic split tunneling is configured by creating a custom attribute and adding it to a group policy.

### Before you begin

To use this feature, you must have AnyConnect release 4.5 (or later). Refer to About Dynamic Split Tunneling for further explanation.

### Procedure

**Step 1** Define the custom attribute type in the WebVPN context with the following command:`anyconnect-custom-attr dynamic-split-exclude-domains description dynamic split exclude domains`

**Step 2** Define the custom attribute names for each cloud/web service that needs access by the client outside the VPN tunnel. For example, add Google_domains to represent a list of DNS domain names pertaining to Google web services. The attribute value contains the list of domain names to exclude from the VPN tunnel and must be comma-separated-values (CSV) format as the following: `anyconnect-custom-data dynamic-split-exclude-domains webex.com, webexconnect.com, tags.tiqcdn.com`

**Step 3** Attach the previously defined custom attribute to a certain policy group with the following command, executed in the group-policy attributes context: `anyconnect-custom dynamic-split-exclude-domains value webex_service_domains`

### What to do next

If split include tunneling is configured, a dynamic split exclusion is enforced only if at least one of the DNS response IP addresses is part of the split-include network. If there is no overlap between any of the DNS response IP addresses and any of the split-include networks, enforcing dynamic split exclusion is not necessary since traffic matching all DNS response IP addresses is already excluded from tunneling.

# Configure the Management VPN Tunnel

A management VPN tunnel ensures connectivity to the corporate network whenever the client system is powered up, not just when a VPN connection is established by the end user. You can perform patch management on out-of-the-office endpoints, especially devices that are infrequently connected by the user, via VPN, to the office network. Endpoint OS login scripts which require corporate network connectivity will also benefit from this feature.

The management VPN tunnel is meant to be transparent to the end user; therefore, network traffic initiated by user applications is not impacted, by default, but instead directed outside the management VPN tunnel.

If a user complains of slow logins, it may be an indication that the management tunnel was not configured appropriately. Refer to the Cisco Secure Client Administration Guide for additional requirements, incompatibilities, limitations, and troubleshooting of management VPN tunnel.

### Before you begin

Requires AnyConnect release 4.7 (or later)

### Procedure

**Step 1** Add the uploaded profile (profileMgmt) to the group policy (MgmtTunGrpPolicy) mapped to the tunnel group used by the management tunnel connection:

To indicate the profile is the AnyConnect Management VPN Profile, include **type vpn-mgmt** on the **anyconnect profiles** command. A regular AnyConnect VPN profile is type user.

```
group-policy MgmtTunGrpPolicy attributes
  webvpn
     anyconnect profiles value profileMgmt type vpn-mgmt
```

**Step 2** To deploy the management VPN profile through user tunnel connection, add the uploaded profile *(profileMgmt)* to the group policy *(DfltGrpPolicy)* mapped to the tunnel group used by the user tunnel connection:

```
group-policy DfltGrpPolicy attributes
  webvpn
     anyconnect profiles value profileMgmt type vpn-mgmt
```

# Viewing Active VPN Sessions

The following topics explain how to view VPN session information.

## Viewing Active Secure Client Sessions by IP Address Type

To view active Secure Client sessions using the command line interface, enter the **show vpn-sessiondb anyconnect filter p-ipversion** or **show vpn-sessiondb anyconnect filter a-ipversion** command in privileged EXEC mode.

- Display the active Secure Client sessions which are filtered by the endpoint's public IPv4 or IPv6 address. The public address is the address assigned to the endpoint by the enterprise.

  **show vpn-sessiondb anyconnect filter p-ipversion {v4 | v6}**

- Display the active Secure Client sessions which are filtered by the endpoint's assigned IPv4 or IPv6 address. The assigned address is the address assigned to the Secure Client by the ASA.

  **show vpn-sessiondb anyconnect filter a-ipversion {v4 | v6}**

### Example Output from show vpn-sessiondb anyconnect filter p-ipversion [v4 | v6] command

```
hostname(config)# show vpn-sessiondb anyconnect filter p-ipversion v4

Session Type: AnyConnect

Username     : user1                  Index        : 40
Assigned IP  : 192.168.17.10          Public IP    : 198.51.100.1
Protocol     : AnyConnect-Parent SSL-Tunnel
License      : AnyConnect Premium
Encryption   : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)RC4
Hashing      : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)SHA1
Bytes Tx     : 10570                  Bytes Rx     : 8085
Group Policy : GroupPolicy_SSLACCLIENT
Tunnel Group : SSLACCLIENT
Login Time   : 15:17:12 UTC Mon Oct 22 2012
Duration     : 0h:00m:09s
Inactivity   : 0h:00m:00s
```

```
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none
```

### Output from show vpn-sessiondb anyconnect filter a-ipversion [v4 | v6] command

```
hostname(config)# show vpn-sessiondb anyconnect filter a-ipversion v6

Session Type: AnyConnect

Username     : user1                  Index        : 45
Assigned IP  : 192.168.17.10
Public IP    : 2001:DB8:8:1:90eb:3fe5:9eea:fb29
Assigned IPv6: 2001:DB8:9:1::24
Protocol     : AnyConnect-Parent SSL-Tunnel
License      : AnyConnect Premium
Encryption   : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)RC4
Hashing      : AnyConnect-Parent: (1)none  SSL-Tunnel: (1)SHA1
Bytes Tx     : 10662                  Bytes Rx     : 17248
Group Policy : GroupPolicy_SSL_IPv6   Tunnel Group : SSL_IPv6
Login Time   : 17:42:42 UTC Mon Oct 22 2012
Duration     : 0h:00m:33s
Inactivity   : 0h:00m:00s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none
```

## Viewing Active LAN to LAN VPN Sessions by IP Address Type

To view active clientless SSL VPN sessions using the command line interface, enter the **show vpn-sessiondb l2l filter ipversion** command in privileged EXEC mode.

This command shows active lan to lan VPN sessions filtered by the connection's public IPv4 or IPv6 address.

The public address is the address assigned to the endpoint by the enterprise.

```
show vpn-sessiondb l2l filter ipversion {v4 | v6}
```

# About ISE Policy Enforcement

The Cisco Identity Services Engine (ISE) is a security policy management and control platform. It automates and simplifies access control and security compliance for wired, wireless, and VPN connectivity. Cisco ISE is primarily used to provide secure access and guest access, support bring your own device (BYOD) initiatives, and enforce usage policies in conjunction with Cisco TrustSec.

The ISE Change of Authorization (CoA) feature provides a mechanism to change the attributes of an authentication, authorization, and accounting (AAA) session after it is established. When a policy changes for a user or user group in AAA, CoA packets can be sent directly to the ASA from the ISE to reinitialize authentication and apply the new policy. An Inline Posture Enforcement Point (IPEP) is not required to apply access control lists (ACLs) for each VPN session established with the ASA.

ISE policy enforcement is supported on the following VPN clients:

- IPSec
- Secure Client

- L2TP/IPSec

**Note** Some policy elements such as Dynamic ACL (dACL) and Security Group Tag (SGT) are supported, whereas policy elements such as VLAN assignment and IP address assignment are not supported.

The system flow is as follows:

1. An end user requests a VPN connection.
2. The ASA authenticates the user to the ISE and receives a user ACL that provides limited access to the network.
3. An accounting start message is sent to the ISE to register the session.
4. Posture assessment occurs directly between the NAC agent and the ISE. This process is transparent to the ASA.
5. The ISE sends a policy update to the ASA via a CoA "policy push." This identifies a new user ACL that provides increased network access privileges.

**Note** Additional policy evaluations may occur during the lifetime of the connection, transparent to the ASA, via subsequent CoA updates.

This flow model differs from most scenarios that use RADIUS CoA. For wired/wireless 802.1x authentications, RADIUS CoA does not include any attributes. It only triggers the second authentication in which all attributes, such as DACL, are attached. For the ASA VPN posture, there is no second authentication. All of the attributes are returned in the RADIUS CoA. The VPN session is active and it is not possible to change most of the VPN user settings. The only settings that can be changed by a CoA activation are RedirectURL, RedirectACL, and Security Group Tag (SGT).

# Configure RADIUS Server Groups for ISE Policy Enforcement

To enable ISE policy assessment and enforcement, configure a RADIUS AAA server group for the ISE servers and add the servers to the group. When you configure the tunnel group for the VPN, you specify this server group for AAA services in the group.

### Procedure

**Step 1** Create the RADIUS AAA server group.

**aaa-server** *group_name* **protocol radius**

```
hostname(config)# aaa-server servergroup1 protocol radius
hostname(config-aaa-server-group)#
```

**Step 2** Enable the RADIUS dynamic authorization (CoA) services for the AAA server group.

**dynamic-authorization** [**port** *number*]

Specifying a port is optional. The default is 1700, the range is 1024 to 65535.

When you use the server group in a VPN tunnel, the RADIUS server group will be registered for CoA notification and the ASA will listen to the port for the CoA policy updates from ISE

```
hostname(config-aaa-server-group)# dynamic-authorization
```

**Step 3** If you do not want to use ISE for authentication, enable authorize-only mode for the RADIUS server group.

**authorize-only**

This indicates that when this server group is used for authorization, the RADIUS Access Request message will be built as an "Authorize Only" request as opposed to the configured password methods defined for the AAA server. If you do configure a common password using **radius-common-pw** command for the RADIUS server, it will be ignored.

For example, you would use authorize-only mode if you want to use certificates for authentication rather than this server group. You would still use this server group for authorization and accounting in the VPN tunnel.

```
hostname(config-aaa-server-group)# authorize-only
```

**Step 4** Enable the periodic generation of RADIUS interim-accounting-update messages.

**interim-accounting-update** [**periodic** [*hours*]]

ISE maintains a directory of active sessions based on the accounting records that it receives from NAS devices like the ASA. However, if ISE does not receive any indication that the session is still active (accounting message or posture transactions) for a period of 5 days, it will remove the session record from its database. To ensure that long-lived VPN connections are not removed, configure the group to send periodic interim-accounting-update messages to ISE for all active sessions.

- **periodic** [*hours*] enables the periodic generation and transmission of accounting records for every VPN session that is configured to send accounting records to the server group in question. You can optionally include the interval, in hours, for sending these updates. The default is 24 hours, the range is 1 to 120.
- (No parameters.) If you use this command without the **periodic** keyword, the ASA sends interim-accounting-update messages only when a VPN tunnel connection is added to a clientless VPN session. When this happens the accounting update is generated in order to inform the RADIUS server of the newly assigned IP address.

```
hostname(config-aaa-server-group)# interim-accounting-update periodic 12
```

**Step 5** (Optional.) Merge a downloadable ACL with the ACL received in the Cisco AV pair from a RADIUS packet.

**merge-dacl** {**before-avpair** | **after-avpair**}

This option applies only to VPN connections. For VPN users, ACLs can be in the form of Cisco AV pair ACLs, downloadable ACLs, and an ACL that is configured on the ASA. This option determines whether or not the downloadable ACL and the AV pair ACL are merged, and does not apply to any ACLs configured on the ASA.

The default setting is **no merge dacl**, which specifies that downloadable ACLs will not be merged with Cisco AV pair ACLs. If both an AV pair and a downloadable ACL are received, the AV pair has priority and is used.

The **before-avpair** option specifies that the downloadable ACL entries should be placed before the Cisco AV pair entries.

The **after-avpair** option specifies that the downloadable ACL entries should be placed after the Cisco AV pair entries.

```
hostname(config)# aaa-server servergroup1 protocol radius
hostname(config-aaa-server-group)# merge-dacl before-avpair
```

**Step 6** (Optional.) Specify the maximum number of requests sent to a RADIUS server in the group before trying the next server.

**max-failed-attempts** *number*

The range is from 1 and 5. The default is 3.

If you configured a fallback method using the local database (for management access only), and all the servers in the group fail to respond, then the group is considered to be unresponsive, and the fallback method is tried. The server group remains marked as unresponsive for a period of 10 minutes (by default), so that additional AAA requests within that period do not attempt to contact the server group, and the fallback method is used immediately. To change the unresponsive period from the default, see the **reactivation-mode** command in the next step.

If you do not have a fallback method, the ASA continues to retry the servers in the group.

```
hostname(config-aaa-server-group)# max-failed-attempts 2
```

**Step 7** (Optional.) Specify the method (reactivation policy) by which failed servers in a group are reactivated.

**reactivation-mode** {**depletion** [**deadtime** *minutes*] | **timed**}

Where:

- **depletion** [**deadtime** *minutes*] reactivates failed servers only after all of the servers in the group are inactive. This is the default reactivation mode. You can specify the amount of time, between 0 and 1440 minutes, that elapses between the disabling of the last server in the group and the subsequent reenabling of all servers. The default is 10 minutes.
- **timed** reactivates failed servers after 30 seconds of down time.

```
hostname(config-aaa-server-group)# reactivation-mode deadtime 20
```

**Step 8** (Optional.) Send accounting messages to all servers in the group.

**accounting-mode simultaneous**

To restore the default of sending messages only to the active server, enter the **accounting-mode single** command.

```
hostname(config-aaa-server-group)# accounting-mode simultaneous
```

**Step 9** Add the ISE RADIUS servers to the group.

**aaa-server** *group_name* [(*interface_name*)] **host** {*server_ip* | *name*} [*key*]

Where:

- *group_name* is the name of the RADIUS server group.
- (*interface_name*) is the name of the interface through which the server is reached. The default is (inside). The parentheses are required.
- **host** {*server_ip* | *name*} is the IP address or the hostname of the ISE RADIUS server.
- *key* is the optional key for encrypting the connection. You can more easily enter this key on the **key** command after entering the aaa-server-host mode. If you do not configure a key, the connection is not encrypted (plain text). The key is a case-sensitive, alphanumeric string of up to 127 characters that is the same value as the key on the RADIUS server.

You can add more than one server to the group.

```
hostname(config)# aaa-server servergroup1 (inside) host 10.1.1.3
hostname(config-aaa-server-host)# key sharedsecret
hostname(config-aaa-server-host)# exit
```

# Example Configurations for ISE Policy Enforcement

### Configure VPN Tunnel for ISE Dynamic Authentication with Passwords

The following example shows how to configure an ISE server group for dynamic authorization (CoA) updates and hourly periodic accounting. Included is the tunnel group configuration that configures password authentication with ISE.

```
ciscoasa(config)# aaa-server ise protocol radius
ciscoasa(config-aaa-server-group)# interim-accounting-update periodic 1
ciscoasa(config-aaa-server-group)# dynamic-authorization
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server ise (inside) host 10.1.1.3
ciscoasa(config-aaa-server-host)# key sharedsecret
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# tunnel-group aaa-coa general-attributes
ciscoasa(config-tunnel-general)# address-pool vpn
ciscoasa(config-tunnel-general)# authentication-server-group ise
ciscoasa(config-tunnel-general)# accounting-server-group ise
ciscoasa(config-tunnel-general)# exit
```

### Configure VPN Tunnel for ISE Authorization-Only

The following example shows how to configure a tunnel group for local certificate validation and authorization with ISE. Include the authorize-only command in the server group configuration, because the server group will not be used for authentication.

```
ciscoasa(config)# aaa-server ise protocol radius
```

```
ciscoasa(config-aaa-server-group)# authorize-only
ciscoasa(config-aaa-server-group)# interim-accounting-update periodic 1
ciscoasa(config-aaa-server-group)# dynamic-authorization
ciscoasa(config-aaa-server-group)# exit
ciscoasa(config)# aaa-server ise (inside) host 10.1.1.3
ciscoasa(config-aaa-server-host)# key sharedsecret
ciscoasa(config-aaa-server-host)# exit
ciscoasa(config)# tunnel-group aaa-coa general-attributes
ciscoasa(config-tunnel-general)# address-pool vpn
ciscoasa(config-tunnel-general)# authentication certificate
ciscoasa(config-tunnel-general)# authorization-server-group ise
ciscoasa(config-tunnel-general)# accounting-server-group ise
ciscoasa(config-tunnel-general)# exit
```

## Troubleshooting Policy Enforcement

The following commands can be used for debugging.

To trace CoA activity:

```
debug radius dynamic-authorization
```

To trace redirect URL functionality:

```
debug aaa url-redirect
```

To view NP classification rules corresponding to URL redirect functionality:

```
show asp table classify domain url-redirect
```

# Configure Advanced SSL Settings

The ASA uses the Secure Sockets Layer (SSL) protocol and the Transport Layer Security (TLS) to support secure message transmission for ASDM, Clientless SSL VPN, VPN, and browser-based sessions. The ASA supports the SSLv3, TLSv1, TLv1.1, TLSv1.2, and TLSv1.3 protocols for SSL-based VPN and management connections. In addition, DTLS is used for the AnyConnect VPN module of Cisco Secure Client connections.

The following ciphers are supported as noted:

| Cipher | TLSv1.1 / DTLS V1 | TLSV1.2 / DTLSV 1.2 | TLSv1.3 |
|---|---|---|---|
| TLS_AES_128_GCM_SHA256 | no | no | yes |
| TLS_CHACHA20_POLY1305_SHA256 | no | no | yes |
| TLS_AES_256_GCM_SHA384 | no | no | yes |
| AES128-GCM-SHA256 | no | yes | no |
| AES128-SHA | yes | yes | no |
| AES128-SHA256 | no | yes | no |

| Cipher | TLSv1.1 / DTLS V1 | TLSV1.2 / DTLSV 1.2 | TLSv1.3 |
|---|---|---|---|
| AES256-GCM-SHA384 | no | yes | no |
| AES256-SHA | yes | yes | no |
| AES256-SHA256 | no | yes | no |
| DERS-CBC-SHA | no | no | no |
| DES-CBC-SHA | yes | yes | no |
| DHE-RSA-AES128-GCM-SHA256 | no | yes | no |
| DHE-RSA-AES128-SHA | yes | yes | no |
| DHE-RSA-AES128-SHA256 | no | yes | no |
| DHE-RSA-AES256-GCM-SHA384 | no | l | no |
| DHE-RSA-AES256-SHA | yes | yes | no |
| ECDHE-ECDSA-AES128-GCM-SHA256 | no | yes | no |
| ECDHE-ECDSA-AES128-SHA256 | no | yes | no |
| ECDHE-ECDSA-AES256-GCM-SHA384 | no | yes | no |
| ECDHE-ECDSA-AES256-SHA384 | no | yes | no |
| ECDHE-RSA-AES128-GCM-SHA256 | yes | yes | no |
| ECDHE-RSA-AES128-SHA256 | no | yes | no |
| ECDHE-RSA-AES256-GCM-SHA384 | no | yes | no |
| ECDHE-RSA-AES256-SHA384 | no | yes | no |
| NULL-SHA | no | no | no |
| RC4-MD5 | no | no | no |
| RC4-SHA | no | no | no |



**Note** For Release 9.4(1), all SSLv3 keywords have been removed from the ASA configuration, and SSLv3 support has been removed from the ASA. If you have SSLv3 enabled, a boot-time error will appear from the command with the SSLv3 option. The ASA will then revert to the default use of TLSv1.

The Citrix mobile receiver may not support TLS 1.1/1.2 protocols; see https://www.citrix.com/content/dam/citrix/en_us/documents/products-solutions/citrix-receiver-feature-matrix.pdf for compatibility

To specify the minimum protocol version for which the ASA will negotiate SSL/TLS and DTLS connections, perform the following steps:

### Procedure

**Step 1** Set the minimum protocol version for which the ASA will negotiate a connection.

**ssl server-version** [**tlsv1** | **tlsv1.1** | **tlsv1.2** | **tlsv1.3**] [**dtlsv1** | **dtlsv1.2**]

Where:

- **tlsv1**—Enter this keyword to accept SSLv2 ClientHellos and negotiate TLSv1 (or greater)
- **tlsv1.1**—Enter this keyword to accept SSLv2 ClientHellos and negotiate TLSv1.1 (or greater)
- **tlsv1.2**—Enter this keyword to accept SSLv2 ClientHellos and negotiate TLSv1.2 (or greater)
- **tlsv1.3**—Enter this keyword to accept SSLv2 ClientHellos and negotiate TLSv1.3 (or greater)
- **dtlsv1**—Enter this keyword to accept DTLSv1 ClientHellos and negotiate DTLSv1 (or greater)
- **dtlsv1.2**—Enter this keyword to accept DTLSv1.2 ClientHellos and negotiate DTLSv1.2 (or greater)

**Note**
The configuration and use of DTLS applies to Secure Client remote access connections only.

Ensure the TLS session is as secure, or more secure than the DTLS session by using an equal or higher version of TLS than DTLS. Given this, tlsv1.2 is the only acceptable TLS version when choosing dtls1.2; and any TLS version can be used with dtls1 since they are all equal to or greater than DTLS 1.0.

**Example:**

Examples:

```
hostname(config)# ssl server-version tlsv1.1

hostname(config)# ssl server-version tlsv1.2 dtlsv1.2
```

**Step 2** Specify the maximum version of SSL/TLS protocol that ASA uses when acting as a server.

**ssl server-max-version** [**tlsv1** | **tlsv1.1** | **tlsv1.2** | **tlsv1.3**]

If the server-max-version is configured as TLSV1.2, you won't be able to configure TLSV1.3 as the server-version.

**Step 3** Specify the SSL/TLS protocol version that the ASA uses when acting as a client.

**ssl client-version** [**tlsv1** | **tlsv1.1** | **tlsv1.2** | **tlsv1.3**]

Where:

- **tlsv1**—Enter this keyword to specify that the ASA transmits TLSv1 client hellos and negotiates TLSv1 (or greater).
- **tlsv1.1**—Enter this keyword to specify that the ASA transmits TLSv1.1 client hellos and negotiates TLSv1.1 (or greater).

- **tlsv1.2**—Enter this keyword to specify that the ASA transmits TLSv1.2 client hellos and negotiates TLSv1.2 (or greater).
- **tlsv1.3**—Enter this keyword to specify that the ASA transmits TLSv1.3 client hellos and negotiates TLSv1.3 (or greater).

DTLS is not available for SSL client role.

**Example:**

Examples:

```
hostname(config)# ssl client-version tlsv1
```

**Step 4** Specify the maximum version of SSL/TLS protocol that ASA uses when acting as a client.

**ssl client-max-version** [**tlsv1** | **tlsv1.1** | **tlsv1.2** | **tlsv1.3**]

If the client-max-version is configured as TLSV1.2, you won't be able to configure TLSV1.3 as the client-version.

**Step 5** Specify the encryption algorithms for the SSL, DTLS, and TLS protocols.

**ssl cipher** *version* [ *level* | **custom** *string*]

Where:

- The *version* argument specifies the SSL, DTLS, or TLS protocol version. Supported versions include:
  - default—The set of ciphers for outbound connections.
  - dtlsv1—The ciphers for DTLSv1 inbound connections.
  - dtlsv1.2—The ciphers for DTLSv1.2 inbound connections.
  - tlsv1—The ciphers for TLSv1 inbound connections.
  - tlsv1.1—The ciphers for TLSv1.1 inbound connections.
  - tlsv1.2—The ciphers for TLSv1.2 inbound connections.
  - tlsv1.3—The ciphers for TLSv1.3 inbound connections.
- The *level* argument specifies the strength of the ciphers and indicates the minimum level of ciphers that are configured. Valid values in increasing order of strength are:
  - all—Includes all ciphers.
  - low—Includes all ciphers except NULL-SHA.
  - medium (this is the default for all protocol versions)—Includes all ciphers (except NULL-SHA, DES-CBC-SHA, RC4-MD5, RC4-SHA, and DES-CBC3-SHA).
  - fips—Includes all FIPS-compliant ciphers (except NULL-SHA, DES-CBC-SHA, RC4-MD5, RC4-SHA, and DES-CBC3-SHA).
  - high(applies only to TLSv1.2 and TLSv1.3)—Includes only AES-256 with SHA-2 ciphers for TLSv1.2. The strength of all TLSv1.3 ciphers are high.

- Specifying the **custom** *string* option allows you to have full control of the cipher suite using OpenSSL cipher definition strings. For more information, see https://www.openssl.org/docs/apps/ciphers.html.

The recommended setting is **medium**. Using **high** may limit connectivity. Using custom may limit functionality if there are only a few ciphers configured. Restricting the default custom value limits outbound connectivity, including clustering.

The ASA specifies the order of priority for supported ciphers. See the command reference for more information.

This command replaces the ssl encryption command, which has been deprecated starting with Version 9.3(2).

**Step 6** Allow multiple trustpoints on a single interface.

**ssl trust-point** *name* [ [**interface** *vpnlb-ip* ] | [**domain** *domain-name*]

```
hostname(config)# ssl trust-point www-cert domain www.example.com
```

The **name** argument specifies the name of the trustpoint. The **interface** argument specifies the name of the interface on which a trustpoint is configured. The vpnlb-ip keyword applies only to interfaces and associates this trustpoint with the VPN load-balancing cluster IP address on this interface. The **domain***domain-name* keyword-argument pair specifies a trustpoint that is associated with a particular domain name that is used to access the interface.

You may configure a maximum of 16 trustpoints per interface.

If you do not specify an interface or domain, this command creates the fallback trustpoint for all interfaces that do not have a trustpoint configured.

If you enter the **ssl trustpoint ?** command, the available configured trustpoints appear. If you enter the **ssl trust-point** *name* **?** command (for example, **ssl trust-point mysslcert ?**), the available configured interfaces for the trustpoint-SSL certificate association appear.

Observe these guidelines when using this command:

- The value for trustpoint must be the name of the CA trustpoint as configured in the **crypto ca trustpoint name** command.
- The value for interface must be the nameif name of a previously configured interface.
- Removing a trustpoint also removes any **ssl trust-point** entries that reference that trustpoint.
- You can have one ssl trust-point entry for each interface and one that specifies no interfaces.
- You can reuse the same trustpoint for multiple entries.
- A trustpoint configured with the domain keyword may apply to multiple interfaces (depending on how you connect).
- You may only have one **ssl trust-point** per *domain-name* value.
- If the following error appears after you enter this command:

```
error:0B080074:x509 certificate routines:X509_check_private_key:key values
mismatch@x509_cmp.c:339
```

  It means that a user has configured a new certificate to replace a previously configured certificate. No action is required.

- The certificates are chosen in the following order:

- If a connection matches the value of the **domain** keyword, that certificate is chosen first. (**ssl trust-point***name***domain***domain-name* command)
- If a connection is made to the load-balancing address, the vpnlb-ip certificate is chosen. (**ssl trust-point** *name* **interface** *vpnlb-ip* command)
- The certificate configured for the interface. (**ssl trust-point** *name* interface command)
- The default certificate not associated with an interface. (**ssl trust-point** *name*)
- The ASA's self-signed, self-generated certificate.

**Step 7** Specify the DH group to be used with DHE-RSA ciphers that are used by TLS.

```
ssl dh-group [group14 | group15]
```

```
hostname(config)# ssl dh-group group14
```

The group14 and 15 keyword configures DH group 14 (2048-bit modulus, 224-bit prime order subgroup).

Group 14 is not compatible with Java 7. All groups are compatible with Java 8. Groups 14 is FIPS-compliant. The default value is ssl dh-group group14.

**Step 8** Specify the group to be used with ECDHE-ECDSA ciphers that are used by TLS.

```
ssl ecdh-group [group19 | group20 | group21]
```

```
hostname(config)# ssl ecdh-group group20
```

The group19 keyword configures group 19 (256-bit EC). The group20 keyword configures group 20 (384-bit EC). The group21 keyword configures group 21 (521-bit EC).

The default value is ssl ecdh-group group19.

**Note**
ECDSA and DHE ciphers are the highest priority.

### What to do next

You can use the use the following commands to view the TLS/DTLS configuration:

- **show run ssl** if it is not the default TLS/DTLS version.
- **show run ssl all** if it is the default TLS/DTLS version.

# Rate Limit for Preauthenticated SSL Connections

Preauthenticated SSL connections are secure connections that are established before a user or device has completed the authentication process. ASA can receive multiple preauthenticated SSL connections that can increase the memory usage of the device, prevent new SSL connections, make the device unresponsive, and impact performance.

ASA can rate-limit these preauthenticated SSL connections. This limit is calculated as 3 times the VPN connection limit of the device. This feature is enabled by default and is supported only on ASA virtual. For example, for an ASAv10, the VPN connection limit is 250. The preauthentication limit of SSL connections is 3 x 250 = 750.

For a device, when the preauthenticated connection limit exceeds, no new SSL connections are allowed. However, this restriction is not valid for management connections. The device allows new SSL connections only after the preauthenticated SSL connections count becomes zero.

### Supported Devices

- ASAv5
- ASAv10
- ASAv30
- ASAv50
- ASAv100

### Syslog

The device generates syslog 725025 when it reaches the rate-limit threshold for the number of preauthenticated SSL connections. This syslog is generated when the number of preauthenticated SSL connections is high (90% of the limit) and when it is low (70% of the limit). The syslog is rate-limited to one syslog every 10 seconds.

### Counters

Three new counters provide information about the rate-limited preauthenticated SSL connections:

- SOCK_PRE_AUTH_COUNT_EXCEEDED: An increment by one indicates that the number of simultaneous preauthenticated SSL connections has exceeded the VPN limit. New connection attempts are blocked until the SOCK_PRE_AUTH_COUNT counter is 0.
- SOCK_COUNT_RATE_LIMIT: Indicates the connections that the ASA has reset after exceeding the rate limit.
- SOCK_PRE_AUTH_COUNT: Number of concurrent preauthentication connections at any given time.

Use the **show counters** command to view the details of these counters.

### Debug Commands

As debugging output is assigned high priority in the CPU process, it can render the system unusable. We recommend that you use debug commands only during troubleshooting sessions with Cisco TAC.

You can use the following debug commands for this feature:

- **debug menu ctm 1205**: Disable or re-enable the feature.
- **debug menu ctm 1206**: Verify the current status of the feature.

```
vASA# debug menu ctm 1205
Disabling SSL pre-auth connection rate monitoring
```

```
vASA# debug menu ctm 1206
SSL PRE-AUTH connection rate monitoring is Disabled
```

# Persistent IPsec Tunneled Flows

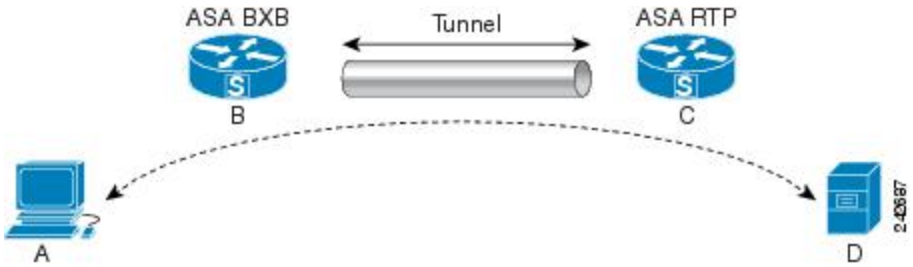
In networks running a version of ASA software prior to Release 8.0.4, existing IPsec LAN-to-LAN or Remote-Access TCP traffic flows going through an IPSec tunnel are dropped when the tunnel drops. The flows are recreated as needed when and if the tunnel comes back up. This policy works well from the resource-management and security standpoints. However, there are cases in which such behavior introduces issues for users, particularly for those migrating from PIX to ASA-only environments and for legacy TCP applications that do not restart easily or in networks that include gateways that tend to drop tunnels frequently.(See CSCsj40681 and CSCsi47630 for details.)

The persistent IPsec tunneled flows feature addresses this issue. With this feature enabled, the ASA preserves and resumes stateful (TCP) tunneled flows. All other flows are dropped when the tunnel drops and must reestablish when a new tunnel comes up.

**Note** This feature supports IPsec LAN-to-LAN tunnels and IPsec Remote-Access tunnels running in Network-Extension Mode. It does not support IPsec or AnyConnect/SSL VPN remote access tunnels.

The following example shows how the persistent IPsec tunneled flows feature works.

***Figure 5: Network Scenario***




In this example the BXB and RTP networks are connected through a secure LAN-to-LAN tunnel by a pair of security appliances. A PC in the BXB network is executing an FTP transfer from a server in the RTP network through the secure tunnel. In this scenario, assume that for some reason the tunnel drops after the PC has logged into the server and started the transfer. Although the tunnel is be reestablished since the data is still attempting to flow, the FTP transfer will not complete. The user must terminate the transfer and start over by logging back into the server. However, if persistent IPsec tunnel flows is enabled, as long as the tunnel is recreated within the timeout interval, the data continues to flow successfully through the new tunnel because the security appliances retain the history (state information) for this flow.

### Scenario

The following sections describe the data flow situations for a dropped and recovered tunnel, first with the persistent IPsec tunneled flows feature disabled, then with the feature enabled. In both of these cases, see the preceding figure for an illustration of the network. In this illustration:

- Flow B-C defines the tunnel and carries the encrypted ESP data.

- Flow A-D is the TCP connection for the FTP transfer and traverses the tunnel defined by flow B-C. This flow also contains state information used by the firewall to inspect the TCP/FTP flow. The state information is vital and is constantly updated by the firewall as the transfer progresses.

**Note** The reverse flows in each direction are omitted for simplicity.

#### Disabled Persistent IPsec Tunneled Flows

When the LAN-2-LAN tunnel drops, both flow A-D and flow B-C and any state information belonging to them are deleted. Subsequently, the tunnel is reestablished, and flow B-C is recreated and is able to resume carrying tunneled data. But the TCP/FTP flow A-D runs into trouble. Because the state information describing the flow up to this point in the FTP transfer has been deleted, the stateful firewall blocks the in-flight FTP data and rejects the flow A-D creation. Having lost the history of this flow ever existing, the firewall treats the FTP transfer as stray TCP packets and drops them. This is the default behavior.

#### Enabled Persistent IPsec Tunneled Flows

With the persistent IPsec tunneled flows feature enabled, as long as the tunnel is recreated within the timeout window, data continues flowing successfully because the ASA still has access to the state information in flow A-D.

With this feature enabled, the ASA treats the flows independently. This means that flow A-D is not deleted when the tunnel defined by flow B-C is dropped. The ASA preserves and resumes stateful (TCP) tunneled flows. All other flows are dropped and must reestablish on the new tunnel. This does not weaken the security policy for tunneled flows, because the ASA drops any packets arriving on flow A-D while the tunnel is down.

Tunneled TCP flows are not dropped, so they rely on the TCP timeout for cleanup. However, if the timeout is disabled for a particular tunneled flow, that flow remains in the system until being cleared manually or by other means (for example, by a TCP RST from the peer).

# Configure Persistent IPsec Tunneled Flows Using CLI

Configuration Example

# Troubleshooting Persistent IPsec Tunneled Flows

Both the **show asp table** and the **show conn** commands can be useful in troubleshooting issues with persistent IPsec tunneled flows.

## Is the Persistent IPsec Tunneled Flows Feature Enabled?

To see whether a particular tunnel has this feature enabled, look at the VPN context associated with the tunnel using the **show asp table** command. The show asp table vpn-context command displays a “+PRESERVE” flag for each context that maintains stateful flows after the tunnel drops, as shown in the following example (bolding added for legibility):

```
hostname(config)# show asp table vpn-context
VPN CTX=0x0005FF54, Ptr=0x6DE62DA0, DECR+ESP+PRESERVE, UP, pk=0000000000, rk=0000000000,
gc=0
```

```
VPN CTX=0x0005B234, Ptr=0x6DE635E0, ENCR+ESP+PRESERVE, UP, pk=0000000000, rk=0000000000,
gc=0

------------------------------------------------------------------------------

hostname(config)# show asp table vpn-context detail

VPN CTX  = 0x0005FF54

Peer IP  = ASA_Private
Pointer  = 0x6DE62DA0
State    = UP
Flags    = DECR+ESP+PRESERVE
SA       = 0x001659BF
SPI      = 0xB326496C
Group    = 0
Pkts     = 0
Bad Pkts = 0
Bad SPI  = 0
Spoof    = 0
Bad Crypto = 0
Rekey Pkt  = 0
Rekey Call = 0

VPN CTX  = 0x0005B234

Peer IP  = ASA_Private
Pointer  = 0x6DE635E0
State    = UP
Flags    = ENCR+ESP+PRESERVE
SA       = 0x0017988D
SPI      = 0x9AA50F43
Group    = 0
Pkts     = 0
Bad Pkts = 0
Bad SPI  = 0
Spoof    = 0
Bad Crypto = 0
Rekey Pkt  = 0
Rekey Call = 0
hostname(config)#
Configuration and Restrictions
This configuration option is subject to the same CLI configuration restrictions as other
sysopt VPN CLI.
```

## Locating Orphaned Flows

If a LAN-to-LAN/Network-Extension-Mode tunnel drops and does not recover before the timeout, there might be a number of orphaned tunnel flows. These flows are not torn down as a result of the tunnel going down, but all the data attempting to flow through them is dropped. To see these flows, use the **show conn** command, as in the following examples (bolding added for emphasis and to show user input):

```
asa2(config)# show conn detail
9 in use, 14 most used
Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,
       B - initial SYN from outside, C - CTIQBE media, D - DNS, d - dump,
       E - outside back connection, F - outside FIN, f - inside FIN,
       G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,
       i - incomplete, J - GTP, j - GTP data, K - GTP t3-response
       k - Skinny media, M - SMTP data, m - SIP media, n - GUP
       O - outbound data, P - inside back connection, p - Phone-proxy TFTP connection,
       q - SQL*Net data, R - outside acknowledged FIN,
```

```
R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,
s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,
V - VPN orphan, W - WAAS,
X - inspected by service module
```

The following example shows sample output from the **show conn** command when an orphan flow exists, as indicated by the **V** flag:

```
hostname# show conn
16 in use, 19 most used
TCP out 192.168.110.251:7393 in 192.168.150.252:21 idle 0:00:00 bytes 1048 flags UOVB
TCP out 192.168.110.251:21137 in 192.168.150.252:21 idle bytes 1048 flags UIOB
```

To limit the report to those connections that have orphan flows, add the **vpn_orphan** option to the **show conn state** command, as in the following example:

```
hostname# show conn state vpn_orphan
14 in use, 19 most used
TCP out 192.168.110.251:7393 in 192.168.150.252:5013 idle 0:00:00 bytes 2841019 flags UOVB
```

# Troubleshooting Using Crypto Archives

### About Crypto Archives

Crypto issues are difficult to triage. Crypto archives help you to troubleshoot these issues. A crypto archive contains crypto session information about the crypto request, peer information, the component that sent the crypto request, and the timed-out crypto session information. ASA does not save keys and initialization vectors (IVs) for the session. For SSL and IPsec, you can also view the following information:

- For SSL: Session SSL version, source, destination IP addresses, and ports.
- For IPsec: IPsec security association information.

A ring can hold 2000 crypto command entries. ASA pushes the crypto command in one of the rings and pulls out the result after completing the crypto request. Crypto archive files now have the timed-out crypto request's ring and entry index. The ring and its entry index help in troubleshooting problematic crypto commands.

The crypto archive has two formats: a text file and a binary file. You can use the **debug menu ctm 103** command to decode the binary file.

For example:

```
ASA# debug menu ctm 103 crypto_eng0_arch_4.bin
[Nitrox V Archive Header v1.0 Info]
ASA Image Version: PIX (9.20) #0: Tue Mar 29 16:20:30 GMT 2022
...
SE SSL microcode: CNN5x-MC-SE-SSL-0011
AE microcode: CNN5x-MC-AE-MAIN-0002
Crypto Engine 0
Crash type: SE Ring Timeout
...
Core Soft Resets: 11
...
Timeout Ring (SE): 12
Timeout Entry: 642
SE TIMEOUT:
Core SE 6 Touts: 2
Core SE 8 Touts: 2
```

```
Core SE 12 Touts: 4
Core SE 32 Touts: 2
Core SE 37 Touts: 1
.....
[Timeout Session Info]
Active: TRUE
Sync: FALSE
Callback: TRUE
Saved Callback: FALSE
Commands in progress: 1
Engine : hardware
Device : n5 (Nitrox V)
Session : ssl
Priority: normal
NP VPN context handle : 0x00000000
Flag : 0
vcid : 0
Block size : 2050
async cb ring index: 0
tls offload rsa: FALSE
Session context:
SSL Version : dtls1.2
SSL Context Type : handshake
Encryption Mode : gcm
Auth Algorithm : null
Hash Algorithm : none
Key Size : 32
SSL V : dtls1.2
Source IP : 82.1.2.2
Source Port : 51915
Dest IP : 82.29.155.32
Dest Port : 443
```

In the above example, the highlighted information shows the timeout ring, the crash time (timeout entry), and SSL session information.

### Supported Devices for Crypto Archives

The following devices with Nitrox V crypto accelerator support crypto archives:

- Cisco Firepower 3105, 3110, 3120, 3130, 3140
- Cisco Firepower 4112, 4115, 4125, 4145
- Cisco Firepower 9300 SM-40, SM-48 and SM-56

# Using SSL Counters

You can use SSL counters to view SSL tunnel information and logs. Information about state machine transitions around connection establishment, additional states, and details are available to aid debugging.

The **debug ssl state** command provides the following information:

- Remote device and interface with associated IPs, ports, and protocol.
- Debugs for SSL connection establishment errors.
- Debugs for verifying decrypted data padding length.

Use the **show counters** command to view the SSL counters. From Version 9.20.1, more SSL counters are available for debugging such as:

- CNT_SSL_NP_CP_EVENT_NULL
- CNT_SSL_NP_CP_EVENT_ENQUEUE_ERR
- CNT_SSL_NP_CP_EVENT_RELEASE
- CNT_SSL_NP_SNP_FLOW_HNDSHK_FAIL
- CNT_SSL_NP_HDL_LOCK_RELEASE
- CNT_SSL_NP_VERIFY_PADDING
- CNT_SSL_NP_MAX_PAD_LEN_EXCEEDED
- CNT_SSL_NP_NO_CIPHERS_COMPATIBLE
- CNT_SSL_NP_CIPHER_LIC_NOT_GOOD

# How to Remove Stuck ASP Table Entries

In Version 9.19.1 and earlier, if there are stuck ASP encrypt rules, you have to reboot the device. In Version 9.20.1 and later, you can remove the stuck encrypt rules from the ASP table using the **debug menu asp 100 <encrypt_rule_id>** command without rebooting the device. Use the **show asp table classify domain encrypt** command to find the *encrypt_rule_id*.

### Guidelines

- As debugging output is assigned high priority in the CPU process, it can render the system unusable. We recommend that you use this debug command only during troubleshooting sessions with Cisco TAC.
- There is no validation to check if the rule is stuck. The device will crash if the system tries to remove a rule that was previously removed using this command.
- The command ID parameter must exactly match an actual ID from the ASP table.

### Example

In the following example, traffic gets dropped when traffic hits the stuck rule **0x7f039846aaaa** instead of hitting the good rule **0x7f039846bbbb**. You can identify the stuck rule from the hit counts. The hit count for the stuck rule is 9999 while that of the good rule is 0.

1. Use the **show asp table classify domain encrypt** command to view the ASP rules.

```
ASAv(config)# show asp table classify domain encrypt
...
out id=0x7f039846aaaa, priority=70, domain=encrypt, deny=false
hits=9999, user_data=0xaaaa, cs_id=0x7f03941866e0, reverse, flags=0x0, protocol=0
src ip/id=1.0.0.0, mask=255.0.0.0, port=0, tag=any
dst ip/id=2.0.0.0, mask=255.0.0.0, port=0, tag=any
src nsg_id=none, dst nsg_id=none
dscp=0x0, input_ifc=any, output_ifc=outside
out id=0x7f039846bbbb, priority=70, domain=encrypt, deny=false //this is a good rule
hits=0, user_data=0xbbbb, cs_id=0x7f03941866e0, reverse, flags=0x0, protocol=0
src ip/id=1.0.0.0, mask=255.0.0.0, port=0, tag=any
```

```
dst ip/id=2.0.0.0, mask=255.0.0.0, port=0, tag=any
src nsg_id=none, dst nsg_id=none
dscp=0x0, input_ifc=any, output_ifc=outside
...
```

2. Use the **debug menu asp 100 <encrypt_rule_id>** command to remove the stuck encrypt rules from the ASP table

```
ASAv(config)# debug menu asp 100 id=0x7f039846aaaa
Encrypt rule 0x7f0398469510 was successfully deleted.
```

3. Use the **show asp table classify domain encrypt** command to verify if ASA has removed the stuck ASP rule.

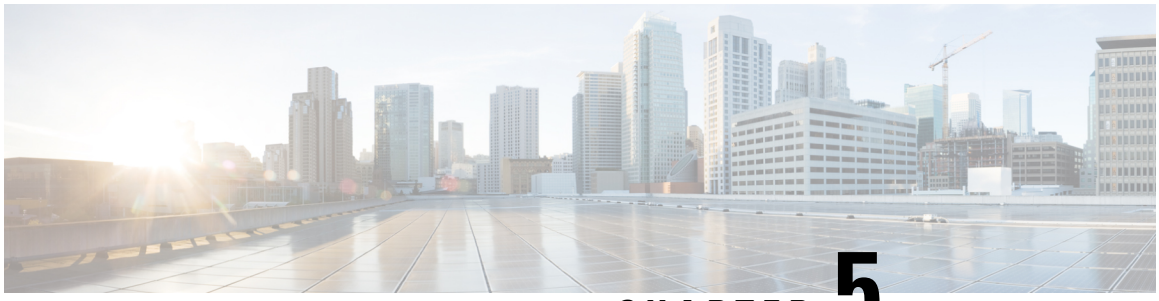
```
ASAv(config)# show asp table classify domain encrypt
...
out id=0x7f039846bbbb, priority=70, domain=encrypt, deny=false //now this rule has hits
hits=10, user_data=0xbbbb, cs_id=0x7f03941866e0, reverse, flags=0x0, protocol=0
src ip/id=1.0.0.0, mask=255.0.0.0, port=0, tag=any
dst ip/id=2.0.0.0, mask=255.0.0.0, port=0, tag=any
src nsg_id=none, dst nsg_id=none
dscp=0x0, input_ifc=any, output_ifc=outside
```

# Clear WebVPN Configurations from ASA

When you use **no webvpn** and **clear configure webvpn** commands, the default WebVPN configurations are not removed. The **http_in** and **http_out** counters are retained to collate compression statistics.

To clear the WebVPN configurations from ASA, do one of the following:

- Disable the compression stats after a bootup using the **no compression all** command.
- Clear the compression statistics counters using the **clear compression all** command.



CHAPTER 5

# Connection Profiles, Group Policies, and Users

This chapter describes how to configure VPN connection profiles (formerly called “tunnel groups”), group policies, and users. This chapter includes the following sections.


- Overview of Connection Profiles, Group Policies, and Users, on page 99
- Connection Profiles, on page 100
- Configure Connection Profiles, on page 104
- Group Policies, on page 130
- Use of a Zone Labs Integrity Server, on page 169
- Configure User Attributes, on page 175
- Best Practices for Configuring and Adjusting VPN Filter ACL, on page 184


## Overview of Connection Profiles, Group Policies, and Users

Groups and users are core concepts in managing the security of virtual private networks (VPNs) and in configuring the ASA. They specify attributes that determine user access to and use of the VPN. A *group* is a collection of users treated as a single entity. *Users* get their attributes from *group policies*. *A connection profile* identifies the group policy for a specific connection. If you do not assign a particular group policy to a user, the default group policy for the connection applies.

In summary, you first configure connection profiles to set the values for the connection. Then you configure group policies. These set values for users in the aggregate. Then you configure users, which can inherit values from groups and configure certain values on an individual user basis. This chapter describes how and why to configure these entities.

**Note** You configure connection profiles using **tunnel-group** commands. In this chapter, the terms “connection profile” and “tunnel group” are often used interchangeably.

Connection profiles and group policies simplify system management. To streamline the configuration task, the ASA provides a default LAN-to-LAN connection profile (DefaultL2Lgroup), a default remote access connection profile for IKEv2 VPN (DefaultRAgroup), a default connection profile for Clientless SSL and Secure Client SSL connections (DefaultWEBVPNgroup), and a default group policy (DfltGrpPolicy). The default connection profiles and group policy provide settings are likely to be common for many users. As you add users, you can specify that they “inherit” parameters from a group policy. Thus you can quickly configure VPN access for large numbers of users.

If you decide to grant identical rights to all VPN users, then you do not need to configure specific connection profiles or group policies, but VPNs seldom work that way. For example, you might allow a finance group to access one part of a private network, a customer support group to access another part, and an MIS group to access other parts. In addition, you might allow specific users within MIS to access systems that other MIS users cannot access. Connection profiles and group policies provide the flexibility to do so securely.



**Note** The ASA also includes the concept of object groups, which are a superset of network lists. Object groups let you define VPN access to ports as well as networks. Object groups relate to ACLs rather than to group policies and connection profiles. For more information about using object groups, see Chapter 20, "Objects" in the general operations configuration guide.

The security appliance can apply attribute values from a variety of sources. It applies them according to the following hierarchy:

1. Dynamic Access Policy (DAP) record
2. Username
3. Group policy
4. Group policy for the connection profile
5. Default group policy

Therefore, DAP values for an attribute have a higher priority than those configured for a user, group policy, or connection profile.

When you enable or disable an attribute for a DAP record, the ASA applies that value and enforces it. For example, when you disable HTTP proxy in dap webvpn configuration mode, the ASA looks no further for a value. When you instead use the no value for the http-proxy command, the attribute is not present in the DAP record, so the security appliance moves down to the AAA attribute in the username, and if necessary, to the group policy and finds a value to apply. The ASA clientless SSL VPN configuration supports only one **http-proxy** and one **https-proxy** command each. We recommend that you use ASDM to configure DAP.

# Connection Profiles

A connection profile consists of a set of records that determines tunnel connection policies. These records identify the servers to which the tunnel user is authenticated, as well as the accounting servers, if any, to which connection information is sent. They also identify a default group policy for the connection, and they contain protocol-specific connection parameters. Connection profiles include a small number of attributes that pertain to creating the tunnel itself. Connection profiles include a pointer to a group policy that defines user-oriented attributes.

The ASA provides the following default connection profiles: DefaultL2Lgroup for LAN-to-LAN connections, DefaultRAgroup for IPSEC remote access connections, and DefaultWEBVPNGroup for SSL VPN (browser-based and Secure Client based) connections. You can modify these default connection profiles, but you cannot delete them. You can also create one or more connection profiles specific to your environment. Connection profiles are local to the ASA and are not configurable on external servers.

**Note** Some profiles (such as IKEv1 in phase 1) may be unable to determine whether an endpoint is remote access or LAN-to-LAN. If it cannot determine the tunnel group, it defaults to

```
tunnel-group-map default-group <tunnel-group-name>
```

(default is *DefaultRAGroup*).

# General Connection Profile Connection Parameters

General parameters are common to all VPN connections. The general parameters include the following:

- Connection profile name—You specify a connection-profile name when you add or edit a connection profile. The following considerations apply:
  - For clients that use preshared keys to authenticate, the connection profile name is the same as the group name that a client passes to the ASA.
  - Clients that use certificates to authenticate pass this name as part of the certificate, and the ASA extracts the name from the certificate.
- Connection type—Connection types include IKEv1 remote-access, IPsec LAN-to-LAN, and AnyConnect (SSL/IKEv2). A connection profile can have only one connection type.
- Authentication, Authorization, and Accounting servers—These parameters identify the server groups or lists that the ASA uses for the following purposes:
  - Authenticating users
  - Obtaining information about services users are authorized to access
  - Storing accounting records

A server group can consist of one or more servers.

- Default group policy for the connection—A group policy is a set of user-oriented attributes. The default group policy is the group policy whose attributes the ASA uses as defaults when authenticating or authorizing a tunnel user.
- Client address assignment method—This method includes values for one or more DHCP servers or address pools that the ASA assigns to clients.
- Password management—This parameter lets you warn a user that the current password is due to expire in a specified number of days (the default is 14 days), then offer the user the opportunity to change the password.
- Strip group and strip realm—These parameters direct the way the ASA processes the usernames it receives. They apply only to usernames received in the form user@realm.

  A realm is an administrative domain appended to a username with the @ delimiter (user@abc). If you strip the realm, the ASA uses the username and the group (if present) for authentication. If you strip the group, the ASA uses the username and the realm (if present) for authentication.

  Enter the strip-realm command to remove the realm qualifier, and enter the strip-group command to remove the group qualilfier from the username during authentication. If you remove both qualifiers,

authentication is based on the *username* alone. Otherwise, authentication is based on the full *username@realm* or *username*<delimiter> *group* string. You must specify strip-realm if your server is unable to parse delimiters.

In addition, for L2TP/IPsec clients only, when you specify the strip-group command the ASA selects the connection profile (tunnel group) for user connections by obtaining the group name from the username presented by the VPN client.

- Authorization required—This parameter lets you require authorization before a user can connect, or turn off that requirement.
- Authorization DN attributes—This parameter specifies which Distinguished Name attributes to use when performing authorization.

# IPsec Tunnel-Group Connection Parameters

IPsec parameters include the following:

- A client authentication method: preshared keys, certificates, or both.
  - For IKE connections based on preshared keys, this is the alphanumeric key itself (up to 128 characters long), associated with the connection policy.
  - Peer-ID validation requirement—This parameter specifies whether to require validating the identity of the peer using the peer's certificate.
  - If you specify certificates or both for the authentication method, the end user must provide a valid certificate in order to authenticate.
- An extended hybrid authentication method: XAUTH and hybrid XAUTH.

  You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for ASA authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID.
- ISAKMP (IKE) keepalive settings. This feature lets the ASA monitor the continued presence of a remote peer and report its own presence to that peer. If the peer becomes unresponsive, the ASA removes the connection. Enabling IKE keepalives prevents hung connections when the IKE peer loses connectivity.

  There are various forms of IKE keepalives. For this feature to work, both the ASA and its remote peer must support a common form. This feature works with the following peers:
  - Cisco AnyConnect VPN Client
  - Cisco IOS software
  - Cisco Secure PIX Firewall

    Non-Cisco VPN clients do not support IKE keepalives.

    If you are configuring a group of mixed peers, and some of those peers support IKE keepalives and others do not, enable IKE keepalives for the entire group. The feature does not affect the peers that do not support it.

If you disable IKE keepalives, connections with unresponsive peers remain active until they time out, so we recommend that you keep your idle timeout short. To change your idle timeout, see Configure Group Policies, on page 133.

**Note** To reduce connectivity costs, disable IKE keepalives if this group includes any clients connecting via ISDN lines. ISDN connections normally disconnect if idle, but the IKE keepalive mechanism prevents connections from idling and therefore from disconnecting.

If you do disable IKE keepalives, the client disconnects only when either its IKE or IPsec keys expire. Failed traffic does not disconnect the tunnel with the Peer Timeout Profile values as it does when IKE keepalives are enabled.

If you have a LAN-to-LAN configuration using IKE main mode, make sure that the two peers have the same IKE keepalive configuration. Both peers must have IKE keepalives enabled or both peers must have it disabled.

- If you configure authentication using digital certificates, you can specify whether to send the entire certificate chain (which sends the peer the identity certificate and all issuing certificates) or just the issuing certificates (including the root certificate and any subordinate CA certificates).
- You can notify users who are using outdated versions of Windows client software that they need to update their client, and you can provide a mechanism for them to get the updated client version. You can configure and change the client-update, either for all connection profiles or for particular connection profiles.
- If you configure authentication using digital certificates, you can specify the name of the trustpoint that identifies the certificate to send to the IKE peer.

# Connection Profile Connection Parameters for SSL VPN Sessions

The table below provides a list of connection profile attributes that are specific to SSL VPN (Secure Client and clientless) connections. In addition to these attributes, you configure general connection profile attributes common to all VPN connections.

**Note** In earlier releases, “connection profiles” were known as “tunnel groups.” You configure a connection profile with tunnel-group commands. This chapter often uses these terms interchangeably.

***Table 7: Connection Profile Attributes for SSL VPN***

| | Function |
|---|---|
| **authentication** | Sets the authentication method, AAA or certificate. |
| **customization** | Identifies the name of a previously defined customization to apply. Customizations determine the appearance of the windows that the user sees upon login. You configure the customization parameters as part of configuring clientless SSL VPN. |

| | Function |
|---|---|
| **nbns-server** | Identifies the name of the NetBIOS Name Service server (nbns-server) to use for CIFS name resolution. |
| **group-alias** | Specifies one or more alternate names by which the server can refer to a connection profile. At login, the user selects the group name from a drop-down menu. |
| **group-url** | Identifies one or more group URLs. If you configure this attribute, users coming in on a specified URL need not select a group at login.<br>A Load Balancing deployment that uses Group URLs for Secure Client connectivity, requires each ASA node in the cluster to configure a Group URL for the virtual cluster address, as well as a Group URL for the node's Load Balancing public address. |
| **dns-group** | Identifies the DNS server group that specifies the DNS server name, domain name, name server, number of retries, and timeout values for a DNS server to use for a connection profile. |
| **hic-fail-group-policy** | Specifies a VPN feature policy if you use the Cisco Secure Desktop Manager to set the Group-Based Policy attribute to "Use Failure Group-Policy" or "Use Success Group-Policy, if criteria match." |
| **override-svc-download** | Overrides downloading the group-policy or username attributes configured for downloading the AnyConnect VPN client to the remote user. |
| **radius-reject-message** | Enables the display of the RADIUS reject message on the login screen when authentication is rejected. |

# Configure Connection Profiles

This section describes the contents and configuration of connection profiles in both single-context mode or multiple-context mode.



**Note** Multiple-context mode applies only to IKEv2 and IKEv1 site to site and does not apply to Secure Client, Clientless SSL VPN, legacy Cisco VPN client, the Apple native VPN client, the Microsoft native VPN client, or cTCP for IKEv1 IPsec.

You can modify the default connection profiles, and you can configure a new connection profile as any of the three tunnel-group types. If you do not explicitly configure an attribute in a connection profile, that attribute gets its value from the default connection profile. The default connection-profile type is remote access. The subsequent parameters depend upon your choice of tunnel type. To see the current configured and default configuration of all your connection profiles, including the default connection profile, enter the **show running-config all tunnel-group** command.

# Maximum Connection Profiles

The maximum number of connection profiles (tunnel groups) that an ASA can support is a function of the maximum number of concurrent VPN sessions for the platform + 5. Attempting to add an additional tunnel group beyond the limit results in the following message: “ERROR: The limit of 30 configured tunnel groups has been reached.”

# Default IPsec Remote Access Connection Profile Configuration

The contents of the default remote-access connection profile are as follows:

```
tunnel-group DefaultRAGroup type remote-access
tunnel-group DefaultRAGroup general-attributes
 no address-pool
 no ipv6-address-pool
 authentication-server-group LOCAL
 accounting-server-group RADIUS
 default-group-policy DfltGrpPolicy
 no dhcp-server
 no strip-realm
 no password-management
 no override-account-disable
 no strip-group
 no authorization-required
 authorization-dn-attributes CN OU
tunnel-group DefaultRAGroup webvpn-attributes
 hic-fail-group-policy DfltGrpPolicy
 customization DfltCustomization
 authentication aaa
 no override-svc-download
 no radius-reject-message
 dns-group DefaultDNS
tunnel-group DefaultRAGroup ipsec-attributes
 no pre-shared-key
 peer-id-validate req
 no chain
 no trust-point
 isakmp keepalive threshold 1500 retry 2
 no radius-sdi-xauth
 isakmp ikev1-user-authentication xauth
tunnel-group DefaultRAGroup ppp-attributes
 no authentication pap
 authentication chap
 authentication ms-chap-v1
 no authentication ms-chap-v2
 no authentication eap-proxy


tunnel-group DefaultRAGroup type remote-access
tunnel-group DefaultRAGroup general-attributes
 no address-pool
 no ipv6-address-pool
 authentication-server-group LOCAL
 accounting-server-group RADIUS
 default-group-policy DfltGrpPolicy
 no dhcp-server
 no strip-realm
 no password-management
 no strip-group
 no authorization-required
 authorization-dn-attributes CN OU
```

```
tunnel-group DefaultRAGroup webvpn-attributes
 hic-fail-group-policy DfltGrpPolicy
 customization DfltCustomization
 authentication aaa
 no override-svc-download
 no radius-reject-message
 dns-group DefaultDNS
tunnel-group DefaultRAGroup ipsec-attributes
 no pre-shared-key
 peer-id-validate req
 no chain
 no trust-point
 isakmp keepalive threshold 1500 retry 2
 no radius-sdi-xauth
 isakmp ikev1-user-authentication xauth
tunnel-group DefaultRAGroup ppp-attributes
 no authentication pap
 authentication chap
 authentication ms-chap-v1
 no authentication ms-chap-v2
 no authentication eap-proxy
```

# IPsec Tunnel-Group General Attributes

The general attributes are common across more than one tunnel-group type. IPsec remote access and clientless SSL VPN tunnels share most of the same general attributes. IPsec LAN-to-LAN tunnels use a subset. Refer to the *Cisco Secure FIrewall ASA Series Command Reference* for complete descriptions of all commands. This section describes, in order, how to configure remote-access and LAN-to-LAN connection profiles.

# Configure Remote-Access Connection Profiles

Use a remote-access connection profile when setting up a connection between the following remote clients and a central-site ASA:

- Secure Client (connecting with SSL or IPsec/IKEv2)
- Clientless SSL VPN (browser-based connecting with SSL)
- Cisco ASA 5500 Easy VPN hardware client (connecting with IPsec/IKEv1)

We also provide a default group policy named DfltGrpPolicy.

To configure a remote-access connection profile, first configure the tunnel-group general attributes, then the remote-access attributes. See the following sections:

- Specify a Name and Type for the Remote Access Connection Profile, on page 107.
- Configure Remote-Access Connection Profile General Attributes, on page 107.
- Configure Double Authentication, on page 111
- Configure Remote-Access Connection Profile IPsec IKEv1 Attributes, on page 113.
- Configure IPsec Remote-Access Connection Profile PPP Attributes, on page 115

## Specify a Name and Type for the Remote Access Connection Profile

### Procedure

Create the connection profile, specifying its name and type, by entering the **tunnel-group** command.

For a remote-access tunnel, the type is **remote-access**.

**tunnel-group** *tunnel_group_name* **type remote-access**

**Example:**

For example, to create a remote-access connection profile named TunnelGroup1, enter the following command:

```
hostname(config)# tunnel-group TunnelGroup1 type remote-access
hostname(config)#
```

## Configure Remote-Access Connection Profile General Attributes

To configure or change the connection profile general attributes, specify the parameters in the following steps:

### Procedure

**Step 1** To configure the general attributes, enter the **tunnel-group general-attributes** task in either single or multiple context mode, which enters tunnel-group general-attributes configuration mode. The prompt changes to indicate the change in mode.

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

**Step 2** Specify the name of the authentication-server group, if any, to use. If you want to use the LOCAL database for authentication if the specified server group fails, append the keyword **LOCAL**:

```
hostname(config-tunnel-general)# authentication-server-group [(interface_name)] groupname
[LOCAL]
hostname(config-tunnel-general)#
```

The name of the authentication server group can be up to 16 characters long.

You can optionally configure interface-specific authentication by including the name of an interface after the group name. The interface name, which specifies where the tunnel terminates, must be enclosed in parentheses. The following command configures interface-specific authentication for the interface named test using the server named servergroup1 for authentication:

```
hostname(config-tunnel-general)# authentication-server-group (test) servergroup1
hostname(config-tunnel-general)#
```

**Step 3** Specify the name of the authorization-server group, if any, to use. When you configure this value, users must exist in the authorization database to connect:

```
hostname(config-tunnel-general)# authorization-server-group groupname
hostname(config-tunnel-general)#
```

The name of the authorization server group can be up to 16 characters long. For example, the following command specifies the use of the authorization-server group FinGroup:

```
hostname(config-tunnel-general)# authorization-server-groupFinGroup
hostname(config-tunnel-general)#
```

**Step 4** Specify the name of the accounting-server group, if any, to use:

```
hostname(config-tunnel-general)# accounting-server-group groupname
hostname(config-tunnel-general)#
```

The name of the accounting server group can be up to 16 characters long. For example, the following command specifies the use of the accounting-server group named comptroller:

```
hostname(config-tunnel-general)# accounting-server-group comptroller
hostname(config-tunnel-general)#
```

**Step 5** Specify the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy policyname
hostname(config-tunnel-general)#
```

The name of the group policy can be up to 64 characters long. The following example sets DfltGrpPolicy as the name of the default group policy:

```
hostname(config-tunnel-general)# default-group-policy DfltGrpPolicy
hostname(config-tunnel-general)#
```

**Step 6** Specify the names or IP addresses of the DHCP server (up to 10 servers), and the names of the DHCP address pools (up to 6 pools). The defaults are no DHCP server and no address pool. The dhcp-server command will allow you to configure the ASA to send additional options to the specified DHCP servers when it is trying to get IP addresses for VPN clients. See the dhcp-server command in the Cisco Secure Firewall ASA Series Command Reference guide for more information.

```
hostname(config-tunnel-general)# dhcp-server server1 [...server10]
hostname(config-tunnel-general)# address-pool [(interface name)] address_pool1
[...address_pool6]
hostname(config-tunnel-general)#
```

**Note**
If you specify an interface name, you must enclosed it within parentheses.

You configure address pools with the **ip local pool** command in global configuration mode.

**Step 7** Specify the name of the NAC authentication server group, if you are using Network Admission Control, to identify the group of authentication servers to be used for Network Admission Control posture validation. Configure at least one Access Control Server to support NAC. Use the **aaa-server** command to name the ACS group. Then use the **nac-authentication-server-group** command, using the same name for the server group.

The following example identifies acs-group1 as the authentication server group to be used for NAC posture validation:

```
hostname(config-group-policy)# nac-authentication-server-group acs-group1
hostname(config-group-policy)
```

The following example inherits the authentication server group from the default remote access group:

```
hostname(config-group-policy)# no nac-authentication-server-group
hostname(config-group-policy)
```

**Note**
NAC requires a Cisco Trust Agent on the remote host.

**Step 8** Specify whether to strip the group or the realm from the username before passing it on to the AAA server. The default is not to strip either the group name or the realm:

```
hostname(config-tunnel-general)# strip-group
hostname(config-tunnel-general)# strip-realm
hostname(config-tunnel-general)#
```

A realm is an administrative domain. If you strip the realm, the ASA uses the username and the group (if present) authentication. If you strip the group, the ASA uses the username and the realm (if present) for authentication. Enter the **strip-realm** command to remove the realm qualifier, and use the strip-group command to remove the group qualilfier from the username during authentication. If you remove both qualifiers, authentication is based on the *username* alone. Otherwise, authentication is based on the full *username@realm* or *username*<delimiter> *group* string. You must specify **strip-realm** if your server is unable to parse delimiters.

**Step 9** Optionally, if your server is a RADIUS, RADIUS with NT, or LDAP server, you can enable password management.

**Note**
If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

Sun—The DN configured on the ASA to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.

Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

This feature, which is disabled by default, warns a user when the current password is about to expire. The default is to begin warning the user 14 days before expiration:

```
hostname(config-tunnel-general)# password-management
```

```
hostname(config-tunnel-general)#
```

If the server is an LDAP server, you can specify the number of days (0 through 180) before expiration to begin warning the user about the pending expiration:

```
hostname(config-tunnel-general)# password-management [password-expire in days n]
hostname(config-tunnel-general)#
```

**Note**
The **password-management** command, entered in tunnel-group general-attributes configuration mode replaces the deprecated **radius-with-expiry** command that was formerly entered in tunnel-group ipsec-attributes mode.

When you configure the **password-management** command, the ASA notifies the remote user at login that the user's current password is about to expire or has expired. The ASA then offers the user the opportunity to change the password. If the current password has not yet expired, the user can still log in using that password. The ASA ignores this command if RADIUS or LDAP authentication has not been configured.

Note that this does not change the number of days before the password expires, but rather, the number of days ahead of expiration that the ASA starts warning the user that the password is about to expire.

If you do specify the **password-expire-in-days** keyword, you must also specify the number of days.

Specifying this command with the number of days set to 0 disables this command. The ASA does not notify the user of the pending expiration, but the user can change the password after it expires.

See Configure Microsoft Active Directory Settings for Password Management, on page 126 for more information.

The ASA Version 7.1 and later generally supports password management for the AnyConnect VPN Client, the Cisco IPsec VPN Client, the SSL VPN full-tunneling client, and Clientless connections when authenticating with LDAP or with any RADIUS connection that supports MS-CHAPv2. Password management is *not* supported for any of these connection types for AD (Windows password) or NT 4.0 Domain.

Some RADIUS servers that support MS-CHAP do not currently support MS-CHAPv2. The **password-management** command requires MS-CHAPv2, so please check with your vendor.

**Note**
The RADIUS server (for example, Cisco ACS) could proxy the authentication request to another authentication server. However, from the ASA perspective, it is talking only to a RADIUS server.

For LDAP, the method to change a password is proprietary for the different LDAP servers on the market. Currently, the ASA implements the proprietary password management logic only for Microsoft Active Directory and Sun LDAP servers. Native LDAP requires an SSL connection. You must enable LDAP over SSL before attempting to do password management for LDAP. By default, LDAP uses port 636.

**Step 10**

**Step 11** Specify the attribute or attributes to use in deriving a name for an authorization query from a certificate. This attribute specifies what part of the subject DN field to use as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes {primary-attribute
[secondary-attribute] | use-entire-name}
```

For example, the following command specifies the use of the CN attribute as the username for authorization:

```
hostname(config-tunnel-general)# authorization-dn-attributes CN
hostname(config-tunnel-general)#
```

The authorization-dn-attributes are **C** (Country), **CN** (Common Name), **DNQ** (DN qualifier), **EA** (E-mail Address), **GENQ** (Generational qualifier), **GN** (Given Name), **I** (Initials), **L** (Locality), **N** (Name), **O** (Organization), **OU** (Organizational Unit), **SER** (Serial Number), **SN** (Surname), **SP** (State/Province), **T** (Title), **UID** (User ID), and **UPN** (User Principal Name).

**Step 12** Specify whether to require a successful authorization before allowing a user to connect. The default is not to require authorization.

```
hostname(config-tunnel-general)# authorization-required
hostname(config-tunnel-general)#
```

## Configure Double Authentication

Double authentication is an optional feature that requires a user to enter an additional authentication credential, such as a second username and password, on the login screen. Specify the following commands to configure double authentication.

### Procedure

**Step 1** Specify the secondary authentication server group. This command specifies the AAA server group to use as the secondary AAA server.

**Note**
This command applies only to AnyConnect VPN connections.

The secondary server group cannot specify an SDI server group. By default, no secondary authentication is required.

```
hostname(config-tunnel-general)# secondary-authentication-server-group [interface_name]
{none | LOCAL | groupname [LOCAL]} [use-primary-name]
```

If you use the none keyword, no secondary authentication is required. The *groupname* value specifies the AAA server group name. Local specifies the use of the internal server database, and when used with the groupname value, LOCAL specifies fallback.

For example, to set the primary authentication server group to sdi_group and the secondary authentication server group to ldap_server, enter the following commands:

```
hostname(config-tunnel-general)# authentication-server-group
hostname(config-tunnel-general)# secondary-authentication-server-group
```

**Note**
If you use the **use-primary-name** keyword, then the login dialog requests only one username. In addition, if the usernames are extracted from a digital certificate, only the primary username is used for authentication.

**Step 2** If obtaining the secondary username from a certificate, enter **secondary-username-from-certificate**:

```
hostname(config-tunnel-general)# secondary-username-from-certificate C | CN | ... | use-script
```

The values for the DN fields to extract from the certificate for use as a secondary username are the same as for the primary **username-from-certificate** command. Alternatively, you can specify the use-script keyword, which directs the ASA to use a script file generated by ASDM.

For example, to specify the Common Name as the primary username field and Organizational Unit as the secondary username field, enter the following commands:

```
hostname(config-tunnel-general)# tunnel-group test1 general-attributes
hostname(config-tunnel-general)# username-from-certificate cn
hostname(config-tunnel-general)# secondary-username-from-certificate ou
```

**Step 3** Use the **secondary-pre-fill-username** command in tunnel-group webvpn-attributes mode to enable extracting a secondary username from a client certificate for use in authentication. Use the keywords to specify whether this command applies to a clientless connection or an SSL VPN client (AnyConnect) connection and whether you want to hide the extracted username from the end user. This feature is disabled by default. Clientless and SSL-client options can both exist at the same time, but you must configure them in separate commands.

```
hostname(config-tunnel-general)# secondary-pre-fill-username-from-certificate
{clientless | client} [hide]
```

For example, to specify the use of pre-fill-username for both the primary and secondary authentication for a connection, enter the following commands:

```
hostname(config-tunnel-general)# tunnel-group test1 general-attributes
hostname(config-tunnel-general)# pre-fill-username client
hostname(config-tunnel-general)# secondary-pre-fill-username client
```

**Step 4** Specify which authentication server to use to obtain the authorization attributes to apply to the connection. The primary authentication server is the default selection. This command is meaningful only for double authentication.

```
hostname(config-tunnel-general)# authentication-attr-from-server {primary | secondary}
```

For example, to specify the use of the secondary authentication server, enter the following commands:

```
hostname(config-tunnel-general)# tunnel-group test1 general-attributes
hostname(config-tunnel-general)# authentication-attr-from-server secondary
```

**Step 5** Specify which authentication username, primary or secondary, to associate with the session. The default value is primary. With double authentication enabled, it is possible that two distinct usernames are authenticated for the session. The administrator must designate one of the authenticated usernames as the session username. The session username is the username provided for accounting, session database, syslogs, and debug output.

```
hostname(config-tunnel-general)# authenticated-session-username {primary | secondary}
```

For example, to specify that the authentication username associated with the session must come from the secondary authentication server, enter the following commands:

```
hostname(config-tunnel-general)# tunnel-group test1 general-attributes
hostname(config-tunnel-general)# authenticated-session-username secondary
```

## Configure Remote-Access Connection Profile IPsec IKEv1 Attributes

To configure the IPsec IKEv1 attributes for a remote-access connection profile, perform the following steps. The following description assumes that you have already created the remote-access connection profile. Remote-access connection profiles have more attributes than LAN-to-LAN connection profiles.

### Procedure

**Step 1** To specify the IPsec attributes of an remote-access tunnel-group, enter tunnel-group ipsec-attributes mode by entering the following command in either single or multiple context mode. The prompt changes to indicate the mode change.

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

This command enters tunnel-group ipsec-attributes configuration mode, in which you configure the remote-access tunnel-group IPsec attributes in either single or multiple context mode.

For example, the following command designates that the tunnel-group ipsec-attributes mode commands that follow pertain to the connection profile named TG1. Notice that the prompt changes to indicate that you are now in tunnel-group ipsec-attributes mode:

```
hostname(config)# tunnel-group TG1 type remote-access
hostname(config)# tunnel-group TG1 ipsec-attributes
hostname(config-tunnel-ipsec)#
```

**Step 2** Specify the preshared key to support IKEv1 connections based on preshared keys. For example, the following command specifies the preshared key xyzx to support IKEv1 connections for an IPsec IKEv1 remote access connection profile:

```
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key xyzx
hostname(config-tunnel-ipsec)#
```

**Step 3** Specify whether to validate the identity of the peer using the peer's certificate:

```
hostname(config-tunnel-ipsec)# peer-id-validate option
hostname(config-tunnel-ipsec)#
```

The possible *option* values are **req** (required), **cert (**if supported by certificate), and **nocheck (**do not check). The default is **req**.

For example, the following command specifies that peer-id validation is required:

```
hostname(config-tunnel-ipsec)# peer-id-validate req
hostname(config-tunnel-ipsec)#
```

**Step 4** Specify whether to enable sending of a certificate chain. The following command includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-tunnel-ipsec)# chain
hostname(config-tunnel-ipsec)#
```

This attribute applies to all IPsec tunnel-group types.

**Step 5** Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-tunnel-ipsec)# ikev1 trust-point trust-point-name
hostname(config-tunnel-ipsec)#
```

The following command specifies mytrustpoint as the name of the certificate to be sent to the IKE peer:

```
hostname(config-ipsec)# ikev1 trust-point mytrustpoint
```

**Step 6** Specify the ISAKMP keepalive threshold and the number of retries allowed:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold <number> retry <number>
hostname(config-tunnel-ipsec)#
```

The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable ISAKMP keepalives, enter **isakmp keepalive disable**.

For example, the following command sets the IKE keepalive threshold value to 15 seconds and sets the retry interval to 10 seconds:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold 15 retry 10
hostname(config-tunnel-ipsec)#
```

The default value for the **threshold** parameter is 300 for remote-access and 10 for LAN-to-LAN, and the default value for the retry parameter is 2.

To specify that the central site (secure gateway) should never initiate ISAKMP monitoring, enter the following command:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold infinite
hostname(config-tunnel-ipsec)#
```

**Step 7** Specify the ISAKMP hybrid authentication method, XAUTH or hybrid XAUTH.

You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for ASA authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID. Hybrid XAUTH breaks phase 1 of IKE down into the following two steps, together called hybrid authentication:

a) The ASA authenticates to the remote VPN user with standard public key techniques. This establishes an IKE security association that is unidirectionally authenticated.
b) An XAUTH exchange then authenticates the remote VPN user. This extended authentication can use one of the supported legacy authentication methods.

**Note**
Before the authentication type can be set to hybrid, you must configure the authentication server, create a preshared key, and configure a trustpoint.

You can use the **isakmp ikev1-user-authentication** command with the optional interface parameter to specify a particular interface. When you omit the interface parameter, the command applies to all the interfaces and serves as a back-up when the per-interface command is not specified. When there are two **isakmp ikev1-user-authentication** commands specified for a connection profile, and one uses the **interface** parameter and one does not, the one specifying the interface takes precedence for that particular interface.

For example, the following commands enable hybrid XAUTH on the inside interface for a connection profile called example-group:

```
hostname(config)# tunnel-group example-group type remote-access
hostname(config)# tunnel-group example-group ipsec-attributes
hostname(config-tunnel-ipsec)# isakmp ikev1-user-authentication (inside) hybrid
hostname(config-tunnel-ipsec)#
```

## Configure IPsec Remote-Access Connection Profile PPP Attributes

To configure the Point-to-Point Protocol attributes for a remote-access connection profile, perform the following steps. PPP attributes apply *only* to IPsec remote-access connection profiles. The following description assumes that you have already created the IPsec remote-access connection profile.

### Procedure

**Step 1** Enter tunnel-group ppp-attributes configuration mode, in which you configure the remote-access tunnel-group PPP attributes, by entering the following command. The prompt changes to indicate the mode change:

```
hostname(config)# tunnel-group tunnel-group-name type remote-access
hostname(config)# tunnel-group tunnel-group-name ppp-attributes
hostname(config-tunnel-ppp)#
```

For example, the following command designates that the tunnel-group ppp-attributes mode commands that follow pertain to the connection profile named TG1. Notice that the prompt changes to indicate that you are now in tunnel-group ppp-attributes mode:

```
hostname(config)# tunnel-group TG1 type remote-access
hostname(config)# tunnel-group TG1 ppp-attributes
hostname(config-tunnel-ppp)#
```

**Step 2** Specify whether to enable authentication using specific protocols for the PPP connection. The protocol value can be any of the following:

- pap—Enables the use of Password Authentication Protocol for the PPP connection.
- chap—Enables the use of Challenge Handshake Authentication Protocol for the PPP connection.
- ms-chap-v1 or ms-chap-v2—Enables the use of Microsoft Challenge Handshake Authentication Protocol, version 1 or version 2 for the PPP connection.
- eap—Enables the use of Extensible Authentication protocol for the PPP connection.

CHAP and MSCHAPv1 are enabled by default.

The syntax of this command is:

```
hostname(config-tunnel-ppp)# authentication protocol
hostname(config-tunnel-ppp)#
```

To disable authentication for a specific protocol, use the **no** form of the command:

```
hostname(config-tunnel-ppp)# no authentication protocol
hostname(config-tunnel-ppp)#
```

For example, the following command enables the use of the PAP protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# authentication pap
hostname(config-tunnel-ppp)#
```

The following command enables the use of the MS-CHAP, version 2 protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# authentication ms-chap-v2
hostname(config-tunnel-ppp)#
```

The following command enables the use of the EAP-PROXY protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# authentication pap
hostname(config-tunnel-ppp)#
```

The following command disables the use of the MS-CHAP, version 1 protocol for a PPP connection:

```
hostname(config-tunnel-ppp)# no authentication ms-chap-v1
hostname(config-tunnel-ppp)#
```

# Configure LAN-to-LAN Connection Profiles

An IPsec LAN-to-LAN VPN connection profile applies only to LAN-to-LAN IPsec client connections. While many of the parameters that you configure are the same as for IPsec remote-access connection profiles, LAN-to-LAN tunnels have fewer parameters. The following sections show you how to configure a LAN-to-LAN connection profile:


- Specify a Name and Type for a LAN-to-LAN Connection Profile, on page 117
- Configure LAN-to-LAN Connection Profile General Attributes, on page 117
- Configure LAN-to-LAN IPsec IKEv1 Attributes, on page 118


## Default LAN-to-LAN Connection Profile Configuration

The contents of the default LAN-to-LAN connection profile are as follows:

```
tunnel-group DefaultL2LGroup type ipsec-l2l
tunnel-group DefaultL2LGroup general-attributes
 default-group-policy DfltGrpPolicy
tunnel-group DefaultL2LGroup ipsec-attributes
 no ikev1 pre-shared-key
 peer-id-validate req
 no chain
 no ikev1 trust-point
 isakmp keepalive threshold 10 retry 2
```

LAN-to-LAN connection profiles have fewer parameters than remote-access connection profiles, and most of these are the same for both groups. For your convenience in configuring the connection, they are listed separately here. Any parameters that you do not explicitly configure inherit their values from the default connection profile.

## Specify a Name and Type for a LAN-to-LAN Connection Profile

To specify a name and a type for a connection profile, enter the **tunnel-group** command, as follows:

```
hostname(config)# tunnel-group tunnel_group_name type tunnel_type
```

For a LAN-to-LAN tunnel, the type is **ipsec-l2l**.; for example, to create the LAN-to-LAN connection profile named docs, enter the following command:

```
hostname(config)# tunnel-group docs type ipsec-l2l
hostname(config)#
```

## Configure LAN-to-LAN Connection Profile General Attributes

To configure the connection profile general attributes, perform the following steps:

**Procedure**

**Step 1** Enter tunnel-group general-attributes mode by specifying the general-attributes keyword in either single or multiple context mode:

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

**Example:**

For the connection profile named docs, enter the following command:

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

The prompt changes to indicate that you are now in config-general mode, in which you configure the tunnel-group general attributes.

**Step 2** Specify the name of the default group policy:

**default-group-policy** *policyname*

**Example:**

The following command specifies that the name of the default group policy is MyPolicy:

```
hostname(config-tunnel-general)# default-group-policy MyPolicy
hostname(config-tunnel-general)#
```

## Configure LAN-to-LAN IPsec IKEv1 Attributes

To configure the IPsec IKEv1 attributes, perform the following steps:

**Procedure**

**Step 1** To configure the tunnel-group IPsec IKEv1 attributes, enter tunnel-group ipsec-attributes configuration mode by entering the tunnel-group command with the IPsec-attributes keyword in either single or multiple context mode.

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

For example, the following command enters config-ipsec mode so that you can configure the parameters for the connection profile named TG1:

```
hostname(config)# tunnel-group TG1 ipsec-attributes
hostname(config-tunnel-ipsec)#
```

The prompt changes to indicate that you are now in tunnel-group ipsec-attributes configuration mode.

**Step 2** Specify the preshared key to support IKEv1 connections based on preshared keys.

```
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key key
hostname(config-tunnel-ipsec)#
```

For example, the following command specifies the preshared key XYZX to support IKEv1 connections for an LAN-to-LAN connection profile:

```
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key xyzx
hostname(config-tunnel-general)#
```

**Step 3** Specify whether to validate the identity of the peer using the peer's certificate:

```
hostname(config-tunnel-ipsec)# peer-id-validate option
hostname(config-tunnel-ipsec)#
```

The available options are **req** (required), **cert (**if supported by certificate), and **nocheck (**do not check). The default is **req**. For example, the following command sets the peer-id-validate option to **nocheck**:

```
hostname(config-tunnel-ipsec)# peer-id-validate nocheck
hostname(config-tunnel-ipsec)#
```

**Step 4** Specify whether to enable sending of a certificate chain. This action includes the root certificate and any subordinate CA certificates in the transmission:

```
hostname(config-tunnel-ipsec)# chain
hostname(config-tunnel-ipsec)#
```

You can apply this attribute to all tunnel-group types.

**Step 5** Specify the name of a trustpoint that identifies the certificate to be sent to the IKE peer:

```
hostname(config-tunnel-ipsec)# trust-point trust-point-name
hostname(config-tunnel-ipsec)#
```

For example, the following command sets the trustpoint name to mytrustpoint:

```
hostname(config-tunnel-ipsec)# trust-point mytrustpoint
hostname(config-tunnel-ipsec)#
```

You can apply this attribute to all tunnel-group types.

**Step 6** Specify the ISAKMP (IKE) keepalive threshold and the number of retries allowed. The **threshold** parameter specifies the number of seconds (10 through 3600) that the peer is allowed to idle before beginning keepalive monitoring. The **retry** parameter is the interval (2 through 10 seconds) between retries after a keepalive response has not been received. IKE keepalives are enabled by default. To disable IKE keepalives, enter the **no** form of the **isakmp** command:

```
hostname(config)# isakmp keepalive threshold <number> retry <number>
hostname(config-tunnel-ipsec)#
```

For example, the following command sets the ISAKMP keepalive threshold to 15 seconds and sets the retry interval to 10 seconds:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold 15 retry 10
hostname(config-tunnel-ipsec)#
```

The default value for the **threshold** parameter for LAN-to-LAN is 10, and the default value for the retry parameter is 2.

To specify that the central site (secure gateway) should never initiate ISAKMP monitoring, enter the following command:

```
hostname(config-tunnel-ipsec)# isakmp keepalive threshold infinite
hostname(config-tunnel-ipsec)#
```

**Step 7** Specify the ISAKMP hybrid authentication method, XAUTH or hybrid XAUTH.

You use **isakmp ikev1-user-authentication** command to implement hybrid XAUTH authentication when you need to use digital certificates for ASA authentication and a different, legacy method for remote VPN user authentication, such as RADIUS, TACACS+ or SecurID. Hybrid XAUTH breaks phase 1 of IKE down into the following two steps, together called hybrid authentication:

a) The ASA authenticates to the remote VPN user with standard public key techniques. This establishes an IKE security association that is unidirectionally authenticated.
b) An XAUTH exchange then authenticates the remote VPN user. This extended authentication can use one of the supported legacy authentication methods.

**Note**
Before the authentication type can be set to hybrid, you must configure the authentication server, create a preshared key, and configure a trustpoint.

For example, the following commands enable hybrid XAUTH for a connection profile called example-group:

```
hostname(config)# tunnel-group example-group type remote-access
hostname(config)# tunnel-group example-group ipsec-attributes
hostname(config-tunnel-ipsec)# isakmp ikev1-user-authentication hybrid
hostname(config-tunnel-ipsec)#
```

# About Tunnel Groups for Standards-based IKEv2 Clients

A tunnel group is a set of records that contain 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.

The default tunnel group for IPsec remote access is the DefaultRAGroup. You may modify the default tunnel group, but not delete it.

IKEv2 allows asymmetric authentication methods to be configured (that is, preshared key authentication for the originator but certificate authentication or EAP authentication for the responder) using separate local and remote authentication CLIs. Therefore, with IKEv2 you have asymmetric authentication, in which one side authenticates with one credential and the other side uses another credential (either a preshared key, certificate, or EAP).

The DefaultRAGroup should be configured for EAP authentication because these client connections cannot be mapped to a specific tunnel group unless certificate authentication is used with certificate DN matching.

## Standards-based IKEv2 Attribute Support

The ASA supports the following IKEv2 attributes:

- INTERNAL_IP4_ADDRESS/INTERNAL_IP6_ADDRESS—IPv4 or IPv6 address

**Note** Dual stack (assignment of both an IPv4 and IPv6 address) is not supported for IKEv2. If both an IPv4 and an IPv6 address are requested and both addresses may be assigned, only an IPv4 address is assigned.

- INTERNAL_IP4_NETMASK—IPv4 address network mask
- INTERNAL_IP4_DNS/INTERNAL_IP6_DNS—Primary/Secondary DNS address
- INTERNAL_IP4_NBNS—Primary/Secondary WINS address
- INTERNAL_IP4_SUBNET/INTERNAL_IP6_SUBNET—Split-tunneling lists
- APPLICATION_VERSION—Ignored. No response is sent to avoid communicating any version information about the ASA for security reasons. However, the client configuration payload request may include this attribute, and the string appears on the ASA in the **vpn-sessiondb** command output and in the syslog.

## DAP Support

To allow DAP policy configuration per connection type, a new Client Type, IPsec-IKEv2-Generic-RA, can be used to apply specific policy for this connection type.

## Tunnel Group Selection for Remote Access Clients

The following table provides a list of remote access clients and their available tunnel group options:

| Remote Access Client | Tunnel Group List | Group URL | Certificate DN Matching | Default Group (DefaultRAGroup) | Other |
|---|---|---|---|---|---|
| AnyConnect VPN Client | Yes | Yes | Yes | Yes | N/A |

| | | | | | |
|---|---|---|---|---|---|
| Windows L2TP/IPsec<br>(Main Mode IKEv1) | No | No | • Yes (when using local machine certificates)<br>• No (when using PSK) | Yes | N/A |
| Standards-based IKEv2 | No | No | • Yes (when using local machine certificates)<br>• No (when using EAP authentication) | Yes<br>**Note** You must use the DefaultRAGroup tunnel group. | N/A |

## Authentication Support for Standards-based IKEv2 Clients

The following table provides a list of standards-based IKEv2 clients and their supported authentication methods:



**Note** Authentication method limitations are based on lack of support on the client, not on the ASA. All EAP method authentication is proxied by the ASA between the client and EAP server. EAP method support is based on client and EAP server support for the EAP method.

| Client Type/ Authentication Method | EAP-TLS | EAP-MSCHAPv2 | EAP-MD5 | Certificate Only | PSK |
|---|---|---|---|---|---|
| StrongSwan on Linux | N/A | • ISE—Yes<br>• ACS—Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | • ISE—Yes<br>• ACS—Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | Yes | Yes |
| StrongSwan on Android | N/A | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | No | Yes | N/A |
| Windows 7/8/8.1 | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | N/A | Yes | NA |

| Client Type/ Authentication Method | EAP-TLS | EAP-MSCHAPv2 | EAP-MD5 | Certificate Only | PSK |
|---|---|---|---|---|---|
| Windows Phone | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | N/A | N/A | N/A |
| Samsung Knox | N/A | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | Yes | N/A |
| iOS 8 | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | N/A | Yes | Yes |
| Android Native Client | N/A | • ISE—Yes<br>• ACS —Yes<br>• FreeRadius—Yes<br>• AD via FreeRadius—Yes | N/A | Yes | Yes |

## Add Multiple Certificate Authentication

The Aggregate Authentication protocol has been extended to define the protocol exchange for multiple-certificate authentication and utilize this for both session types. After the client makes an SSL connection and enters into aggregate authentication, another SSL connection is made, and the ASA sees that the client requires certificate authentication and requests the client certificate.

The ASA configures the required authentication for an Secure Client connection of a remote-access type tunnel group. A tunnel-group mapping is performed with the existing methods such as certificate rule mapping, group-url, and so on, but then the required authentication methods are negotiated with the client.

### Example

tunnel-group <name> webvpn-attributes

authentication {aaa [certificate | multiple-certificate] | multiple-certificate [aaa | saml] | saml [certificate | multiple-certificate]}

The authentication options are AAA only, certificate only, multiple-certificate only, AAA and certificate, AAA and multiple-certificate, SAML, SAML and certificate, or Multiple certificates and SAML.

```
ASA(config)# tunnel-group AnyConnect webvpn-attributes
ASA(config-tunnel-webvpn)# authentication?
tunnel-group-webvpn mode commands/options:
aaa         Use username and password for authentication
certificate Use certificate for authentication
multiple-certificate Use multiple certificates for authentication
saml   Use SAML for authentication
ASA(config-tunnel-webvpn)# authentication multiple-certificate?

tunnel-group-webvpn mode commands/options:
aaa  Use username and password for authentication
saml  Use SAML for authentication
<cr>

ASA(config-tunnel-webvpn)# authentication aaa?

tunnel-group-webvpn mode commands/options:
certificate Use certificate for authentication
multiple-certificate Use multiple certificates for authentication
<cr>ASA(config-tunnel-webvpn)# authentication aaa?

ASA(config-tunnel-webvpn)# authentication saml?
tunnel-group-webvpn mode commands/options:
certificate Use certificate for authentication
multiple-certificate Use multiple certificates for authentication
<cr>
```

## Configure the query-identity Option for Retrieval of EAP Identity

The Microsoft Windows 7 IKEv2 client sends an IP address as the Internet Key Exchange (IKE) identity that prevents the Cisco ASA server from using it efficiently for tunnel-group lookup. The ASA must be configured with the **query-identity** option for EAP authentication to allow the ASA to retrieve a valid EAP identity from the client.

For certificate-based authentication, the ASA 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.

You can obtain the certificates from the Microsoft Certificate Server or other CA server.

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

**Note** DHCP intercept is supported for IKEv2 to allow Windows to do split-tunneling. This feature only works with IPv4 split-tunneling attributes.

### Procedure

**Step 1** To set the connection type to IPsec remote access, enter the **tunnel-group** command. The syntax is **tunnel-group** *name* **type** *type,* where name is the name you assign to the tunnel group, and type is the type of tunnel:

In the following example, the IKEv2 preshared key is configured as 44kkaol59636jnfx:

```
hostname(config-tunnel-ipsec)# ikev2 local-authentication pre-shared-key 44kkaol59636jnfx
```

**Note**
You must configure the **ikev2 remote-authentication pre-shared-key** command or **ikev2 remote-authentication certificate** command to complete the authentication.

**Step 2** To specify Extensible Authentication Protocol (EAP) as the method that supports user authentication with standards-based, third-party IKEv2 remote access clients, use the **ikev2 remote-authentication eap** [**query-identity**] command.

**Note**
Before you can enable EAP for remote authentication, you must configure local authentication using a certificate and configure a valid trustpoint using the **ikev2 local-authentication** {**certificate** *trustpoint*} command. Otherwise, the EAP authentication request is rejected.

You may configure multiple options that allow the client to use any of the configured options, but not all, for remote authentication.

For IKEv2 connections, the tunnel group mapping must know which authentication methods to allow for remote authentication (PSK, certificate, and EAP) and local authentication (PSK and certificate), and which trust point to use for local authentication. Currently, mapping is performed using the IKE ID, which is taken from the peer or peer certificate field value (using the certificate map). If both options fail, then the in-coming connection is mapped to the default remote access tunnel group, DefaultRAGroup. A certificate map is an applicable option only when the remote peer is authenticated via a certificate. This map allows mapping to different tunnel groups. For certificate authentication only, the tunnel group lookup is performed using rules or using the default setting. For EAP and PSK authentication, the tunnel group lookup is performed using the IKE ID on the client (it matches the tunnel group name) or using the default setting.

For EAP authentication, you must use the DefaultRAGroup tunnel group unless the client allows the IKE ID and username to be configured independently.

The following example shows an EAP request for authentication being denied:

```
ciscoasa(config-tunnel-ipsec)# ikev2 remote-authentication eap query-identity
ciscoasa(config-tunnel-ipsec)# ikev2 remote-authentication certificate
ciscoasa(config-tunnel-ipsec)# ikev2 local-authentication pre-shared-key 12345678
ERROR: The local-authentication method is required to be certificate based
if remote-authentication allows EAP
ciscoasa(config-tunnel-ipsec)# ikev2 local-authentication certificate myIDcert
```

**Step 3** Save your changes.

```
hostname(config)# write memory
```

```
hostname(config)#
```

To verify that the tunnel is up and running, use the **show vpn-sessiondb summary** or **show crypto ipsec sa** command.

# Configure Microsoft Active Directory Settings for Password Management

If you are using an LDAP directory server for authentication, password management is supported with the Sun Microsystems JAVA System Directory Server (formerly named the Sun ONE Directory Server) and the Microsoft Active Directory.

- Sun—The DN configured on the ASA to access a Sun directory server must be able to access the default password policy on that server. We recommend using the directory administrator, or a user with directory administrator privileges, as the DN. Alternatively, you can place an ACI on the default password policy.
- Microsoft—You must configure LDAP over SSL to enable password management with Microsoft Active Directory.

To use password management with Microsoft Active Directory, you must set certain Active Directory parameters as well as configuring password management on the ASA. This section describes the Active Directory settings associated with various password management actions. These descriptions assume that you have also enabled password management on the ASA and configured the corresponding password management attributes. The specific steps in this section refer to Active Directory terminology under Windows 2000. This section assumes that you are using an LDAP directory server for authentication.

## Use Active Directory to Force the User to Change Password at Next Logon

To force a user to change the user password at the next logon, specify the **password-management** command in tunnel-group general-attributes configuration mode on the ASA and perform the following steps under Active Directory:

### Procedure

**Step 1** Choose **Start** > **Programs** > **Administrative Tools** > **Active Directory Users and Computers**.

**Step 2** Right-click to choose **Username** > **Properties** > **Account**.

**Step 3** Check the **User must change password at next logon** check box.

The next time this user logs on, the ASA displays the following prompt: “New password required. Password change required. You must enter a new password with a minimum length *n* to continue.” You can set the minimum required password length, *n*, as part of the Active Directory configuration at Start > Programs > Administrative Tools > Domain Security Policy > Windows Settings > Security Settings > Account Policies > Password Policy. Select **Minimum password length**.

## Use Active Directory to Specify Maximum Password Age

To enhance security, you can specify that passwords expire after a certain number of days. To specify a maximum password age for a user password, specify the **password-management** command in tunnel-group general-attributes configuration mode on the ASA and perform the following steps under Active Directory:



**Note** The **radius-with-expiry** command, formerly configured as part of tunnel-group remote-access configuration to perform the password age function, is deprecated. The **password-management** command, entered in tunnel-group general-attributes mode, replaces it.

### Procedure

**Step 1** Choose **Start** > **Programs** > **Administrative Tools** > **Domain Security Policy** > **Windows Settings** > **Security Settings** > **Account Policies** > **Password Policy**.

**Step 2** Double-click Maximum password age.

**Step 3** Check the **Define this policy setting** check box and specify the maximum password age, in days, that you want to allow.

## Use Active Directory to Enforce Minimum Password Length

To enforce a minimum length for passwords, specify the **password-management** command in tunnel-group general-attributes configuration mode on the ASA and perform the following steps under Active Directory:

### Procedure

**Step 1** Chose **Start** > **Programs** > **Administrative Tools** > **Domain Security Policy**.

**Step 2** Chose **Windows Settings** > **Security Settings** > **Account Policies** > **Password Policy**.

**Step 3** Double-click **Minimum Password Length**.

**Step 4** Check the **Define this policy setting** check box and specify the minimum number of characters that the password must contain.

## Use Active Directory to Enforce Password Complexity

To enforce complex passwords—for example, to require that a password contain upper- and lowercase letters, numbers, and special characters—enter the **password-management** command in tunnel-group general-attributes configuration mode on the ASA and perform the following steps under Active Directory:

**Procedure**

**Step 1** Choose **Start** > **Programs** > **Administrative Tools** > **Domain Security Policy. Select Windows Settings** > **Security Settings** > **Account Policies** > **Password Policy**.

**Step 2** Double-click Password must meet complexity requirements to open the Security Policy Setting dialog box.

**Step 3** Check the Define this policy setting check box and select **Enable**.

Enforcing password complexity takes effect only when the user changes passwords; for example, when you have configured Enforce password change at next login or Password expires in *n* days. At login, the user receives a prompt to enter a new password, and the system will accept only a complex password.

# Configure the Connection Profile for RADIUS/SDI Message Support for the Secure Client

This section describes procedures to ensure that the AnyConnect VPN client using RSA SecureID Software tokens can properly respond to user prompts delivered to the client through a RADIUS server proxying to an SDI server(s).

**Note** If you have configured the double-authentication feature, SDI authentication is supported only on the primary authentication server.

When a remote user connects to the ASA with the AnyConnect VPN client and attempts to authenticate using an RSA SecurID token, the ASA communicates with the RADIUS server, which in turn, communicates with the SDI server about the authentication.

During authentication, the RADIUS server presents access challenge messages to the ASA. Within these challenge messages are reply messages containing text from the SDI server. The message text is different when the ASA is communicating directly with an SDI server than when communicating through the RADIUS proxy. Therefore, in order to appear as a native SDI server to the Secure Client, the ASA must interpret the messages from the RADIUS server.

Also, because the SDI messages are configurable on the SDI server, the message text on the ASA must match (in whole or in part) the message text on the SDI server. Otherwise, the prompts displayed to the remote client user may not be appropriate for the action required during authentication. The Secure Client may fail to respond and authentication may fail.

Configure the Security Appliance to Support RADIUS/SDI Messages, on page 128 describes how to configure the ASA to ensure successful authentication between the client and the SDI server.

## Configure the Security Appliance to Support RADIUS/SDI Messages

To configure the ASA to interpret SDI-specific RADIUS reply messages and prompt the Secure Client user for the appropriate action, perform the following steps:

### Procedure

**Step 1** Configure a connection profile (tunnel group) to forward RADIUS reply messages in a manner that simulates direct communication with an SDI server using the **proxy-auth sdi** command from tunnel-group webvpn configuration mode. Users authenticating to the SDI server must connect over this connection profile.

**Example:**

```
hostname(config)# tunnel-group sales webvpn attributes
hostname(tunnel-group-webvpn)# proxy-auth sdi
```

**Step 2** Configure the RADIUS reply message text on the ASA to match (in whole or in part) the message text sent by the RADIUS server with the **proxy-auth_map sdi** command from tunnel-group webvpn configuration mode.

The default message text used by the ASA is the default message text used by Cisco Secure Access Control Server (ACS). If you are using Cisco Secure ACS, and it is using the default message text, you do not need to configure the message text on the ASA. Otherwise, use the **proxy-auth_map sdi** command to ensure the message text matches.

The table below shows the message code, the default RADIUS reply message text, and the function of each message. Because the security appliance searches for strings in the order that they appear in the table, you must ensure that the string you use for the message text is not a subset of another string.

For example, “new PIN” is a subset of the default message text for both new-pin-sup and next-ccode-and-reauth. If you configure new-pin-sup as “new PIN,” when the security appliance receives “new PIN with the next card code” from the RADIUS server, it will match the text to the new-pin-sup code instead of the next-ccode-and-reauth code.

SDI Op-codes, Default Message Text, and Message Function

| Message Code | Default RADIUS Reply Message Text | Function |
|---|---|---|
| next-code | Enter Next PASSCODE | Indicates the user must enter the NEXT tokencode without the PIN. |
| new-pin-sup | Please remember your new PIN | Indicates the new system PIN has been supplied and displays that PIN for the user. |
| new-pin-meth | Do you want to enter your own pin | Requests from the user which new PIN method to use to create a new PIN. |
| new-pin-req | Enter your new Alpha-Numerical PIN | Indicates a user-generated PIN and requests that the user enter the PIN. |
| new-pin-reenter | Reenter PIN: | Used internally by the ASA for user-supplied PIN confirmation. The client confirms the PIN without prompting the user. |
| new-pin-sys-ok | New PIN Accepted | Indicates the user-supplied PIN was accepted. |

| Message Code | Default RADIUS Reply Message Text | Function |
|---|---|---|
| next-ccode-and-reauth | new PIN with the next card code | Follows a PIN operation and indicates the user must wait for the next tokencode and to enter both the new PIN and next tokencode to authenticate. |
| ready-for-sys- pin | ACCEPT A SYSTEM GENERATED PIN | Used internally by the ASA to indicate the user is ready for the system-generated PIN. |

The following example enters aaa-server-host mode and changes the text for the RADIUS reply message new-pin-sup:

```
hostname(config)# aaa-server radius_sales host 10.10.10.1
hostname(config-aaa-server-host)# proxy-auth_map sdi new-pin-sup "This is your new PIN"
```

# Group Policies

This section describes group policies and how to configure them.

A group policy is a set of user-oriented attribute/value pairs for IPsec connections that are stored either internally (locally) on the device or externally on a RADIUS server. The connection profile uses a group policy that sets terms for user connections after the tunnel is established. Group policies let you apply whole sets of attributes to a user or a group of users, rather than having to specify each attribute individually for each user.

Enter the **group-policy** commands in global configuration mode to assign a group policy to users or to modify a group policy for specific users.

The ASA includes a default group policy. In addition to the default group policy, which you can modify but not delete, you can create one or more group policies specific to your environment.

You can configure internal and external group policies. Internal groups are configured on the ASA's internal database. External groups are configured on an external authentication server, such as RADIUS. Group policies include the following attributes:

- Identity
- Server definitions
- Client firewall settings
- Tunneling protocols
- IPsec settings
- Hardware client settings
- Filters
- Client configuration settings

- Connection settings

# Modify the Default Group Policy

The ASA supplies a default group policy. You can modify this default group policy, but you cannot delete it. A default group policy, named DfltGrpPolicy, always exists on the ASA, but this default group policy does not take effect unless you configure the ASA to use it. When you configure other group policies, any attribute that you do not explicitly specify inherits its value from the default group policy.

**Note** Secure Client profiles, including any or all Secure Client Profile Types (such as Network Access Manager, Umbrella, and so on), that are configured on (and then assigned to) the DfltGrpPolicy, are not inherited by other group policies, unless the other group policies explicitly are configured to inherit from the DfltGrpPolicy. In other words, Secure Client profiles that are associated with the DfltGrpPolicy are not inherited when specific Secure Client profiles are configured on a group policy.

To view the default group policy, enter the following command:

```
hostname(config)# show running-config all group-policy DfltGrpPolicy
hostname(config)#
```

To configure the default group policy, enter the following command:

```
hostname(config)# group-policy DfltGrpPolicy internal
hostname(config)#
```

**Note** The default group policy is always internal. Despite the fact that the command syntax is hostname(config)# group-policy DfltGrpPolicy {internal | external}, you cannot change its type to external.

To change any of the attributes of the default group policy, use the **group-policy attributes** command to enter attributes mode, then specify the commands to change whatever attributes that you want to modify:

```
hostname(config)# group-policy DfltGrpPolicy attributes
```

**Note** The attributes mode applies only to internal group policies.

The default group policy, DfltGrpPolicy, that the ASA provides is as follows:

```
hostname# show run all group-policy DfltGrpPolicy
group-policy DfltGrpPolicy internal
group-policy DfltGrpPolicy attributes
 banner none
 wins-server none
 dns-server value 10.10.10.1.1
```

```
dhcp-network-scope none
vpn-access-hours none
vpn-simultaneous-logins 3
vpn-idle-timeout 30
vpn-idle-timeout alert-interval 1
vpn-session-timeout none
vpn-session-timeout alert-interval 1
vpn-filter none
vpn-tunnel-protocol ikev1 ikev2 l2tp-ipsec ssl-client

password-storage disable
ip-comp disable
re-xauth disable
group-lock none
pfs disable
ipsec-udp disable
ipsec-udp-port 10000
split-tunnel-policy tunnelall
ipv6-split-tunnel-policy tunnelall
split-tunnel-network-list none
default-domain value cisco.com
split-dns none
split-tunnel-all-dns disable
intercept-dhcp 255.255.255.255 disable
secure-unit-authentication disable
user-authentication disable
user-authentication-idle-timeout 30
ip-phone-bypass disable
client-bypass-protocol disable
gateway-fqdn none
leap-bypass disable
nem disable
backup-servers keep-client-config
msie-proxy server none
msie-proxy method no-modify
msie-proxy except-list none
msie-proxy local-bypass disable
msie-proxy pac-url none
msie-proxy lockdown enable
vlan none
nac-settings none
address-pools none
ipv6-address-pools none
smartcard-removal-disconnect enable
scep-forwarding-url none
client-firewall none
client-access-rule none
webvpn
 url-list none
 filter none
 homepage none
 html-content-filter none

 http-proxy disable

 anyconnect ssl dtls enable
 anyconnect mtu 1406
 anyconnect firewall-rule client-interface private none
 anyconnect firewall-rule client-interface public none
 anyconnect keep-installer installed
 anyconnect ssl keepalive 20
 anyconnect ssl rekey time none
 anyconnect ssl rekey method none
 anyconnect dpd-interval client 30
```

```
  anyconnect dpd-interval gateway 30
  anyconnect ssl compression none
  anyconnect dtls compression lzs
  anyconnect modules none
  anyconnect profiles none
  anyconnect ask none
  customization none
  keep-alive-ignore 4
  http-comp gzip
  download-max-size 2147483647
  upload-max-size 2147483647
  post-max-size 2147483647
  user-storage none
  storage-objects value cookies,credentials
  storage-key none
  hidden-shares none

  activex-relay enable
  unix-auth-uid 65534
  unix-auth-gid 65534
  file-entry enable
  file-browsing enable
  url-entry enable
  deny-message value Login was successful, but because certain criteria have not been met
or due to some specific group policy, you do not have permission to use any of the VPN
features. Contact your IT administrator for more information

  anyconnect ssl df-bit-ignore disable
  anyconnect routing-filtering-ignore disable

  always-on-vpn profile-setting
```

You can modify the default group policy, and you can also create one or more group policies specific to your environment.

# Configure Group Policies

A group policy can apply to any kind of tunnel. In each case, if you do not explicitly define a parameter, the group takes the value from the default group policy.

You can perform these configuration tasks in both single context mode or multiple-context mode:

**Note** Multiple-context mode applies only to IKEv2 and IKEv1 site to site and does not apply to AnyConnect, Clientless SSL VPN, the Apple native VPN client, the Microsoft native VPN client, or cTCP for IKEv1 IPsec.

## Configure an External Group Policy

External group policies take their attribute values from the external server that you specify. For an external group policy, you must identify the AAA server group that the ASA can query for attributes and specify the password to use when retrieving attributes from the external AAA server group. If you are using an external authentication server, and if your external group-policy attributes exist in the same RADIUS server as the users that you plan to authenticate, you have to make sure that there is no name duplication between them.

**Note** External group names on the ASA refer to user names on the RADIUS server. In other words, if you configure external group X on the ASA, the RADIUS server sees the query as an authentication request for user X. So external groups are really just user accounts on the RADIUS server that have special meaning to the ASA. If your external group attributes exist in the same RADIUS server as the users that you plan to authenticate, there must be no name duplication between them.

The ASA supports user authorization on an external LDAP or RADIUS server. Before you configure the ASA to use an external server, you must configure the server with the correct ASA authorization attributes and, from a subset of these attributes, assign specific permissions to individual users. Follow the instructions in Configure an External AAA Server for VPN, on page 291 to configure your external server.

### Procedure

To configure an external group policy, perform the following step and specify a name and type for the group policy, along with the server-group name and a password:

```
hostname(config)# group-policy group_policy_name type server-group server_group_name password
 server_password
hostname(config)#
```

**Note**
For an external group policy, RADIUS is the only supported AAA server type.

For example, the following command creates an external group policy named ExtGroup that gets its attributes from an external RADIUS server named ExtRAD and specifies that the password to use when retrieving the attributes is newpassword:

```
hostname(config)# group-policy ExtGroup external server-group ExtRAD password newpassword
hostname(config)#
```

**Note**
You can configure several vendor-specific attributes (VSAs), as described in Configure an External AAA Server for VPN, on page 291. If a RADIUS server is configured to return the Class attribute (#25), the ASA uses that attribute to authenticate the Group Name. On the RADIUS server, the attribute must be formatted as: OU=*groupname*; where *groupname* is identical to the Group Name configured on the ASA—for example, OU=Finance.

## Create an Internal Group Policy

To configure an internal group policy, enter configuration mode, use the group-policy command, specify a name, and the **internal** type for the group policy:

```
hostname(config)# group-policy group_policy_name internal
hostname(config)#
```

For example, the following command creates the internal group policy named GroupPolicy1:

```
hostname(config)# group-policy GroupPolicy1 internal
hostname(config)#
```

**Note** You cannot change the name of a group policy after you create it.

You can configure the attributes of an internal group policy by copying the values of a preexisting group policy by appending the keyword **from** and specifying the name of the existing policy:

```
hostname(config)# group-policy group_policy_name internal from group_policy_name
hostname(config-group-policy)#
```

For example, the following command creates the internal group policy named GroupPolicy2 by copying the attributes of GroupPolicy1:

```
hostname(config)# group-policy GroupPolicy2 internal from GroupPolicy1
hostname(config-group-policy)#
```

# Configure General Internal Group Policy Attributes

## Group Policy Name

The group policy name was chosen when the internal group policy was created. You cannot change the name of a group policy once it has been created. See Create an Internal Group Policy, on page 134 for more information.

## Configure the Group Policy Banner Message

Specify the banner, or welcome message, if any, that you want to display. The default is no banner. The message that you specify is displayed on remote clients when they connect. To specify a banner, enter the **banner** command in group-policy configuration mode. The banner text can be up to 500 characters long.

**Note** Ensure that you use normal line breaks in the banner dialog and not “\n”.

The overall banner length, which is displayed during post-login on the VPN remote client, has increased from 510 to 4000 characters in ASA version 9.5.1.

**Note** A carriage-return and line-feed included in the banner counts as two characters.

To delete a banner, enter the **no** form of this command. Be aware that using the **no** version of the command deletes all banners for the group policy.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying a value for the banner string, as follows:

```
hostname(config-group-policy)# banner {value banner_string | none}
```

The following example shows how to create a banner for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# banner value Welcome to Cisco Systems ASA 9.0.
```

## Specify Address Pools for Remote Access Connections

When remote access clients connect to the ASA, the ASA can assign the client an IPv4 or IPv6 address based on the group-policy specified for the connection.

You can specify a list of up to six local address pools to use for local address allocation. The order in which you specify the pools is significant. The ASA allocates addresses from these pools in the order in which the pools appear in this command.

## Assign an IPv4 Address Pool to an Internal Group Policy

### Before you begin

Create the IPv4 address pool.

### Procedure

**Step 1** Enter group policy configuration mode.

**group-policy** *value* **attributes**

**Example:**

```
hostname> en
hostname# config t
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)#
```

**Step 2** Assign the address pool named ipv4-pool1, ipv4-pool2, and ipv4pool3 to the FirstGroup group policy. You are allowed to specify up to 6 address pools for group-policy.

**address-pools value** *pool-name1 pool-name2 pool-name6*

**Example:**

```
asa4(config-group-policy)# address-pools value ipv4-pool1 ipv4-pool2 ipv4-pool3
asa4(config-group-policy)#
```

**Step 3** (Optional) Use the **no address-pools value pool-name** command to remove the address-pools from the group policy configuration and return the address pool setting to inherit the address pool information from other sources such as the DefltGroupPolicy.

**no address-pools value** *pool-name1 pool-name2 pool-name6*

**Example:**

```
hostname(config-group-policy)# no address-pools value ipv4-pool1 ipv4-pool2 ipv4-pool3
hostname(config-group-policy)#
```

**Step 4** (Optional) The **address-pools none** command disables this attribute from being inherited from other sources of policy, such as the DefltGrpPolicy.

```
hostname(config-group-policy)# address-pools none
hostname(config-group-policy)#
```

**Step 5** (Optional) The **no address pools none** command removes the **address-pools none** command from the group policy, restoring the default value, which is to allow inheritance.

```
hostname(config-group-policy)# no address-pools none
hostname(config-group-policy)#
```

## Assign an IPv6 Address Pool to an Internal Group Policy

**Before you begin**

Create the IPv6 address pool. See IP Addresses for VPNs, on page 185.

**Procedure**

**Step 1** Enter group policy configuration mode.

**group-policy** *value* **attributes**

**Example:**

```
hostname> en
hostname# config t
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)#
```

**Step 2** Assign the address pool named ipv6-pool to the FirstGroup group policy. You can assign up to six ipv6 address pools to a group policy.

**Example:**

This example shows ipv6-pool1, ipv6-pool2, and ipv6-pool3 being assigned to the FirstGroup group policy.

```
hostname(config-group-policy)# ipv6-address-pools value ipv6-pool1 ipv6-pool2 ipv6-pool3
hostname(config-group-policy)#
```

**Step 3** (Optional) Use the **no ipv6-address-pools value pool-name** command to remove the address-pools from the group policy configuration and return the address pool setting to inherit the address pool information from other sources such as the DfltGroupPolicy.

**no ipv6-address-pools value pool-name1 pool-name2 pool-name6**

**Example:**

```
hostname(config-group-policy)# no ipv6-address-pools value ipv6-pool1 ipv6-pool2 ipv6-pool3
hostname(config-group-policy)#
```

**Step 4** (Optional) Use the **ipv6-address-pools none** command to disable this attribute from being inherited from other sources of policy, such as the DfltGrpPolicy.

```
hostname(config-group-policy)# ipv6-address-pools none
hostname(config-group-policy)#
```

**Step 5** (Optional) Use the **no ipv6-address pools none** command to remove the **ipv6-address-pools none** command from the group policy, restoring the default value, which is to allow inheritance.

```
hostname(config-group-policy)# no ipv6-address-pools none
hostname(config-group-policy)#
```

## Specify the Tunneling Protocol for the Group Policy

Specify the VPN tunnel type for this group policy by entering the **vpn-tunnel-protocol**{ ikev1 | ikev2 | l2tp-ipsec | ssl-client} command from group-policy configuration mode.

The default value is to inherit the attributes of the Default Group Policy. To remove the attribute from the running configuration, enter the **no** form of this command.

The parameter values for this command include:

- ikev1—Negotiates an IPsec IKEv1 tunnel between two peers (the Cisco VPN Client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- ikev2—Negotiates an IPsec IKEv2 tunnel between two peers (the Secure Client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- **l2tp-ipsec**—Negotiates an IPsec tunnel for an L2TP connection.
- ssl-client—Negotiates an SSL tunnel using TLS or DTLS with the Secure Client.

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure the IPsec IKEv1 tunneling mode for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
```

```
hostname(config-group-policy)# vpn-tunnel-protocol ikev1
hostname(config-group-policy)#
```

## Specify a VLAN for Remote Access or Apply a Unified Access Control Rule to the Group Policy

Filters consist of rules that determine whether to allow or reject tunneled data packets coming through the ASA, based on criteria such as source address, destination address, and protocol. You can specify an IPv4 or IPv6 unified access control list for your group policy or allow it to inherit the ACLs specified in the Default Group Policy.

Choose one of the following options to specify an egress VLAN (also called "VLAN mapping") for remote access or specify an ACL to filter the traffic:

**Note** When doing VLAN mapping with IPv6, the outside (destination) address must be unique for each of the VLANs so that decrypted traffic is routed to inside networks. You cannot have the same destination network with different VLANs and route metrics.

- Enter the following command in group-policy configuration mode to specify the egress VLAN for remote access VPN sessions assigned to this group policy or to a group policy that inherits this group policy:

  [**no**] **vlan** {*vlan_id* |**none**}

  *no vlan* removes the *vlan_id* from the group policy. The group policy inherits the vlan value from the default group policy.

  *none* removes the *vlan_id* from the group policy and disables VLAN mapping for this group policy. The group policy does not inherit the vlan value from the default group policy.

  *vlan_id* is the number of the VLAN, in decimal format, to assign to remote access VPN sessions that use this group policy. The VLAN must be configured on this ASA per the instructions in the "Configuring VLAN Subinterfaces and 802.1Q Trunking" in the general operations configuration guide.

**Note** For clientless VPN connections, the egress VLAN feature works for only HTTP protocol.

- Specify the name of the access control rule (ACL) to apply to VPN session, using the **vpn-filter** command in group policy mode. You can specify an IPv4 or IPv6 ACL using the vpn-filter command.

**Note** You can also configure this attribute in username mode, in which case the value configured under username supersedes the group-policy value.

```
hostname(config-group-policy)# vpn-filter {value ACL name | none}
hostname(config-group-policy)#
```

You configure ACLs to permit or deny various types of traffic for this group policy. You then enter the **vpn-filter** command to apply those ACLs.

To remove the ACL, including a null value created by entering the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from another group policy.

A group policy can inherit this value from another group policy. To prevent inheriting a value, enter the **none** keyword instead of specifying an ACL name. The **none** keyword indicates that there is no ACL and sets a null value, thereby disallowing an ACL.

The following example shows how to set a filter that invokes an ACL named acl_vpn for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-filter acl_vpn
hostname(config-group-policy)#
```

A **vpn-filter** command is applied to post-decrypted traffic after it exits a tunnel and pre-encrypted traffic before it enters a tunnel. An ACL that is used for a vpn-filter should not also be used for an interface access-group. When a **vpn-filter** command is applied to a group policy that governs Remote Access VPN client connections, the ACL should be configured with the client assigned IP addresses in the **src_ip** position of the ACL and the local network in the **dest_ip** position of the ACL.

When a **vpn-filter** command is applied to a group-policy that governs a LAN to LAN VPN connection, the ACL should be configured with the remote network in the **src_ip** position of the ACL and the local network in the **dest_ip** position of the ACL.

Caution should be used when constructing the ACLs for use with the vpn-filter feature. The ACLs are constructed with the post-decrypted traffic in mind. However, ACLs are also applied to the traffic in the opposite direction. For this pre-encrypted traffic that is destined for the tunnel, the ACLs are constructed with the **src_ip** and **dest_ip** positions swapped.

Also note that the VPN filter applies to initial connections only. It does not apply to secondary connections, such as a SIP media connection, that are opened due to the action of application inspection.

In the following example, the vpn-filter is used with a Remote Access VPN client. This example assumes that the client assigned IP address is 10.10.10.1/24 and the local network is 192.168.1.0/24.

The following ACE allows the Remote Access VPN client to telnet to the local network:

```
hostname(config-group-policy)# access-list vpnfilt-ra permit 10.10.10.1 255.255.255.255
192.168.1.0 255.255.255.0 eq 23
```

The following ACE allows the local network to telnet to the Remote Access client:

```
hostname(config-group-policy)# access-list vpnfilt-ra permit 10.10.10.1 255.255.255.255 eq
 23 192.168.1.0 255.255.255.0
```



**Note** The ACE **access-list vpnfilt-ra permit 10.10.10.1 255.255.255.255 192.168.1.0 255.255.255.0 eq 23** allows the local network to initiate a connection to the Remote Access client on any TCP port if it uses a source port of 23. The ACE **access-list vpnfilt-ra permit 10.10.10.1 255.255.255.255 eq 23 192.168.1.0 255.255.255.0** allows the Remote Access client to initiate a connection to the local network on any TCP port if it uses a source port of 23.

In the next example, the vpn-filter is used with a LAN to LAN VPN connection. This example assumes that the remote network is 10.0.0.0/24 and the local network is 192.168.1.0/24. The following ACE allows remote network to telnet to the local network:

```
hostname(config-group-policy)# access-list vpnfilt-l2l permit 10.0.0.0 255.255.255.0
192.168.1.0 255.255.255.0 eq 23
```

The following ACE allows the local network to telnet to the remote network:

```
hostname(config-group-policy)# access-list vpnfilt-l2l permit 10.0.0.0 255.255.255.0 eq 23
 192.168.1.0 255.255.255.0
```



**Note** The ACE **access-list vpnfilt-l2l permit 10.0.0.0 255.255.255.0 192.168.1.0 255.255.255.0 eq 23** allows the local network to initiate a connection to the remote network on any TCP port if it uses a source port of 23. The ACE **access-list vpnfilt-l2l permit 10.0.0.0 255.255.255.0 eq 23 192.168.1.0 255.255.255.0** allows the remote network to initiate a connection to the local network on any TCP port if it uses a source port of 23.

## Specify VPN Access Hours for a Group Policy

### Before you begin

Create a time range. See the "Configuring Time Ranges" in the general operations configuration guide.

### Procedure

**Step 1** Enter group policy configuration mode.

**group-policy** *value* **attributes**

**Example:**

```
hostname> en
hostname# config t
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)#
```

**Step 2** You can set the VPN access hours by associating a configured time-range policy with a group policy using the **vpn-access-hours** command in group-policy configuration mode. This command assigns a VPN access time range named business-hours to the group policy named FirstGroup.

A group policy can inherit a time-range value from a default or specified group policy. To prevent this inheritance, enter the **none** keyword instead of the name of a time-range in this command. This keyword sets VPN access hours to a null value, which allows no time-range policy.

**vpn-access-hours value** {*time-range-name* | **none**}

**Example:**

```
hostname(config-group-policy)# vpn-access-hours value business-hours
hostname(config-group-policy)#
```

## Specify Simultaneous VPN Logins for a Group Policy

You can set a limit on the number of simultaneous sessions a given user can maintain for a group policy. The default is 3 simultaneous sessions.

Stale Secure Client, IPsec Client, or Clientless sessions (sessions that are terminated abnormally) might remain in the session database, even though a "new" session has been established with the same username.

If the allowed number of simultaneous sessions is 1, and the same user logs in again after an abnormal termination, then the stale session is removed from the database, and the new session is established. If, however, the existing session is still an active connection and the same user logs in again, perhaps from another PC, the first session is logged off and removed from the database, and the new session is established.

If the number of allowed simultaneous sessions is greater than 1, then, when the user has reached that maximum number and tries to log in again, the session with the longest idle time is logged off. If all current sessions have been idle an equally long time, then the oldest session is logged off. This action frees up a session and allows the new login.

Once the maximum session limit is reached, it takes some time for the system to delete the oldest session. Thus, a user might not be able to immediately log on and might have to retry the new connection before it completes successfully. This should not be a problem if users log off their sessions as expected. You can optionally remove the delay by configuring the system to not wait for the deletion to complete and immediately allow the new user connection.

### Procedure

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | Specify the number of simultaneous logins allowed for any user, using the **vpn-simultaneous-logins** *integer* command in group-policy configuration mode. | **vpn-simultaneous-logins** *integer*<br>The default value is 3. The range is an integer from 0 through 2147483647. A group policy can inherit this value from another group policy. Enter 0 to disable login and prevent user access. The following example shows how to allow a maximum of 4 simultaneous logins for the group policy named FirstGroup:<br>`hostname(config)# group-policy FirstGroup attributes`<br>`hostname(config-group-policy)# vpn-simultaneous-logins 4`<br>**Note**<br>• While the maximum limit for the number of simultaneous logins is very large, allowing several simultaneous logins could compromise security and affect performance. |

| | Command or Action | Purpose |
|---|---|---|
| | | • When you connect to different tunnel groups with different group policies, **vpn-simultaneous-logins** deletes user sessions even if the existing sessions use a different group policy. |
| **Step 2** | (Optional.) When the simultaneous login limit is reached, configure the system to establish new sessions without waiting for the oldest session to be deleted. | **vpn-simultaneous-login-delete-no-delay**<br>This option is disabled by default.<br>`hostname(config)# group-policy FirstGroup attributes`<br>`hostname(config-group-policy)# vpn-simultaneous-login-delete-no-delay` |

## Restrict Access to a Specific Connection Profile

Specify whether to restrict remote users to access only through the connection profile, using the **group-lock** command in group-policy configuration mode.

```
hostname(config-group-policy)# group-lock {value tunnel-grp-name | none}
hostname(config-group-policy)# no group-lock
hostname(config-group-policy)#
```

The *tunnel-grp-name* variable specifies the name of an existing connection profile that the ASA requires for the user to connect. Group-lock restricts users by checking if the group configured in the VPN client is the same as the connection profile to which the user is assigned. If it is not, the ASA prevents the user from connecting. If you do not configure group-lock, the ASA authenticates users without regard to the assigned group. Group locking is disabled by default.

To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

To disable group-lock, enter the **group-lock** command with the **none** keyword. The none keyword sets group-lock to a null value, thereby allowing no group-lock restriction. It also prevents inheriting a group-lock value from a default or specified group policy

## Specify the Maximum VPN Connection Time in a Group Policy

### Procedure

**Step 1** (Optional) Configure a maximum amount of time for VPN connections, using the **vpn-session-timeout** {*minutes* command in group-policy configuration mode or in username configuration mode.

The minimum time is 1 minute, and the maximum time is 35791394 minutes. There is no default value. At the end of this period of time, the ASA terminates the connection.

The following example shows how to set a VPN session timeout of 180 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-session-timeout 180
hostname(config-group-policy)#
```

The following example shows how to set a VPN session timeout of 180 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-session-timeout 180
hostname(config-username)#
```

Other actions using the [**no**] **vpn-session-timeout** {*minutes* | **none**} command:

- To remove the attribute from this policy and allow inheritance, enter the **no vpn-session-timeout** form of this command.
- To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter **vpn-session-timeout none**.

**Step 2** Configure the time at which a session timeout alert message is displayed to the user using the **vpn-session-timeout alert-interval** {*minutes* | } command.

This alert message tells users how many minutes left until their VPN session is automatically disconnected. The following example shows how to specify that users will be notified 20 minutes before their VPN session is disconnected. You can specify a range of 1-30 minutes.

```
hostname(config-webvpn)# vpn-session-timeout alert-interval 20
```

Other actions using the [**no**] **vpn-session-timeout alert-interval** {*minutes* | **none**} command:

- Use the no form of the command to indicate that the VPN session timeout alert-interval attribute will be inherited from the Default Group Policy:

  ```
  hostname(config-webvpn)# no vpn-session-timeout alert-interval
  ```

- The **vpn-session-timeout alert-interval none** indicates that users will not receive an alert.

## Specify a VPN Session Idle Timeout for a Group Policy

### Procedure

**Step 1** (Optional) To configure a VPN idle timeout period use the **vpn-idle-timeout** *minutes* command in group-policy configuration mode or in username configuration mode.

If there is no communication activity on the connection in this period, the ASA terminates the connection. The minimum time is 1 minute, the maximum time is 35791394 minutes, and the default is 30 minutes.

The following example shows how to set a VPN idle timeout of 15 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-idle-timeout 15
hostname(config-group-policy)#
```

Other actions using the [**no**] **vpn-idle-timeout** {*minutes* | **none**} command:

- Enter **vpn-idle-timeout none** to disable VPN idle timeout and pevent inheriting a timeout value.

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-idle-timeout none
hostname(config-group-policy)#
```

  This results in Secure Client (both SSL and IPsec/IKEv2) and Clientless VPN using the global webvpn **default-idle-timeout** *seconds* value. This command is entered in webvpn-config mode, for example: `hostnamee(config-webvpn)# default-idle-timeout 300`. The default is 1800 seconds (30 min), the range is 60-86400 seconds.

  For all webvon connections , the **default-idle-timeout** value is enforced only if **vpn-idle-timeout none** is set in the group policy/username attribute. A non-zero idle timeout value is required by ASA for all Secure Client connections.

  For Site-to-Site (IKEv1, IKEv2) and IKEv1 remote-access VPNs, we recommend you Disable timeout and allow for an unlimited idle period.

- To disable the idle timeout for this group policy or user policy, enter **no vpn-idle-timeout**. The value will be inherited.

- If you do not set **vpn-idle-timeout** at all, in anyway, the value is inherited, which defaults to 30 minutes.

**Note**
**vpn-idle-timeout** controls only the maximum time for a parent session. Child sessions (SSL/DTLS) are aggressively terminated much sooner by a hardcoded 5-minute TCP inactivity timeout or on failing the 3×Dead Peer Detection (DPD) check. For more details, see the note in Configure Dead Peer Detection. For further details on DPD, keepalives, and timeout attributes, see Answer AnyConnect FAQ - Tunnels, DPDs, and Inactivity Timer.

**Step 2** (Optional) You can optionally configure the time at which an idle timeout alert message is displayed to the user using the **vpn-idle-timeout alert-interval** {*minutes*} command.

This alert message tells users how many minutes they have left until their VPN session is disconnected due to inactivity. The default alert interval is one minute.

The following example shows how to set a VPN idle timeout alert interval of 3 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-idle-timeout alert-interval 3
hostname(config-username)#
```

Other actions using the [**no**] **vpn-idle-timeout alert-interval** {*minutes* | **none**} command:

- The **none** parameter indicates that users will not receive an alert.

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-idle-timeout none
hostname(config-username)#
```

- To remove the alert interval for this group or user policy enter **no vpn-idle-timeout alert-interval**. The value will be inherited.

- If you do not set this parameter at all, the default alert interval is one minute.

# Configure WINS and DNS Servers for a Group Policy

You can specify primary and secondary WINS servers and DNS servers. The default value in each case is none. To specify these servers, perform the following steps:

### Procedure

**Step 1** Specify the primary and secondary WINS servers:

```
hostname(config-group-policy)# wins-server value {ip_address [ip_address] | none}
hostname(config-group-policy)#
```

The first IP address specified is that of the primary WINS server. The second (optional) IP address is that of the secondary WINS server. Specifying the **none** keyword instead of an IP address sets WINS servers to a null value, which allows no WINS servers and prevents inheriting a value from a default or specified group policy.

Every time that you enter the **wins-server** command, you overwrite the existing setting. For example, if you configure WINS server x.x.x.x and then configure WINS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole WINS server. The same is true for multiple servers. To add a WINS server rather than overwrite previously configured servers, include the IP addresses of all WINS servers when you enter this command.

The following example shows how to configure WINS servers with the IP addresses 10.10.10.15 and 10.10.10.30 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# wins-server value 10.10.10.15 10.10.10.30
hostname(config-group-policy)#
```

**Step 2** Specify the primary and secondary DNS servers:

```
hostname(config-group-policy)# dns-server value {ip_address [ip_address] | none}
hostname(config-group-policy)#
```

The first IP address specified is that of the primary DNS server. The second (optional) IP address is that of the secondary DNS server. Specifying the **none** keyword instead of an IP address sets DNS servers to a null value, which allows no DNS servers and prevents inheriting a value from a default or specified group policy. You can specify up to four DNS server addresses: up to two IPv4 addresses and two IPv6 addresses.

Every time that you enter the **dns-server** command, you overwrite the existing setting. For example, if you configure DNS server x.x.x.x and then configure DNS server y.y.y.y, the second command overwrites the first, and y.y.y.y becomes the sole DNS server. The same is true for multiple servers. To add a DNS server rather than overwrite previously configured servers, include the IP addresses of all DNS servers when you enter this command.

The following example shows how to configure DNS servers with the IP addresses 10.10.10.15, 10.10.10.30, 2001:DB8::1, and 2001:DB8::2 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# dns-server value 10.10.10.15 10.10.10.30
2001:DB8::1 2001:DB8::2
hostname(config-group-policy)#
```

**Step 3** If there is no default domain name specified in the **DefaultDNS DNS** server group, you must specify a default domain. Use the domain name and top level domain for example, **example.com.**

```
asa4(config)# group-policy FirstGroup attributes
asa4(config-group-policy)# default-domain value example.com
asa4(config-group-policy)#
```

**Step 4** (Optional.) Configure the DHCP network scope:

**dhcp-network-scope** {*ip_address* | **none**}

If you configure DHCP servers for the address pool in the connection profile, the DHCP scope identifies the subnets to use for the pool for this group. The DHCP server must also have addresses in the same subnet identified by the scope. The scope allows you to select a subset of the address pools defined in the DHCP server to use for this specific group.

If you do not define a network scope, the DHCP server assigns IP addresses in the order of the address pools configured. It goes through the pools until it identifies an unassigned address.

To specify a scope, enter a routeable address on the same subnet as the desired pool, but not within the pool. The DHCP server determines which subnet this IP address belongs to and assigns an IP address from that pool.

We recommend using the IP address of an interface whenever possible for routing purposes. For example, if the pool is 10.100.10.2-10.100.10.254, and the interface address is 10.100.10.1/24, use 10.100.10.1 as the DHCP scope. Do not use the network number. You can use DHCP for IPv4 addressing only. If the address you choose is not an interface address, you might need to create a static route for the scope address.

Specifying **none** prevents DHCP address assignment, for example, from a default or inherited group policy.

**Example:**

The following example enters attribute configuration mode for FirstGroup and sets the DHCP scope to 10.100.10.1.

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# dhcp-network-scope 10.100.10.1
```

## Set the Split-Tunneling Policy

Set the rules for tunneling traffic by specifying the split-tunneling policy for IPv4 traffic:

**split-tunnel-policy** {**tunnelall** | **tunnelspecified** | **excludespecified**}

**no split-tunnel-policy**

Set the rules for tunneling traffic by specifying the split-tunneling policy for IPv6 traffic:

**ipv6-split-tunnel-policy** {**tunnelall** | **tunnelspecified** | **excludespecified**}

**no ipv6-split-tunnel-policy**

The policies options are:

- **tunnelspecified**— Tunnels all traffic to or from the networks specified in the Network List through the tunnel. Data to all other addresses travels in the clear and is routed by the remote user's Internet service provider.

  For versions of ASA 9.1.4 and higher, when you specify an include list, you can also specify an exclude list for a subnet inside the include range. Addresses in the excluded subnet will not be tunneled, and the rest of the include list will be. The networks in the exclusion list will not be sent over the tunnel. The exclusion list is specified using deny entries, and the inclusion list is specified using permit entries.

- **excludespecified**—Does not tunnel traffic to or from the networks specified in the Network List. Traffic from or to all other addresses is tunneled. . The VPN client profile that is active on the client must have Local LAN Access enabled. This option works with Secure Clients only.

**Note** Networks in the exclusion list that are not a subset of the include list are ignored by the client.

- **tunnelall —**Specifies that all traffic goes through the tunnel. This policy disables split tunneling. Remote users have access to the corporate network, but they do not have access to local networks. This is the default option.

**Note** Split tunneling is a traffic management feature, not a security feature. For optimum security, we recommend that you do not enable split tunneling.

### Example

The following examples shows how to set a split tunneling policy of tunneling only specified networks for the group policy named FirstGroup for IPv4 and IPv6:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-policy tunnelspecified

hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipv6-split-tunnel-policy tunnelspecified
```

## Specify a Network List for Split-Tunneling

In split tunneling, network lists determine what network traffic travels across the tunnel. Secure Client makes split tunneling decisions on the basis of a network list, which is an ACL.

```
hostname(config-group-policy)# split-tunnel-network-list {value access-list_name | none}
```

```
hostname(config-group-policy)# no split-tunnel-network-list value [access-list_name]
```

- **value** access-list name — identifies an ACL that enumerates the networks to tunnel or not tunnel. The ACL can be a unified ACL with ACEs that specify both IPv4 and IPv6 addresses.
- **none** — indicates that there is no network list for split tunneling; the ASA tunnels all traffic. Specifying the **none** keyword sets a split tunneling network list with a null value, thereby disallowing split tunneling. It also prevents inheriting a default split tunneling network list from a default or specified group policy.

To delete a network list, enter the **no** form of this command. To delete all split tunneling network lists, enter the **no split-tunnel-network-list** command without arguments. This command deletes all configured network lists, including a null list if you created one by entering the **none** keyword.

When there are no split tunneling network lists, users inherit any network lists that exist in the default or specified group policy. To prevent users from inheriting such network lists, enter the **split-tunnel-network-list none** command.

**Example**

The following example shows how to create a network list named FirstList, and add it to the group policy named FirstGroup. FistList is an exclusion list and an inclusion list that is a subnet of the exclusion list:

```
hostname(config)# split-tunnel-policy tunnelspecified
hostname(config)# access-list FirstList deny ip 10.10.10.0 255.255.255.0 any
hostname(config)# access-list FirstList permit ip 10.0.0.0 255.0.0.0 any

hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-tunnel-network-list value FirstList
```

The following example shows how to create a network list named v6, and add the v6 split tunnel policy to the group policy named GroupPolicy_ipv6-ikev2. v6 is an exclusion list and an inclusion list that is a subnet of the exclusion list:

```
hostname(config)# access-list v6 extended permit ip fd90:5000::/32 any6
hostname(config)# access-list v6 extended deny ip fd90:5000:3000:2880::/64 any6

hostname(config)# group-policy GroupPolicy_ipv6-ikev2 internal
hostname(config)# group-policy GroupPolicy_ipv6-ikev2 attributes
hostname(config-group-policy)# vpn-tunnel-protocol ikev2 ssl-client
hostname(config-group-policy)# ipv6-split-tunnel-policy tunnelspecified
hostname(config-group-policy)# split-tunnel-network-list value v6
```

### Verify the Split Tunnel Configuration

Run the **show runn group-policy attributes** command to verify your configuration. This example shows that the administrator has set both an IPv4 and IPv6 network policy and used the network list (unified ACL), **FirstList** for both policies.

```
hostname(config-group-policy)# show runn group-policy FirstGroup attributes
group-policy FirstGroup attributes
 split-tunnel-policy tunnelspecified
 ipv6-split-tunnel-policy tunnelspecified
 split-tunnel-network-list value FirstList
```

# Configure Domain Attributes for Split Tunneling

You can specify a default domain name or a list of domains to be resolved through the split tunnel, which we refer to as split DNS.

AnyConnect 3.1 supports true split DNS functionality for Windows and Mac OS X platforms. If the group policy on the security appliance enables split-include tunneling, and if it specifies the DNS names to be tunneled, AnyConnect tunnels any DNS queries that match those names to the private DNS server. True split DNS allows tunnel access to only DNS requests that match the domains pushed to the client by the ASA. These requests are not sent in the clear. On the other hand, if the DNS requests do not match the domains pushed down by the ASA, AnyConnect lets the DNS resolver on the client operating system submit the host name in the clear for DNS resolution.



**Note** Split DNS supports standard and update queries (including A, AAAA, NS, TXT, MX, SOA, ANY, SRV, PTR, and CNAME). PTR queries matching any of the tunneled networks are allowed through the tunnel.

For Mac OS X, AnyConnect can use true split-DNS for a certain IP protocol only if one of the following conditions is met:

- Split-DNS is configured for one IP protocol (such as IPv4), and Client Bypass Protocol is configured for the other IP protocol (such as IPv6) in the group policy (with no address pool configured for the latter IP protocol).
- Split-DNS is configured for both IP protocols.

### Define a Default Domain Name

The ASA passes the default domain name to the Secure Client. The client appends the domain name to DNS queries that omit the domain field. This domain name applies only to tunneled packets.When there are no default domain names, users inherit the default domain name in the default group policy.

To specify the default domain name for users of the group policy, enter the **default-domain** command in group-policy configuration mode. To delete a domain name, enter the **no** form of this command.

```
hostname(config-group-policy)# default-domain {value domain-name | none}
hostname(config-group-policy)# no default-domain [domain-name]
```

The **value** domain-name parameter identifies the default domain name for the group. To specify that there is no default domain name, enter the **none** keyword. This command sets a default domain name with a null value, which disallows a default domain name and prevents inheriting a default domain name from a default or specified group policy.

To delete all default domain names, enter the **no default-domain** command without arguments. This command deletes all configured default domain names, including a null list if you created one by entering the **default-domain** command with the **none** keyword. The **no** form allows inheriting a domain name.

The following example shows how to set a default domain name of FirstDomain for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# default-domain value FirstDomain
```

### Define a List of Domains for Split Tunneling

Enter a list of domains to be resolved through the split tunnel, in addition to the default domain. Enter the **split-dns** command in group-policy configuration mode. To delete a list, enter the **no** form of this command.

When there are no split tunneling domain lists, users inherit any that exist in the default group policy. To prevent users from inheriting such split tunneling domain lists, enter the **split-dns** command with the **none** keyword.

To delete all split tunneling domain lists, enter the **no split-dns** command without arguments. This deletes all configured split tunneling domain lists, including a null list created by issuing the **split-dns** command with the **none** keyword.

The parameter **value** domain-name provides a domain name that the ASA resolves through the split tunnel. The **none** keyword indicates that there is no split DNS list. It also sets a split DNS list with a null value, thereby disallowing a split DNS list, and prevents inheriting a split DNS list from a default or specified group policy. The syntax of the command is as follows:

```
hostname(config-group-policy)# split-dns {value domain-name1 [domain-name2... domain-nameN]
 | none}
hostname(config-group-policy)# no split-dns [domain-name domain-name2 domain-nameN]
```

Enter a single space to separate each entry in the list of domains. There is no limit on the number of entries, but the entire string can be no longer than 492 characters. You can use only alphanumeric characters, hyphens (-), and periods (.). If the default domain name is to be resolved through the tunnel, you must explicitly include that name in this list.

The following example shows how to configure the domains Domain1, Domain2, Domain3, and Domain4 to be resolved through split tunneling for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# split-dns value Domain1 Domain2 Domain3 Domain4
```



**Note** When configuring split DNS, ensure the private DNS servers specified do not overlap with the DNS servers configured for the client platform. If they do, name resolution does not function properly and queries may be dropped.

## Configure DHCP Intercept for Windows XP and Split Tunneling

A Microsoft XP anomaly results in the corruption of domain names if split tunnel options exceed 255 bytes. To avoid this problem, the ASA limits the number of routes it sends to 27 to 40 routes, with the number of routes dependent on the classes of the routes.

DHCP Intercept lets Microsoft Windows XP clients use split-tunneling with the ASA. The ASA replies directly to the Microsoft Windows XP client DHCP Inform message, providing that client with the subnet mask, domain name, and classless static routes for the tunnel IP address. For Windows clients prior to Windows XP, DHCP Intercept provides the domain name and subnet mask. This is useful in environments in which using a DHCP server is not advantageous.

The **intercept-dhcp** command enables or disables DHCP intercept.

```
hostname(config-group-policy)# intercept-dhcp netmask {enable | disable}
```

```
hostname(config-group-policy)#
```

The *netmask* variable provides the subnet mask for the tunnel IP address. The **no** form of this command removes the DHCP intercept from the configuration:

[**no**] **intercept-dhcp**

The following example shows how to set DHCP Intercepts for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# intercept-dhcp enable
```

# Configure Browser Proxy Settings for use with Remote Access Clients

Follow these steps to configure the proxy server parameters for a client.

### Procedure

**Step 1** Configure a browser proxy server and port for a client device by entering the **msie-proxy server** command in group-policy configuration mode:

```
hostname(config-group-policy)# msie-proxy server {value server[:port] | none}
hostname(config-group-policy)#
```

The default value is **none**, which is not specifying any proxy server settings on the browser of the client device. To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy server
hostname(config-group-policy)#
```

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to configure the IP address 192.168.10.1 as a browser proxy server, using port 880, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy server value 192.168.21.1:880
hostname(config-group-policy)#
```

**Step 2** Configure the browser proxy actions ("methods") for a client device by entering the **msie-proxy method** command in group-policy configuration mode.

```
hostname(config-group-policy)# msie-proxy method [auto-detect | no-modify |
no-proxy | use-server]
hostname(config-group-policy)#
```

The default value is **no-modify**. To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy method [auto-detect | no-modify |
no-proxy | use-server]
hostname(config-group-policy)#
```

The available methods are as follows:

- **auto-detect**—Enables the use of automatic proxy server detection in the browser for the client device.
- **no-modify**—Leaves the HTTP browser proxy server setting in the browser unchanged for this client device.
- **no-proxy—**Disables the HTTP proxy setting in the browser for the client device.
- **use-server—**Sets the HTTP proxy server setting in the browser to use the value configured in the **msie-proxy server** command.

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to configure auto-detect as the browser proxy setting for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy method auto-detect
hostname(config-group-policy)#
```

The following example configures the browser proxy setting for the group policy named FirstGroup to use the server QAserver, port 1001 as the server for the client device:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy server QAserver:port 1001
hostname(config-group-policy)# msie-proxy method use-server
hostname(config-group-policy)#
```

**Step 3** Configure browser proxy exception list settings for a local bypass on the client device by entering the **msie-proxy except-list** command in group-policy configuration mode. These addresses are not accessed by a proxy server. This list corresponds to the Exceptions box in the Proxy Settings dialog box.

```
hostname(config-group-policy)# msie-proxy except-list {value server[:port] | none}
hostname(config-group-policy)#
```

To remove the attribute from the configuration, use the **no** form of the command:

```
hostname(config-group-policy)# no msie-proxy except-list
hostname(config-group-policy)#
```

- **value** server:port—Specifies the IP address or name of an MSIE server and port that is applied for this client device. The port number is optional.

- **none**—Indicates that there is no IP address/hostname or port and prevents inheriting an exception list.

By default, msie-proxy except-list is disabled.

The line containing the proxy server IP address or hostname and the port number must be less than 100 characters long.

The following example shows how to set a browser proxy exception list, consisting of the server at IP address 192.168.20.1, using port 880, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy except-list value 192.168.20.1:880
hostname(config-group-policy)#
```

**Step 4** Enable or disable browser proxy local-bypass settings for a client device by entering the **msie-proxy local-bypass** command in group-policy configuration mode.

```
hostname(config-group-policy)# msie-proxy local-bypass {enable | disable}
hostname(config-group-policy)#
```

To remove the attribute from the configuration, use the **no** form of the command.

```
hostname(config-group-policy)# no msie-proxy local-bypass {enable | disable}
hostname(config-group-policy)#
```

By default, msie-proxy local-bypass is disabled.

The following example shows how to enable browser proxy local-bypass for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# msie-proxy local-bypass enable
hostname(config-group-policy)#
```

# Configure Security Attributes for IPsec (IKEv1) Clients

To specify the security settings for a group, perform these steps.

### Procedure

**Step 1** Specify whether to let users store their login passwords on the client system, using the **password-storage** command with the **enable** keyword in group-policy configuration mode. To disable password storage, use the **password-storage** command with the **disable** keyword.

```
hostname(config-group-policy)# password-storage {enable | disable}
hostname(config-group-policy)#
```

For security reasons, password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites.

To remove the password-storage attribute from the running configuration, enter the **no** form of this command:

```
hostname(config-group-policy)# no password-storage
hostname(config-group-policy)#
```

Specifying the **no** form enables inheritance of a value for password-storage from another group policy.

This command does not apply to interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# password-storage enable
hostname(config-group-policy)#
```

**Step 2** Specify whether to enable IP compression, which is disabled by default.

**Note**
IP compression is not supported for IPsec IKEv2 connections.

```
hostname(config-group-policy)# ip-comp {enable | disable}
hostname(config-group-policy)#
```

To enable LZS IP compression, enter the **ip-comp** command with the **enable** keyword in group-policy configuration mode. To disable IP compression, enter the **ip-comp** command with the **disable** keyword.

To remove the **ip-comp** attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value from another group policy.

```
hostname(config-group-policy)# no ip-comp
hostname(config-group-policy)#
```

Enabling data compression might speed up data transmission rates for remote dial-in users connecting with modems.

**Tip**
Data compression increases the memory requirement and CPU usage for each user session and consequently decreases the overall throughput of the ASA. For this reason, we recommend that you enable data compression only for remote users connecting with a modem. Design a group policy specific to modem users, and enable compression only for them.

**Step 3** Specify whether to require that users reauthenticate on IKE re-key by using the **re-xauth** command with the **enable** keyword in group-policy configuration mode.

**Note**
IKE re-key is not supported for IKEv2 connections.

If you enable reauthentication on IKE re-key, the ASA prompts the user to enter a username and password during initial Phase 1 IKE negotiation and also prompts for user authentication whenever an IKE re-key occurs. Reauthentication provides additional security.

If the configured re-key interval is very short, users might find the repeated authorization requests inconvenient. To avoid repeated authorization requests, disable reauthentication. To check the configured re-key interval, in monitoring mode, enter the **show crypto ipsec sa** command to view the security association lifetime in seconds and lifetime in kilobytes of data. To disable user reauthentication on IKE re-key, enter the **disable** keyword. Reauthentication on IKE re-key is disabled by default.

```
hostname(config-group-policy)# re-xauth {enable | disable}
hostname(config-group-policy)#
```

To enable inheritance of a value for reauthentication on IKE re-key from another group policy, remove the re-xauth attribute from the running configuration by entering the **no** form of this command:

```
hostname(config-group-policy)# no re-xauth
hostname(config-group-policy)#
```

**Note**
Reauthentication fails if there is no user at the other end of the connection.

**Step 4** Specify whether to enable perfect forward secrecy. In IPsec negotiations, perfect forward secrecy ensures that each new cryptographic key is unrelated to any previous key. A group policy can inherit a value for perfect forward secrecy from another group policy. Perfect forward secrecy is disabled by default. To enable perfect forward secrecy, use the **pfs** command with the **enable** keyword in group-policy configuration mode.

```
hostname(config-group-policy)# pfs {enable | disable}
hostname(config-group-policy)#
```

To disable perfect forward secrecy, enter the **pfs** command with the **disable** keyword.

To remove the perfect forward secrecy attribute from the running configuration and prevent inheriting a value, enter the **no** form of this command.

```
hostname(config-group-policy)# no pfs
hostname(config-group-policy)#
```

# Configure IPsec-UDP Attributes for IKEv1 Clients

IPsec over UDP, sometimes called IPsec through NAT, lets a hardware client connect via UDP to a ASA that is running NAT. It is disabled by default. IPsec over UDP is proprietary; it applies only to remote-access connections, and it requires mode configuration. The ASA exchanges configuration parameters with the client while negotiating SAs. Using IPsec over UDP may slightly degrade system performance.

To enable IPsec over UDP, configure the **ipsec-udp** command with the **enable** keyword in group-policy configuration mode, as follows:

```
hostname(config-group-policy)# ipsec-udp {enable | disable}
hostname(config-group-policy)# no ipsec-udp
```

To use IPsec over UDP, you must also configure the **ipsec-udp-port** command, as described in this section.

To disable IPsec over UDP, enter the **disable** keyword. To remove the IPsec over UDP attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for IPsec over UDP from another group policy.

The following example shows how to set IPsec over UDP for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp enable
```

If you enabled IPsec over UDP, you must also configure the **ipsec-udp-port** command in group-policy configuration mode. This command sets a UDP port number for IPsec over UDP. In IPsec negotiations, the ASA listens on the configured port and forwards UDP traffic for that port even if other filter rules drop UDP traffic. The port numbers can range from 4001 through 49151. The default port value is 10000.

To disable the UDP port, enter the **no** form of this command. This enables inheritance of a value for the IPsec over UDP port from another group policy.

```
hostname(config-group-policy)# ipsec-udp-port port
```

The following example shows how to set an IPsec UDP port to port 4025 for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# ipsec-udp-port 4025
```

# Configure Attributes for VPN Hardware Clients

### Procedure

**Step 1** (Optional) Configure Network Extension Mode with the following command:

[**no**] **nem** [**enable** | **disable**]

Network extension mode lets hardware clients present a single, routable network to the remote private network over the VPN tunnel. PAT does not apply. Therefore, devices behind the Easy VPN Server have direct access to devices on the private network behind the Easy VPN Remote over the tunnel, and only over the tunnel, and vice versa. The hardware client must initiate the tunnel, but after the tunnel is up, either side can initiate data exchange.

**Example:**

The following example shows how to set NEM for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# nem enable
```

To disable NEM, enter the **disable** keyword. To remove the NEM attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from another group policy.

**Step 2** (Optional) Configure Secure Unit Authentication with the following command:

[**no**] **secure-unit-authentication** [**enable** | **disable** ]

Secure unit authentication provides additional security by requiring VPN hardware clients to authenticate with a username and password each time that the client initiates a tunnel. With this feature enabled, the hardware client does not a use the saved username and password if configured. Secure unit authentication is disabled by default.

Secure unit authentication requires that you have an authentication server group configured for the connection profile the hardware client(s) uses. If you require secure unit authentication on the primary ASA, be sure to configure it on any backup servers as well.

**Note**
With this feature enabled, to bring up a VPN tunnel, a user must be present to enter the username and password.

**Example:**

The following example shows how to enable secure unit authentication for the group policy named FirstGroup:

```
hostname(config)#group-policy FirstGroup attributes
hostname(config-group-policy)# secure-unit-authentication enable
```

To disable secure unit authentication, enter the **disable** keyword. To remove the secure unit authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for secure unit authentication from another group policy.

**Step 3** (Optional) Configure User Authentication with the following command:

[**no**] **user-authentication** [**enable** | **disable**]

When enabled, user authentication requires that individual users behind a hardware client authenticate to gain access to the network across the tunnel. Individual users authenticate according to the order of authentication servers that you configure. User authentication is disabled by default.

If you require user authentication on the primary ASA, be sure to configure it on any backup servers as well.

**Example:**

The following example shows how to enable user authentication for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication enable
```

To disable user authentication, enter the **disable** keyword. To remove the user authentication attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for user authentication from another group policy.

**Step 4** Set an idle timeout for individual users that have authenticated with the following command:

[**no**] **user-authentication-idle-timeout** *minutes* | **none** ]

The *minutes* parameter specifies the number of minutes in the idle timeout period. The minimum is 1 minute, the default is 30 minutes, and the maximum is 35791394 minutes.

If there is no communication activity by a user behind a hardware client in the idle timeout period, the ASA terminates the client's access. This timer terminates only the client's access through the VPN tunnel, not the VPN tunnel itself.

**Example:**

The following example shows how to set an idle timeout value of 45 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication enable
hostname(config-group-policy)#user-authentication-idle-timeout 45
```

To delete the idle timeout value, enter the **no** form of this command. This option allows inheritance of an idle timeout value from another group policy. To prevent inheriting an idle timeout value, enter the **user-authentication-idle-timeout** command with the **none** keyword. This command sets the idle timeout with a null value, which disallows an idle timeout and prevents inheriting a user authentication idle timeout value from a default or specified group policy.

**Note**
The idle timeout indicated in response to the **show uauth** command is always the idle timeout value of the user who authenticated the tunnel on the Cisco Easy VPN remote device.

**Step 5** Configure IP Phone Bypass with the followng command:

**ip-phone-bypass enable**

IP Phone Bypass lets IP phones behind hardware clients connect without undergoing user authentication processes. IP Phone Bypass is disabled by default. This only applies when IUA is enabled.

**Note**
You must also configure MAC address exemption on the client to exempt these clients from authentication.

To disable IP Phone Bypass, enter the **disable** keyword. To remove the IP phone Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for IP Phone Bypass from another group policy.

**Step 6** Configure LEAP Bypass with the following command:

**leap-bypass enable**

LEAP Bypass only applies when **user-authentication** is enabled. This command lets LEAP packets from Cisco wireless access point devices establish LEAP authentication and then authenticate again per user authentication. LEAP Bypass is disabled by default.

LEAP users behind a hardware client have a circular dilemma: they cannot negotiate LEAP authentication because they cannot send their credentials to the RADIUS server behind the central site device over the tunnel. The reason they cannot send their credentials over the tunnel is that they have not authenticated on the wireless network. To solve this problem, LEAP Bypass lets LEAP packets, and only LEAP packets, traverse the tunnel to authenticate the wireless connection to a RADIUS server before individual users authenticate. Then the users proceed with individual user authentication.

LEAP Bypass operates correctly under the following conditions:

- **secure-unit-authentication** must be disabled. If interactive unit authentication is enabled, a non-LEAP (wired) device must authenticate the hardware client before LEAP devices can connect using that tunnel.
- **user-authentication** is enabled. Otherwise, LEAP Bypass does not apply.

- Access points in the wireless environment must be Cisco Aironet Access Points running Cisco Discovery Protocol (CDP). The wireless NIC cards for PCs can be other brands.

**Example:**

The following example shows how to set LEAP Bypass for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# user-authentication enable
hostname(config-group-policy)# leap-bypass enable
```

To disable LEAP Bypass, enter the **disable** keyword. To remove the LEAP Bypass attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value for LEAP Bypass from another group policy:

# Configure Group Policy Attributes for Secure Client Connections

After enabling Secure Client connections as described in AnyConnect VPN Client Connections, on page 235, you can enable or require Secure Client features for a group policy. Follow these steps in group-policy webvpn configuration mode:

**Procedure**

**Step 1** Enter group policy webvpn configuration mode. For example:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
```

**Step 2** To disable the permanent installation of the Secure Client on the endpoint computer, use the anyconnect keep-installer command with the **none** keyword. For example:

```
hostname(config-group-webvpn)# anyconnect keep-installer none
hostname(config-group-webvpn)#
```

The default is that permanent installation of the client is enabled. The client remains installed on the endpoint at the end of the Secure Client session.

**Step 3** To enable compression of HTTP data over the Secure Client SSL connection for the group policy, enter the anyconnect ssl compression command. By default, compression is set to **none** (disabled). To enable compression, use the **deflate** keyword. For example:

```
hostname(config-group-webvpn)# anyconnect compression deflate
hostname(config-group-webvpn)#
```

**Step 4** Configure Dead Peer Detection, on page 250

**Step 5** You can ensure that the Secure Client connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle by adjusting the frequency of keepalive messages using the **anyconnect ssl keepalive command:**

**anyconnect ssl keepalive {none** | *seconds***}**

Adjusting keepalives also ensures the Secure Client does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer.

The following example configures the security appliance to enable the Secure Client to send keepalive messages, with a frequency of 300 seconds (5 minutes):

```
hostname(config-group-webvpn)# anyconnect ssl keepalive 300
hostname(config-group-webvpn)#
```

**Step 6** To enable the Secure Client to perform a re-key on an SSL session, use the anyconnect ssl rekey command:

anyconnect ssl rekey {**method** {**ssl** | **new-tunnel**} | **time** *minutes* | **none**}}

By default, re-key is disabled.

Specifying the method as new-tunnel specifies that the Secure Client establishes a new tunnel during SSL re-key. Specifying the method as none disables re-key. Specifying the method as ssl specifies that SSL renegotiation takes place during re-key. Instead of specifying the method, you can specify the time; that is, the number of minutes from the start of the session until the re-key takes place, from 1 through 10080 (1 week).

The following example configures the Secure Client to renegotiate with SSL during re-key and configures the re-key to occur 30 minutes after the session begins:

```
hostname(config-group-webvpn)# anyconnect ssl rekey method ssl
hostname(config-group-webvpn)# anyconnect ssl rekey time 30
hostname(config-group-webvpn)#
```

**Step 7** The Client Protocol Bypass feature allows you to configure how the Secure Client manages IPv4 traffic when ASA is expecting only IPv6 traffic or how it manages IPv6 traffic when it is expecting only IPv4 traffic.

When the Secure Client makes a VPN connection to the ASA, the ASA could assign it an IPv4, IPv6, or both an IPv4 and IPv6 address. If the ASA assigns the Secure Client connection only an IPv4 address or only an IPv6 address, you can now configure the Client Bypass Protocol to drop network traffic for which the ASA did not assign an IP address, or allow that traffic to bypass the ASA and be sent from the client unencrypted or “in the clear.”

For example, assume that the ASA assigns only an IPv4 address to the Secure Client connection and the endpoint is dual stacked. When the endpoint attempts to reach an IPv6 address, if Client Bypass Protocol is disabled, the IPv6 traffic is dropped; however, if Client Bypass Protocol is enabled, the IPv6 traffic is sent from the client in the clear.

If establishing an IPsec tunnel (as opposed to an SSL connection), the ASA is not notified whether or not IPv6 is enabled on the client, so ASA always pushes down the client bypass protocol setting.

Use the client-bypass-protocol command to enable or disable the client bypass protocol feature. This is the command syntax:

**client-bypass-protocol {enable | disable}**

The following example enables client bypass protocol:

```
hostname(config-group-policy)# client-bypass-protocol enable
hostname(config-group-policy)#
```

The following example disables client bypass protocol:

```
hostname(config-group-policy)# client-bypass-protocol disable
hostname(config-group-policy)#
```

The following example removes an enabled or disabled client bypass protocol setting:

```
hostname(config-group-policy)# no client-bypass-protocol enable
hostname(config-group-policy)#
```

**Step 8** If you have configured Load Balancing between your ASAs, specify the FQDN of the ASA in order to resolve the ASA IP address used for re-establishing the VPN session. This setting is critical to support client roaming between networks of different IP protocols (such as IPv4 to IPv6).

You cannot use the ASA FQDN present in the Secure Client profile to derive the ASA IP address after roaming. The addresses may not match the correct device (the one the tunnel was established to) in the load balancing scenario.

If the device FQDN is not pushed to the client, the client will try to reconnect to whatever IP address the tunnel had previously established. In order to support roaming between networks of different IP protocols (from IPv4 to IPv6), Secure Client must perform name resolution of the device FQDN after roaming, so that it can determine which ASA address to use for re-establishing the tunnel. The client uses the ASA FQDN present in its profile during the initial connection. During subsequent session reconnects, it always uses the device FQDN pushed by ASA (and configured by the administrator in the group policy), when available. If the FQDN is not configured, the ASA derives the device FQDN (and sends it to the client) from whatever is set under Device Setup > Device Name/Password and Domain Name.

If the device FQDN is not pushed by the ASA, the client cannot re-establish the VPN session after roaming between networks of different IP protocols.

Use the gateway-fqdn command to configure the FQDN of the ASA. This is the command syntax:

**gateway-fqdn** { **value** *FQDN_Name* | **none**} or **no gateway-fqdn**

The following example defines the FQDN of the ASA as ASAName.example.cisco.com

```
hostname(config-group-policy)# gateway-fqdn value ASAName.example.cisco.com
hostname(config-group-policy)#
```

The following example removes the FQDN of the ASA from the group policy. The group policy then inherits this value from the Default Group Policy.

```
hostname(config-group-policy)# no gateway-fqdn
hostname(config-group-policy)#
```

The following example defines the FQDN as an empty value. The global FQDN configured using hostname and domain-name commands will be used if available.

```
hostname(config-group-policy)# gateway-fqdn none
```

```
hostname(config-group-policy)#
```

## Configure Backup Server Attributes

Configure backup servers if you plan on using them. IPsec backup servers let a VPN client connect to the central site when the primary ASA is unavailable.When you configure backup servers, the ASA pushes the server list to the client as the IPsec tunnel is established. Backup servers do not exist until you configure them, either on the client or on the primary ASA.

Configure backup servers either on the client or on the primary ASA. If you configure backup servers on the ASA, it pushes the backup server policy to the clients in the group, replacing the backup server list on the client if one is configured.



**Note** If you are using hostnames, it is wise to have backup DNS and WINS servers on a separate network from that of the primary DNS and WINS servers. Otherwise, if clients behind a hardware client obtain DNS and WINS information from the hardware client via DHCP, and the connection to the primary server is lost, and the backup servers have different DNS and WINS information, clients cannot be updated until the DHCP lease expires. In addition, if you use hostnames and the DNS server is unavailable, significant delays can occur.

To configure backup servers, enter the **backup-servers** command in group-policy configuration mode:

```
hostname(config-group-policy)# backup-servers {server1 server2... server10 |
clear-client-config | keep-client-config}
```

To remove a backup server, enter the **no** form of this command with the backup server specified. To remove the backup-servers attribute from the running configuration and enable inheritance of a value for backup-servers from another group policy, enter the **no** form of this command without arguments.

```
hostname(config-group-policy)# no backup-servers [server1 server2... server10 |
clear-client-config | keep-client-config]
```

The **clear-client-config** keyword specifies that the client uses no backup servers. The ASA pushes a null server list.

The **keep-client-config** keyword specifies that the ASA sends no backup server information to the client. The client uses its own backup server list, if configured. This is the default.

The *server1 server 2.... server10* parameter list is a space-delimited, priority-ordered list of servers for the VPN client to use when the primary ASA is unavailable. This list identifies servers by IP address or hostname. The list can be 500 characters long, and it can contain up to10 entries.

The following example shows how to configure backup servers with IP addresses 10.10.10.1 and 192.168.10.14, for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# backup-servers 10.10.10.1 192.168.10.14
```

# Configure Network Admission Control Parameters

The group-policy NAC commands in this section all have default values. Unless you have a good reason for changing them, accept the default values for these parameters.

The ASA uses Extensible Authentication Protocol (EAP) over UDP (EAPoUDP) messaging to validate the posture of remote hosts. Posture validation involves the checking of a remote host for compliancy with safety requirements before the assignment of a network access policy. An Access Control Server must be configured for Network Admission Control before you configure NAC on the security appliance.

The Access Control Server downloads the posture token, an informational text string configurable on the ACS, to the security appliance to aid in system monitoring, reporting, debugging, and logging. A typical posture token is Healthy, Checkup, Quarantine, Infected, or Unknown. Following posture validation or clientless authentication, the ACS downloads the access policy for the session to the security appliance.

To configure Network Admission Control settings for the default group policy or an alternative group policy, perform the following steps.

### Procedure

**Step 1** (Optional) Configure the status query timer period. The security appliance starts the status query timer after each successful posture validation and status query response. The expiration of this timer triggers a query for changes in the host posture, referred to as a status query. Enter the number of seconds in the range 30 through 1800. The default setting is 300.

To specify the interval between each successful posture validation in a Network Admission Control session and the next query for changes in the host posture, use the **nac-sq-period** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-sq-period seconds
hostname(config-group-policy)#
```

To inherit the value of the status query timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-sq-period [seconds]
hostname(config-group-policy
```

The following example changes the value of the status query timer to 1800 seconds:

```
hostname(config-group-policy)# nac-sq-period 1800
hostname(config-group-policy)#
```

The following example inherits the value of the status query timer from the default group policy:

```
hostname(config-group-policy)# no nac-sq-period
hostname(config-group-policy)#
```

**Step 2** (Optional) Configure the NAC revalidation period. The security appliance starts the revalidation timer after each successful posture validation. The expiration of this timer triggers the next unconditional posture validation. The security appliance maintains posture validation during revalidation. The default group policy becomes

effective if the Access Control Server is unavailable during posture validation or revalidation. Enter the interval in seconds between each successful posture validation. The range is 300 through 86400. The default setting is 36000.

To specify the interval between each successful posture validation in a Network Admission Control session, use the **nac-reval-period** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-reval-period seconds
hostname(config-group-policy)#
```

To inherit the value of the Revalidation Timer from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-reval-period [seconds]
hostname(config-group-policy)#
```

The following example changes the revalidation timer to 86400 seconds:

```
hostname(config-group-policy)# nac-reval-period 86400
hostname(config-group-policy)
```

The following example inherits the value of the revalidation timer from the default group policy:

```
hostname(config-group-policy)# no nac-reval-period
hostname(config-group-policy)#
```

**Step 3** (Optional) Configure the default ACL for NAC. The security appliance applies the security policy associated with the selected ACL if posture validation fails. Specify **none** or an extended ACL. The default setting is **none**. If the setting is **none** and posture validation fails, the security appliance applies the default group policy.

To specify the ACL to be used as the default ACL for Network Admission Control sessions that fail posture validation, use the **nac-default-acl** command in group-policy configuration mode:

```
hostname(config-group-policy)# nac-default-acl {acl-name | none}
hostname(config-group-policy)#
```

To inherit the ACL from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac-default-acl [acl-name | none]
hostname(config-group-policy)#
```

The elements of this command are as follows:

- *acl-name*—Specifies the name of the posture validation server group, as configured on the ASA using the **aaa-server host** command. The name must match the server-tag variable specified in that command.
- **none**—Disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation.

Because NAC is disabled by default, VPN traffic traversing the ASA is not subject to the NAC Default ACL until NAC is enabled.

The following example identifies acl-1 as the ACL to be applied when posture validation fails:

```
hostname(config-group-policy)# nac-default-acl acl-1
hostname(config-group-policy)#
```

The following example inherits the ACL from the default group policy:

```
hostname(config-group-policy)# no nac-default-acl
hostname(config-group-policy)#
```

The following example disables inheritance of the ACL from the default group policy and does not apply an ACL to NAC sessions that fail posture validation:

```
hostname(config-group-policy)# nac-default-acl none
hostname(config-group-policy)#
```

**Step 4** Configure NAC exemptions for VPN. By default, the exemption list is empty.The default value of the filter attribute is **none**. Enter the **vpn-nac-exempt command** once for each operating system (and ACL) to be matched to exempt remote hosts from posture validation.

To add an entry to the list of remote computer types that are exempt from posture validation, use the **vpn-nac-exempt** command in group-policy configuration mode:

```
hostname(config-group-policy)# vpn-nac-exempt os "os name" [filter {acl-name | none}]
[disable]
hostname(config-group-policy)#
```

To disable inheritance and specify that all hosts are subject to posture validation, use the **none** keyword immediately following **vpn-nac-exempt**:

```
hostname(config-group-policy)# vpn-nac-exempt none
hostname(config-group-policy)#
```

To remove an entry from the exemption list, use the **no** form of this command and name the operating system (and ACL) in the entry to be removed:

```
hostname(config-group-policy)# no vpn-nac-exempt [os "os name"] [filter {acl-name | none}]
 [disable]
hostname(config-group-policy)#
```

To remove all entries from the exemption list associated with this group policy and inherit the list from the default group policy, use the **no** form of this command without specifying additional keywords:

```
hostname(config-group-policy)# no vpn-nac-exempt
hostname(config-group-policy)#
```

The syntax elements for these commands are as follows:

- *acl-name*—Name of the ACL present in the ASA configuration.

- disable—Disables the entry in the exemption list without removing it from the list.
- **filter**—(Optional) Apply an ACL to filter the traffic if the computer matches the OS name.
- **none**—When entered immediately after **vpn-nac-exempt**, this keyword disables inheritance and specifies that all hosts are subject to posture validation.When entered immediately after **filter**, this keyword indicates that the entry does not specify an ACL.
- **OS—**Exempts an operating system from posture validation.
- *os name*—Operating system name. Quotation marks are required only if the name includes a space (for example, “Windows XP”).

The following example disables inheritance and specifies that all hosts will be subject to posture validation:

```
hostname(config-group-policy)# no vpn-nac-exempt none
hostname(config-group-policy)
```

The following example removes all entries from the exemption list:

```
hostname(config-group-policy)# no vpn-nac-exempt
hostname(config-group-policy)
```

**Step 5** Enable or disable Network Admission Control by entering the following command:

```
hostname(config-group-policy)# nac {enable | disable}
hostname(config-group-policy)#
```

To inherit the NAC setting from the default group policy, access the alternative group policy from which to inherit it, then use the **no** form of this command:

```
hostname(config-group-policy)# no nac [enable | disable]
hostname(config-group-policy)#
```

By default, NAC is disabled. Enabling NAC requires posture validation for remote access. If the remote computer passes the validation checks, the ACS server downloads the access policy for the ASA to enforce. NAC is disabled by default.

An Access Control Server must be present on the network.

The following example enables NAC for the group policy:

```
hostname(config-group-policy)# nac enable
hostname(config-group-policy)#
```

## Configure VPN Client Firewall Policies

A firewall isolates and protects a computer from the Internet by inspecting each inbound and outbound packet of data to determine whether to allow it through the firewall or to drop it. Firewalls provide extra security if remote users in a group have split tunneling configured. In this case, the firewall protects the user’s computer,

and thereby the corporate network, from intrusions by way of the Internet or the user's local LAN. Remote users connecting to the ASA with the VPN client can choose the appropriate firewall option.

Set personal firewall policies that the ASA pushes to the VPN client during IKE tunnel negotiation by using the **client-firewall** command in group-policy configuration mode. To delete a firewall policy, enter the **no** form of this command.

To delete all firewall policies, enter the **no client-firewall** command without arguments. This command deletes all configured firewall policies, including a null policy if you created one by entering the **client-firewall** command with the **none** keyword.

When there are no firewall policies, users inherit any that exist in the default or other group policy. To prevent users from inheriting such firewall policies, enter the **client-firewall** command with the **none** keyword.

The Add or Edit Group Policy dialog box on the Client Firewall tab lets you configure firewall settings for VPN clients for the group policy being added or modified.



**Note** Only VPN clients running Microsoft Windows can use these firewall features. They are currently not available to hardware clients or other (non-Windows) software clients.

In the first scenario, a remote user has a personal firewall installed on the PC. The VPN client enforces firewall policy defined on the local firewall, and it monitors that firewall to make sure it is running. If the firewall stops running, the VPN client drops the connection to the ASA. (This firewall enforcement mechanism is called Are You There (AYT), because the VPN client monitors the firewall by sending it periodic "are you there?" messages; if no reply comes, the VPN client knows the firewall is down and terminates its connection to the ASA.) The network administrator might configure these PC firewalls originally, but with this approach, each user can customize his or her own configuration.

In the second scenario, you might prefer to enforce a centralized firewall policy for personal firewalls on VPN client PCs. A common example would be to block Internet traffic to remote PCs in a group using split tunneling. This approach protects the PCs, and therefore the central site, from intrusions from the Internet while tunnels are established. This firewall scenario is called push policy or Central Protection Policy (CPP). On the ASA, you create a set of traffic management rules to enforce on the VPN client, associate those rules with a filter, and designate that filter as the firewall policy. The ASA pushes this policy down to the VPN client. The VPN client then in turn passes the policy to the local firewall, which enforces it.

## Configure Secure Client Firewall Policies

Firewall rules for the Secure Client can specify IPv4 and IPv6 addresses.

### Before you begin

You have created Unified Access Rules with IPv6 addresses specified.

### Procedure

**Step 1** Enter webvpn group policy configuration mode.

**webvpn**

**Example:**

```
hostname(config)# group-policy ac-client-group attributes
hostname(config-group-policy)# webvpn
```

**Step 2** Specify an access control rule for the private or public network rule. The private network rule is the rule applied to the VPN virtual adapter interface on the client.

**anyconnect firewall-rule client-interface {private | public} value** *[RuleName]*

```
hostname(config-group-webvpn)# anyconnect firewall-rule client-interface private value
ClientFWRule
```

**Step 3** Display the group policy attributes as well as the webvpn policy attribute for the group policy.

**show runn group-policy** [value]

**Example:**

```
hostname(config-group-webvpn)# show run group-policy FirstGroup
group-policy FirstGroup internal
group-policy FirstGroup attributes
webvpn
  anyconnect firewall-rule client-interface private value ClientFWRule
```

**Step 4** Remove the client firewall rule from the private network rule.

**no anyconnect firewall-rule client-interface private value** [*RuleName*]

**Example:**

```
hostname(config-group-webvpn)# no anyconnect firewall-rule client-interface private value
hostname(config-group-webvpn)#
```

# Use of a Zone Labs Integrity Server

This section introduces the Zone Labs Integrity server, also called the Check Point Integrity server, and presents an example procedure for configuring the ASA to support the Zone Labs Integrity server. The Integrity server is a central management station for configuring and enforcing security policies on remote PCs. If a remote PC does not conform to the security policy dictated by the Integrity server, it is not granted access to the private network protected by the Integrity server and ASA.

The VPN client software and the Integrity client software are co-resident on a remote PC. The following steps summarize the actions of the remote PC, ASA, and Integrity server in the establishment of a session between the PC and the enterprise private network:

1. The VPN client software (residing on the same remote PC as the Integrity client software) connects to the ASA and tells the ASA what type of firewall client it is.
2. After the ASA approves the client firewall type, the ASA passes Integrity server address information back to the Integrity client.
3. With the ASA acting as a proxy, the Integrity client establishes a restricted connection with the Integrity server. A restricted connection is only between the Integrity client and the Integrity server.

4. The Integrity server determines if the Integrity client is in compliance with the mandated security policies. If the Integrity client is in compliance with security policies, the Integrity server instructs the ASA to open the connection and provide the Integrity client with connection details.
5. On the remote PC, the VPN client passes connection details to the Integrity client and signals that policy enforcement should begin immediately and the Integrity client can enter the private network.
6. After the VPN connection is established, the Integrity server continues to monitor the state of the Integrity client using client heartbeat messages.



**Note** The current release of the ASA supports one Integrity server at a time, even though the user interfaces support the configuration of up to five Integrity servers. If the active Integrity server fails, configure another one on the ASA and then reestablish the VPN client session.

To configure the Integrity server, perform the following steps:

### Procedure

**Step 1** Configure an Integrity server using the IP address 10.0.0.5.

```
zonelabs-Integrity server-address {hostname1 | ip-address1}
```

**Example:**

```
hostname(config)# zonelabs-Integrity server-address 10.0.0.5
```

**Step 2** Specify port 300 (the default port is 5054).

```
zonelabs-integrity port port-number
```

**Example:**

```
hostname(config)# zonelabs-integrity port 300
```

**Step 3** Specify the inside interface for communications with the Integrity server.

```
zonelabs-integrity interface interface
```

**Example:**

```
hostname(config)# zonelabs-integrity interface inside
```

**Step 4** Ensure that the ASA waits 12 seconds for a response from either the active or standby Integrity servers before declaring the Integrity server as failed and closing the VPN client connections.

**Note**
If the connection between the ASA and the Integrity server fails, the VPN client connections remain open by default so that the enterprise VPN is not disrupted by the failure of an Integrity server. However, you may want to close the VPN connections if the Zone Labs Integrity server fails.

```
zonelabs-integrity fail-timeout timeout
```

**Example:**

```
hostname(config)# zonelabs-integrity fail-timeout 12
```

**Step 5** Configure the ASA so that connections to VPN clients close when the connection between the ASA and the Zone Labs Integrity server fails.

```
zonelabs-integrity fail-close
```

**Example:**

```
hostname(config)# zonelabs-integrity fail-close
```

**Step 6** Return the configured VPN client connection fail state to the default and ensure that the client connections remain open.

```
zonelabs-integrity fail-open
```

**Example:**

```
hostname(config)# zonelabs-integrity fail-open
```

**Step 7** Specify that the Integrity server connects to port 300 (the default is port 80) on the ASA to request the server SSL certificate.

```
zonelabs-integrity ssl-certificate-port cert-port-number
```

**Example:**

```
hostname(config)# zonelabs-integrity ssl-certificate-port 300
```

**Step 8** While the server SSL certificate is always authenticated, specify that the client SSL certificate of the Integrity server be authenticated.

```
zonelabs-integrity ssl-client-authentication {enable | disable}
```

**Example:**

```
hostname(config)# zonelabs-integrity ssl-client-authentication enable
```

# Set the Firewall Client Type to Zone Labs

**Procedure**

| | Command or Action | Purpose |
|---|---|---|
| **Step 1** | To set the firewall client type to the Zone Labs Integrity type, enter the following command:<br>**Example:**<br>`hostname(config)# client-firewall req zonelabs-integrity` | **client-firewall {opt \| req} zonelabs-integrity** |

### What to do next

For more information, see Configure VPN Client Firewall Policies, on page 167. The command arguments that specify firewall policies are not used when the firewall type is **zonelabs-integrity**, because the Integrity server determines these policies.

# Set the Client Firewall Parameters

Enter the following commands to set the appropriate client firewall parameters. You can configure only one instance of each command. For more information, see Configure VPN Client Firewall Policies, on page 167.

- Cisco Integrated Firewall

```
hostname(config-group-policy)# client-firewall {opt | req} cisco-integrated
acl-in ACL acl-out ACL
```

- Cisco Security Agent

```
hostname(config-group-policy)# client-firewall {opt | req} cisco-security-agent
```

- No Firewall

```
hostname(config-group-policy)# client-firewall none
```

- Custom Firewall

```
hostname(config-group-policy)# client-firewall {opt | req} custom vendor-id num product-id
 num policy {AYT | CPP acl-in ACL acl-out ACL} [description string]
```

- Zone Labs Firewalls

```
hostname(config-group-policy)# client-firewall {opt | req} zonelabs-integrity
```

**Note** When the firewall type is **zonelabs-integrity**, do not include arguments. The Zone Labs Integrity Server determines the policies.

```
hostname(config-group-policy)# client-firewall {opt | req} zonelabs-zonealarm
 policy {AYT | CPP acl-in ACL acl-out ACL}

hostname(config-group-policy)# client-firewall {opt | req}
zonelabs-zonealarmorpro policy {AYT | CPP acl-in ACL acl-out ACL}

client-firewall {opt | req} zonelabs-zonealarmpro policy {AYT | CPP acl-in
 ACL acl-out ACL}
```

- Sygate Personal Firewalls

```
hostname(config-group-policy)# client-firewall {opt | req} sygate-personal

hostname(config-group-policy)# client-firewall {opt | req} sygate-personal-pro
```

```
hostname(config-group-policy)# client-firewall {opt | req} sygate-security-agent
```

- Network Ice,Black Ice Firewall

```
hostname(config-group-policy)# client-firewall {opt | req} networkice-blackice
```

*Table 8: client-firewall Command Keywords and Variables*

| Parameter | Description |
|---|---|
| **acl-in** ACL | Provides the policy the client uses for inbound traffic. |
| **acl-out** ACL | Provides the policy the client uses for outbound traffic. |
| **AYT** | Specifies that the client PC firewall application controls the firewall policy. The ASA checks to make sure that the firewall is running. It asks, "Are You There?" If there is no response, the ASA tears down the tunnel. |
| **cisco-integrated** | Specifies Cisco Integrated firewall type. |
| **cisco-security-agent** | Specifies Cisco Intrusion Prevention Security Agent firewall type. |
| **CPP** | Specifies Policy Pushed as source of the VPN client firewall policy. |
| **custom** | Specifies Custom firewall type. |
| **description** string | Describes the firewall. |
| **networkice-blackice** | Specifies Network ICE Black ICE firewall type. |
| **none** | Indicates that there is no client firewall policy. Sets a firewall policy with a null value, thereby disallowing a firewall policy. Prevents inheriting a firewall policy from a default or specified group policy. |
| **opt** | Indicates an optional firewall type. |
| **product-id** | Identifies the firewall product. |
| **req** | Indicates a required firewall type. |
| **sygate-personal** | Specifies the Sygate Personal firewall type. |
| **sygate-personal-pro** | Specifies Sygate Personal Pro firewall type. |
| **sygate-security-agent** | Specifies Sygate Security Agent firewall type. |
| **vendor-id** | Identifies the firewall vendor. |
| **zonelabs-integrity** | Specifies Zone Labs Integrity Server firewall type. |
| **zonelabs-zonealarm** | Specifies Zone Labs Zone Alarm firewall type. |
| **zonelabs-zonealarmorpro policy** | Specifies Zone Labs Zone Alarm or Pro firewall type. |
| **zonelabs-zonealarmpro policy** | Specifies Zone Labs Zone Alarm Pro firewall type. |

The following example shows how to set a client firewall policy that requires Cisco Intrusion Prevention Security Agent for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-firewall req cisco-security-agent
hostname(config-group-policy)#
```

# Configure Client Access Rules

Configure rules that limit the remote access client types and versions that can connect via IPsec through the ASA by using the **client-access-rule** command in group-policy configuration mode. Construct rules according to these guidelines:

- If you do not define any rules, the ASA permits all connection types.
- When a client matches none of the rules, the ASA denies the connection. If you define a deny rule, you must also define at least one permit rule; otherwise, the ASA denies all connections.
- For both software and hardware clients, type and version must exactly match their appearance in the **show vpn-sessiondb remote** display.
- The ***** character is a wildcard, which you can enter multiple times in each rule. For example, **client-access rule 3 deny type * version 3.*** creates a priority 3 client access rule that denies all client types running versions 3.x software.
- You can construct a maximum of 25 rules per group policy.
- There is a limit of 255 characters for an entire set of rules.
- You can enter n/a for clients that do not send client type and/or version.

To delete a rule, enter the **no** form of this command. This command is equivalent to the following command:

```
hostname(config-group-policy)# client-access-rule 1 deny type "Cisco VPN Client" version
4.0
```

To delete all rules, enter the **no client-access-rule command** without arguments. This deletes all configured rules, including a null rule if you created one by issuing the **client-access-rule** command with the **none** keyword.

By default, there are no access rules. When there are no client access rules, users inherit any rules that exist in the default group policy.

To prevent users from inheriting client access rules, enter the **client-access-rule** command with the **none** keyword. The result of this command is that all client types and versions can connect.

```
hostname(config-group-policy)# client-access rule priority {permit | deny} type
type version {version | none}

hostname(config-group-policy)# no client-access rule [priority {permit | deny} type
 type version version]
```

The table below explains the meaning of the keywords and parameters in these commands.

*Table 9: client-access rule Command Keywords and Variables*

| Parameter | Description |
|---|---|
| **deny** | Denies connections for devices of a particular type and/or version. |
| **none** | Allows no client access rules. Sets client-access-rule to a null value, thereby allowing no restriction. Prevents inheriting a value from a default or specified group policy. |
| **permit** | Permits connections for devices of a particular type and/or version. |
| *priority* | Determines the priority of the rule. The rule with the lowest integer has the highest priority. Therefore, the rule with the lowest integer that matches a client type and/or version is the rule that applies. If a lower priority rule contradicts, the ASA ignores it. |
| **type** *type* | Identifies device types via free-form strings. The string must match exactly its appearance in the **show vpn-sessiondb remote** display, except that you can enter the * character as a wildcard. |
| **version** version | Identifies the device version via free-form strings, for example 7.0. A string must match exactly its appearance in the **show vpn-sessiondb remote** display, except that you can enter the * character as a wildcard. |

The following example shows how to create client access rules for the group policy named FirstGroup. These rules permit Cisco VPN clients running software version 4.x, while denying all Windows NT clients:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# client-access-rule 1 deny type WinNT version *
hostname(config-group-policy)# client-access-rule 2 permit "Cisco VPN Client"
version 4.*
```

**Note** The “type” field is a free-form string that allows any value, but that value must match the fixed value that the client sends to the ASA at connect time.

# Configure User Attributes

This section describes user attributes and how to configure them.

By default, users inherit all user attributes from the assigned group policy. The ASA also lets you assign individual attributes at the user level, overriding values in the group policy that applies to that user. For example, you can specify a group policy giving all users access during business hours, but give a specific user 24-hour access.

# View the Username Configuration

To display the configuration for all usernames, including default values inherited from the group policy, enter the **all** keyword with the **show running-config username** command, as follows:

```
hostname# show running-config all username
hostname#
```

This displays the encrypted password and the privilege level, for all users, or, if you supply a username, for that specific user. If you omit the **all** keyword, only explicitly configured values appear in this list. The following example displays the output of this command for the user named testuser:

```
hostname# show running-config all username testuse
username testuser password 12RsxXQnphyr/I9Z encrypted privilege 15
```

# Configure Attributes for Individual Users

To configure specific users, you assign a password (or no password) and attributes to a user using the **username** command, which enters username mode. Any attributes that you do not specify are inherited from the group policy.

The internal user authentication database consists of the users entered with the **username** command. The **login** command uses this database for authentication. To add a user to the ASA database, enter the **username** command in global configuration mode. To remove a user, use the **no** version of this command with the username you want to remove. To remove all usernames, use the **clear configure username** command without appending a username.

## Set a User Password and Privilege Level

Enter the **username** command to assign a password and a privilege level for a user. You can enter the **nopassword** keyword to specify that this user does not require a password. If you do specify a password, you can specify whether that password is stored in an encrypted form.

The optional **privilege** keyword lets you set a privilege level for this user. Privilege levels range from 0 (the lowest) through 15. System administrators generally have the highest privilege level. The default level is 2.

```
hostname(config)# username name {nopassword | password password [encrypted]}
[privilege priv_level]}
```

```
hostname(config)# no username [name]
```

The table below describes the meaning of the keywords and variables used in this command.

username Command Keywords and Variables

| Keyword/Variable | Meaning |
|---|---|
| **encrypted** | Indicates that the password is encrypted. |
| *name* | Provides the name of the user. |
| **nopassword** | Indicates that this user needs no password. |

| Keyword/Variable | Meaning |
|---|---|
| password password | Indicates that this user has a password, and provides the password. |
| privilege priv_level | Sets a privilege level for this user. The range is from 0 to 15, with lower numbers having less ability to use commands and administer the ASA. The default privilege level is 2. The typical privilege level for a system administrator is 15. |

By default, VPN users that you add with this command have no attributes or group policy association. You must explicitly configure all values.

The following example shows how to configure a user named anyuser with an encrypted password of pw_12345678 and a privilege level of 12:

```
hostname(config)# username anyuser password pw_12345678 encrypted privilege
 12
hostname(config)#
```

## Configure User Attributes

After configuring the user's password (if any) and privilege level, you set the other attributes. These can be in any order. To remove any attribute-value pair, enter the **no** form of the command.

Enter username mode by entering the **username** command with the **attributes** keyword:

```
hostname(config)# username name attributes
hostname(config-username)#
```

The prompt changes to indicate the new mode. You can now configure the attributes.

## Configure VPN User Attributes

The VPN user attributes set values specific to VPN connections, as described in the following sections.

### Configure Inheritance

You can let users inherit from the group policy the values of attributes that you have not configured at the username level. To specify the name of the group policy from which this user inherits attributes, enter the **vpn-group-policy** command. By default, VPN users have no group-policy association:

```
hostname(config-username)# vpn-group-policy group-policy-name
hostname(config-username)# no vpn-group-policy group-policy-name
```

For an attribute that is available in username mode, you can override the value of an attribute in a group policy for a particular user by configuring it in username mode.

The following example shows how to configure a user named anyuser to use attributes from the group policy named FirstGroup:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-group-policy FirstGroup
hostname(config-username)#
```

### Configure Access Hours

Associate the hours that this user is allowed to access the system by specifying the name of a configured time-range policy:

To remove the attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a time-range value from another group policy. To prevent inheriting a value, enter the **vpn-access-hours none** command. The default is unrestricted access.

```
hostname(config-username)# vpn-access-hours value {time-range | none}
hostname(config-username)# vpn-access-hours value none
hostname(config)#
```

The following example shows how to associate the user named anyuser with a time-range policy called 824:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-access-hours 824
hostname(config-username)#
```

### Configure Maximum Simultaneous Logins

Specify the maximum number of simultaneous logins allowed for this user. The range is 0 through 2147483647. The default is 3 simultaneous logins. To remove the attribute from the running configuration, enter the **no** form of this command. Enter 0 to disable login and prevent user access.

```
hostname(config-username)# vpn-simultaneous-logins integer
hostname(config-username)# no vpn-simultaneous-logins
hostname(config-username)# vpn-session-timeout alert-interval none
```



**Note** While the maximum limit for the number of simultaneous logins is very large, allowing several could compromise security and affect performance.

The following example shows how to allow a maximum of 4 simultaneous logins for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-simultaneous-logins 4
hostname(config-username)#
```

### Configure the Idle Timeout

#### Procedure

**Step 1** (Optional) To configure a VPN idle timeout period use the **vpn-idle-timeout** *minutes* command in group-policy configuration mode or in username configuration mode.

If there is no communication activity on the connection in this period, the ASA terminates the connection. The minimum time is 1 minute, the maximum time is 35791394 minutes, and the default is 30 minutes.

The following example shows how to set a VPN idle timeout of 15 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-idle-timeout 15
hostname(config-group-policy)#
```

Other actions using the [**no**] **vpn-idle-timeout** {*minutes* | **none**} command:

- Enter **vpn-idle-timeout none** to disable VPN idle timeout and pevent inheriting a timeout value.

  ```
  hostname(config)# group-policy FirstGroup attributes
  hostname(config-group-policy)# vpn-idle-timeout none
  hostname(config-group-policy)#
  ```

  This results in Secure Client (both SSL and IPsec/IKEv2) and Clientless VPN using the global webvpn **default-idle-timeout** *seconds* value. This command is entered in webvpn-config mode, for example: `hostnamee(config-webvpn)# default-idle-timeout 300`. The default is 1800 seconds (30 min), the range is 60-86400 seconds.

  For all webvon connections , the **default-idle-timeout** value is enforced only if **vpn-idle-timeout none** is set in the group policy/username attribute. A non-zero idle timeout value is required by ASA for all Secure Client connections.

  For Site-to-Site (IKEv1, IKEv2) and IKEv1 remote-access VPNs, we recommend you Disable timeout and allow for an unlimited idle period.

- To disable the idle timeout for this group policy or user policy, enter **no vpn-idle-timeout**. The value will be inherited.
- If you do not set **vpn-idle-timeout** at all, in anyway, the value is inherited, which defaults to 30 minutes.

**Note**
**vpn-idle-timeout** controls only the maximum time for a parent session. Child sessions (SSL/DTLS) are aggressively terminated much sooner by a hardcoded 5-minute TCP inactivity timeout or on failing the 3×Dead Peer Detection (DPD) check. For more details, see the note in Configure Dead Peer Detection. For further details on DPD, keepalives, and timeout attributes, see Answer AnyConnect FAQ - Tunnels, DPDs, and Inactivity Timer.

**Step 2** (Optional) You can optionally configure the time at which an idle timeout alert message is displayed to the user using the **vpn-idle-timeout alert-interval** {*minutes*} command.

This alert message tells users how many minutes they have left until their VPN session is disconnected due to inactivity. The default alert interval is one minute.

The following example shows how to set a VPN idle timeout alert interval of 3 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-idle-timeout alert-interval 3
hostname(config-username)#
```

Other actions using the [**no**] **vpn-idle-timeout alert-interval** {*minutes* | **none**} command:

- The **none** parameter indicates that users will not receive an alert.

  ```
  hostname(config)# username anyuser attributes
  hostname(config-username)# vpn-idle-timeout none
  ```

```
hostname(config-username)#
```

- To remove the alert interval for this group or user policy enter **no vpn-idle-timeout alert-interval**. The value will be inherited.
- If you do not set this parameter at all, the default alert interval is one minute.

## Configure the Maximum Connect Time

### Procedure

**Step 1** (Optional) Configure a maximum amount of time for VPN connections, using the **vpn-session-timeout** {*minutes* command in group-policy configuration mode or in username configuration mode.

The minimum time is 1 minute, and the maximum time is 35791394 minutes. There is no default value. At the end of this period of time, the ASA terminates the connection.

The following example shows how to set a VPN session timeout of 180 minutes for the group policy named FirstGroup:

```
hostname(config)# group-policy FirstGroup attributes
hostname(config-group-policy)# vpn-session-timeout 180
hostname(config-group-policy)#
```

The following example shows how to set a VPN session timeout of 180 minutes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-session-timeout 180
hostname(config-username)#
```

Other actions using the [**no**] **vpn-session-timeout** {*minutes* | **none**} command:

- To remove the attribute from this policy and allow inheritance, enter the **no vpn-session-timeout** form of this command.
- To allow an unlimited timeout period, and thus prevent inheriting a timeout value, enter **vpn-session-timeout none**.

**Step 2** Configure the time at which a session timeout alert message is displayed to the user using the **vpn-session-timeout alert-interval** {*minutes* | } command.

This alert message tells users how many minutes left until their VPN session is automatically disconnected. The following example shows how to specify that users will be notified 20 minutes before their VPN session is disconnected. You can specify a range of 1-30 minutes.

```
hostname(config-webvpn)# vpn-session-timeout alert-interval 20
```

Other actions using the [**no**] **vpn-session-timeout alert-interval** {*minutes* | **none**} command:

- Use the no form of the command to indicate that the VPN session timeout alert-interval attribute will be inherited from the Default Group Policy:

```
hostname(config-webvpn)# no vpn-session-timeout alert-interval
```

- The **vpn-session-timeout alert-interval none** indicates that users will not receive an alert.

## Apply an ACL Filter

Specify the name of a previously-configured, user-specific ACL to use as a filter for VPN connections. To disallow an ACL and prevent inheriting an ACL from the group policy, enter the **vpn-filter** command with the none keyword. To remove the ACL, including a null value created by issuing the **vpn-filter none** command, enter the **no** form of this command. The **no** option allows inheritance of a value from the group policy. There are no default behaviors or values for this command.

You configure ACLs to permit or deny various types of traffic for this user. Note that the VPN filter applies to initial connections only. It does not apply to secondary connections, such as a SIP media connection, that are opened due to the action of application inspection. You then use the **vpn-filter** command to apply those ACLs.

```
hostname(config-username)# vpn-filter {value ACL_name | none}
hostname(config-username)# no vpn-filter
hostname(config-username)#
```



**Note** Clientless SSL VPN does not use ACLs defined in the **vpn-filter** command.

The following example shows how to set a filter that invokes an ACL named acl_vpn for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-filter value acl_vpn
hostname(config-username)#
```

## Specify the IPv4 Address and Netmask

Specify the IP address and netmask to assign to a particular user. To remove the IP address, enter the **no** form of this command.

```
hostname(config-username)# vpn-framed-ip-address {ip_address}
hostname(config-username)# no vpn-framed-ip-address
hostname(config-username)
```

The following example shows how to set an IP address of 10.92.166.7 for a user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ip-address 10.92.166.7
hostname(config-username)
```

Specify the network mask to use with the IP address specified in the previous step. If you used the **no vpn-framed-ip-address** command, do not specify a network mask. To remove the subnet mask, enter the **no** form of this command. There is no default behavior or value.

```
hostname(config-username)# vpn-framed-ip-netmask {netmask}
hostname(config-username)# no vpn-framed-ip-netmask
hostname(config-username)
```

The following example shows how to set a subnet mask of 255.255.255. 254 for a user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ip-netmask 255.255.255.254
hostname(config-username)
```

### Specify the IPv6 Address and Netmask

Specify the IPv6 address and netmask to assign to a particular user. To remove the IP address, enter the **no** form of this command.

```
hostname(config-username)# vpn-framed-ipv6-address {ip_address}
hostname(config-username)# no vpn-framed-ipv6-address
hostname(config-username)
```

The following example shows how to set an IP address and netmask of 2001::3000:1000:2000:1/64 for a user named anyuser. This address indicates a prefix value of 2001:0000:0000:0000 and an interface ID of 3000:1000:2000:1.

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-framed-ipv6-address 2001::3000:1000:2000:1/64
hostname(config-username)
```

### Specify the Tunnel Protocol

Specify the VPN tunnel types (IPsec or clientless SSL VPN) that this user can use. The default is taken from the default group policy, the default for which is IPsec. To remove the attribute from the running configuration, enter the **no** form of this command.

```
hostname(config-username)# vpn-tunnel-protocol {webvpn | IPsec}
hostname(config-username)# no vpn-tunnel-protocol [webvpn | IPsec]
hostname(config-username)
```

The parameter values for this command are as follows:

- **IPsec—**Negotiates an IPsec tunnel between two peers (a remote access client or another secure gateway). Creates security associations that govern authentication, encryption, encapsulation, and key management.
- **webvpn—**Provides clientless SSL VPN access to remote users via an HTTPS-enabled web browser, and does not require a client

Enter this command to configure one or more tunneling modes. You must configure at least one tunneling mode for users to connect over a VPN tunnel.

The following example shows how to configure clientless SSL VPN and IPsec tunneling modes for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# vpn-tunnel-protocol webvpn
hostname(config-username)# vpn-tunnel-protocol IPsec
hostname(config-username)
```

### Restrict Remote User Access

Configure the **group-lock** attribute with the **value** keyword to restrict remote users to access only through the specified, preexisting connection profile. Group-lock restricts users by checking whether the group configured in the VPN client is the same as the connection profile to which the user is assigned. If it is not, the ASA prevents the user from connecting. If you do not configure group-lock, the ASA authenticates users without regard to the assigned group.

To remove the **group-lock** attribute from the running configuration, enter the **no** form of this command. This option allows inheritance of a value from the group policy. To disable group-lock, and to prevent inheriting a group-lock value from a default or specified group policy, enter the **group-lock** command with the **none** keyword.

```
hostname(config-username)# group-lock {value tunnel-grp-name | none}
hostname(config-username)# no group-lock
hostname(config-username)
```

The following example shows how to set group lock for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# group-lock value tunnel-group-name
hostname(config-username)
```

### Enable Password Storage for Software Client Users

Specify whether to let users store their login passwords on the client system. Password storage is disabled by default. Enable password storage only on systems that you know to be in secure sites. To disable password storage, enter the **password-storage** command with the **disable** keyword. To remove the password-storage attribute from the running configuration, enter the **no** form of this command. This enables inheritance of a value for password-storage from the group policy.

```
hostname(config-username)# password-storage {enable | disable}
hostname(config-username)# no password-storage
hostname(config-username)
```

This command has no bearing on interactive hardware client authentication or individual user authentication for hardware clients.

The following example shows how to enable password storage for the user named anyuser:

```
hostname(config)# username anyuser attributes
hostname(config-username)# password-storage enable
hostname(config-username)
```

# Best Practices for Configuring and Adjusting VPN Filter ACL

This section provides best practices to follow while updating an existing VPN filter ACL without interrupting the traffic.

### Update an Existing VPN Filter ACL

Follow these steps when you want to update a vpn-filter ACL applied on the ASA device:

1. Create a new vpn-filter ACL on your system (Example: *new_acl.txt*).
2. Download the current vpn-filter ACL from the device (Example: *old_acl.txt*).
3. Create modify instructions for the ACL:

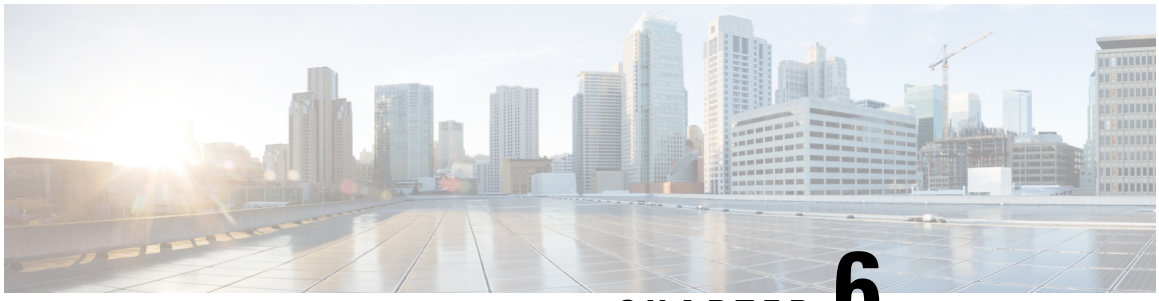
```
* Add update in-progress to ACL remark
echo ?access-list <name> line 1 ACL update in-progress? > push.txt
* Delete old rules
sed ?s/^/no /g? old.acl >> push.txt
 * Add new rules
cat new.acl >> push.txt
 * Remove update in-progress to ACL remark
echo ?no access-list <name> ACL update in-progress? >> push.txt
```

4. Upload push.txt to the device.

### Replace an existing VPN Filter ACL with a new one

Follow these steps to replace a vpn-filter ACL that is applied on the ASA device:

1. Creating a new vpn-filter ACL each time you want to replace an existing one.
2. Update the group-policy with the vpn-filter ACL.
3. Delete the old vpn-filter ACL applied on the device.



CHAPTER 6

# IP Addresses for VPNs


- Configure an IP Address Assignment Policy, on page 185
- Configure Local IP Address Pools, on page 187
- Configure AAA Addressing, on page 189
- Configure DHCP Addressing, on page 190


## Configure an IP Address Assignment Policy

The ASA can use one or more of the following methods for assigning IP addresses to remote access clients. If you configure more than one address assignment method, the ASA searches each of the options until it finds an IP address. By default, all methods are enabled.

- aaa Retrieves addresses from an external authentication, authorization, and accounting server on a per-user basis. If you are using an authentication server that has IP addresses configured, we recommend using this method. This method is available for IPv4 and IPv6 assignment policies.
- dhcp Obtains IP addresses from a DHCP server. If you want to use DHCP, you must configure a DHCP server. You must also define the range of IP addresses that the DHCP server can use. This method is available for IPv4 assignment policies.
- **local** Internally configured address pools are the easiest method of address pool assignment to configure. If you choose local, you must also use the **ip-local-pool** command to define the range of IP addresses to use. This method is available for IPv4 and IPv6 assignment policies.
  - Allow the reuse of an IP address so many minutes after it is released—Delays the reuse of an IP address after its return to the address pool. Adding a delay helps to prevent problems firewalls can experience when an IP address is reassigned quickly. By default the ASA does not impose a delay. This configurable element is available for IPv4 assignment policies.

**Note** When you configure both a DHCP server and a local IP address pool for address assignment, automatic fallback to the local pool (if the DHCP server is unavailable) works only for SSL-based remote access VPN connections.

For IPsec-based remote access VPN connections, this fallback may not complete in time if the DHCP server is unreachable, which can cause the connection to fail.

To avoid connection issues with IPsec-based remote access VPN, do one of the following:

- Make sure the DHCP server is reachable, or
- Use only a local IP address pool for address assignment.

Use one of the following methods to specify a way to assign IP addresses to remote access clients.

# Configure IPv4 Address Assignments

### Procedure

Enable an address assignment method for the ASA to use when assigning IPv4 address to VPN connections. The available methods to obtain an IP address are from a AAA server, DHCP server, or a local address pool. All of these methods are enabled by default.

**vpn-addr-assign** {**aaa** | **dhcp** | **local** [**reuse-delay** *minutes*]}

**Example:**

For example, you can configure the reuse of an IP address for between 0 and 480 minutes after the IP address has been released.

```
hostname(config)#vpn-addr-assign aaa
hostname(config)#vpn-addr-assign local reuse-delay 180
```

This example uses the no form of the command to disable an address assignment method.

```
hostname(config)# no vpn-addr-assign dhcp
```

# Configure IPv6 Address Assignments

### Procedure

Enable an address assignment method for the ASA to use when assigning IPv6 address to VPN connections. The available methods to obtain an IP address are from a AAA server or a local address pool. Both of these methods are enabled by default.

**ipv6-vpn-addr-assign** {**aaa** | local}

**Example:**

```
hostname(config)# ipv6-vpn-addr-assign aaa
```

This example uses the no form of the command to disable an address assignment method.

```
hostname(config)# no ipv6-vpn-addr-assign local
```

# View Address Assignment Methods

### Procedure

Use one of these methods to view the address assignment method configured on the ASA:

- View IPv4 Address Assignments

  Show the configured address assignment method. The configured address method could be aaa, dhcp, or local.

  ```
  show running-config all vpn-addr-assign
  vpn-addr-assign aaa
  vpn-addr-assign dhcp
  vpn-addr-assign local
  ```

- View IPv6 Address Assignments

  Show the configured address assignment method. Configured address methods could be aaa or local.

  ```
  show running-config all ipv6-vpn-addr-assign
  ipv6-vpn-addr-assign aaa
  ipv6-vpn-addr-assign local reuse-delay 0
  ```

# Configure Local IP Address Pools

To configure IPv4 address pools to use for VPN remote access tunnels, enter the **ip local pool** command in global configuration mode. To delete address pools, enter the **no** form of this command.

To configure IPv6 address pools to use for VPN remote access tunnels, enter the **ipv6 local pool** command in global configuration mode. To delete address pools, enter the **no** form of this command.

The ASA uses address pools based on the connection profile or group policy for the connection. The order in which you specify the pools is important. If you configure more than one address pool for a connection profile or group policy, the ASA uses them in the order in which you added them to the ASA.

If you assign addresses from a non-local subnet, we suggest that you add pools that fall on subnet boundaries to make adding routes for these networks easier.

**Note** When you modify existing address pools currently in use within an active tunnel-group (that is, open to end users for connections), you must perform the change in a change window and ensure the following:

- The connected users are logged off.
- The address pools are removed from the tunnel-group and modified as required.
- The modified address pools are then added back under the tunnel-group.

If an address pool is not modified in this manner, it may cause inconsistencies in the ASA's behaviour.

# Configure Local IPv4 Address Pools

**Note** When you want to modify an existing address-pool currently in use within an active tunnel-group (i.e. open to end users for connections) on the CLI, it is recommended to perform this change in a change window. The users connected should be logged off, the address pool should be removed from the tunnel-group, modified as required and then added back under the tunnel-group. If not done in this manner, it may cause inconsistencies in the ASA's behavior.

### Procedure

**Step 1** Configure IP address pools as the address assignment method. Enter the **vpn-addr-assign** command with the **local** argument.

**Example:**

```
hostname(config)# vpn-addr-assign local
```

**Step 2** Configure an address pool. The command names the pool, specifies a range of IPv4 addresses and the subnet mask.

**ip local pool***poolname first_address-last_address***mask***mask*

**Example:**

This example configures an IP address pool named *firstpool*.The starting address is 10.20.30.40and the ending address is 10.20.30.50.The network mask is 255.255.255.0.

```
hostname(config)# ip local pool firstpool 10.20.30.40-10.20.30.50 mask 255.255.255.0
```

This example deletes the IP address pool named **firstpool**.

```
hostname(config)# no ip local pool firstpool
```

# Configure Local IPv6 Address Pools

### Procedure

**Step 1** Configures IP address pools as the address assignment method, enter the ipv6-**vpn-addr-assign** command with the **local** argument.

**Example:**

```
hostname(config)# ipv6-vpn-addr-assign local
```

**Step 2** Configures an address pool. The command names the pool, identifies the starting IPv6 address, the prefix length in bits, and the number of addresses to use in the range.

**ipv6 local pool** *pool_name starting_address prefix_length number_of_addresses*

**Example:**

This example configures an IP address pool named *ipv6pool*. The starting address is 2001:DB8::1, the prefix length is 32 bits, and the number of addresses to use in the pool is 100.

```
hostname(config)# ipv6 local pool ipv6pool 2001:DB8::1/32 100
```

This example deletes the IP address pool named *ipv6pool*.

```
hostname(config)# no ipv6 local pool ipv6pool
```

# Configure AAA Addressing

To use a AAA server to assign addresses for VPN remote access clients, you must first configure a AAA server or server group. See the **aaa-server protocol** command in the command reference.

In addition, the user must match a connection profile configured for RADIUS authentication.

The following examples illustrate how to define a AAA server group called RAD2 for the tunnel group named firstgroup. It includes one more step than is necessary, in that previously you might have named the tunnel group and defined the tunnel group type. This step appears in the following example as a reminder that you have no access to subsequent tunnel-group commands until you set these values.

An overview of the configuration that these examples create follows:

```
hostname(config)# vpn-addr-assign aaa
hostname(config)# tunnel-group firstgroup type ipsec-ra
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config)# authentication-server-group RAD2
```

To configure AAA for IP addressing, perform the following steps:

### Procedure

**Step 1** To configure AAA as the address assignment method, enter the **vpn-addr-assign** command with the **aaa** argument:

```
hostname(config)# vpn-addr-assign aaa
hostname(config)#
```

**Step 2** To establish the tunnel group called firstgroup as a remote access or LAN-to-LAN tunnel group, enter the **tunnel-group** command with the **type** keyword. The following example configures a remote access tunnel group.

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

**Step 3** To enter general-attributes configuration mode, which lets you define a AAA server group for the tunnel group called firstgroup, enter the **tunnel-group** command with the **general-attributes** argument.

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

**Step 4** To specify the AAA server group to use for authentication, enter the **authentication-server-group** command.

```
hostname(config-general)# authentication-server-group RAD2
hostname(config-general)#
```

#### What to do next

This command has more arguments that this example includes. For more information, see the command reference.

# Configure DHCP Addressing

To use DHCP to assign addresses for VPN clients, you must first configure a DHCP server and the range of IP addresses that the DHCP server can use. Then you define the DHCP server on a connection profile basis. Optionally, you can also define a DHCP network scope in the group policy associated with a connection profile or username.

The following example defines the DHCP server at 172.33.44.19 for the connection profile named **firstgroup**. The example also defines a DHCP network scope of 10.100.10.1 for the group policy called **remotegroup**. (The group policy called remotegroup is associated with the connection profile called firstgroup). If you do not define a network scope, the DHCP server assigns IP addresses in the order of the address pools configured. It goes through the pools until it identifies an unassigned address.

### Before you begin

You can only use an IPv4 address to identify a DHCP server to assign client addresses. In addition, DHCP options are not forwarded to users, they receive an address assignment only.

## Procedure

**Step 1** Configure IP address pools as the address assignment method.

**vpn-addr-assign dhcp**

**Step 2** Establish the connection profile called **firstgroup** as a remote access connection profile.

**tunnel-group firstgroup type remote-access**

**Step 3** Enter the general-attributes configuration mode for the connection profile so that you can configure a DHCP server.

**tunnel-group firstgroup general-attributes**

**Step 4** Define the DHCP server by IPv4 address, then exit tunnel group configuration mode.

**dhcp-server** *IPv4_address_of_DHCP_server*

You can not define a DHCP server by an IPv6 address. You can specify more than one DHCP server address for a connection profile. Enter the dhcp-server command. This command allows you to configure the ASA to send additional options to the specified DHCP servers when it is trying to get IP addresses for VPN clients.

**Example:**

The example configures a DHCP server at IP address 172.33.44.19. Then, exit tunnel group configuration mode .

```
hostname(config-general)# dhcp-server 172.33.44.19
hostname(config-general)# exit
hostname(config)#
```

**Step 5** If the group does not already exist, create an internal group policy called **remotegroup**.

```
hostname(config)# group-policy remotegroup internal
```

**Step 6** (Optional.) Enter group-policy attributes configuration mode and define the DHCP network scope.

**dhcp-network-scope** *ip_address*

If you configure DHCP servers for the address pool in the connection profile, the DHCP scope identifies the subnets to use for the pool for this group. The DHCP server must also have addresses in the same subnet identified by the scope. The scope allows you to select a subset of the address pools defined in the DHCP server to use for this specific group.

If you do not define a network scope, the DHCP server assigns IP addresses in the order of the address pools configured. It goes through the pools until it identifies an unassigned address.

**Note**
To specify a scope, enter a routeable address on the same subnet as the desired pool, but not within the pool. The DHCP server determines which subnet this IP address belongs to and assigns an IP address from that pool.

We recommend using the IP address of an interface whenever possible for routing purposes. For example, if the pool is 10.100.10.2-10.100.10.254, and the interface address is 10.100.10.1/24, use 10.100.10.1 as the DHCP scope. Do not use the network number. You can use DHCP for IPv4 addressing only. If the address you choose is not an interface address, you might need to create a static route for the scope address.

**Example:**

The following example enters attribute configuration mode for remotegroup and sets the DHCP scope to 10.100.10.1.

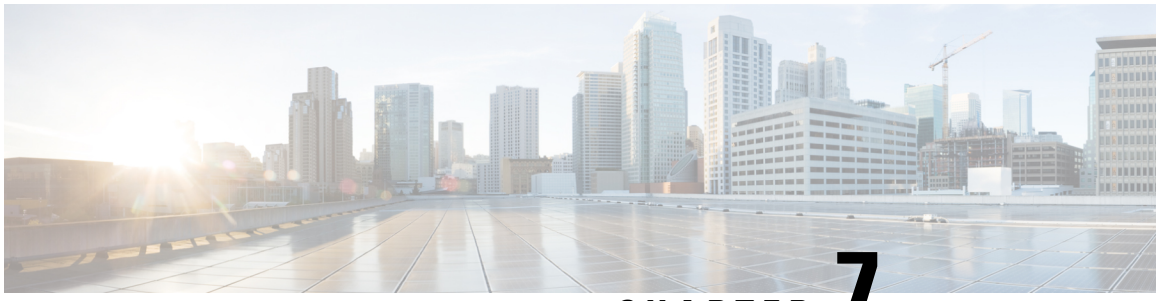
```
hostname(config)# group-policy remotegroup attributes
hostname(config-group-policy)# dhcp-network-scope 10.100.10.1
```

---

**Example**

A summary of the configuration that these examples create follows:

```
hostname(config)# vpn-addr-assign dhcp
hostname(config)# tunnel-group firstgroup type remote-access
hostname(config)# tunnel-group firstgroup general-attributes
hostname(config-general)# dhcp-server 172.33.44.19
hostname(config-general)# exit
hostname(config)# group-policy remotegroup internal
hostname(config)# group-policy remotegroup attributes
hostname(config-group-policy)# dhcp-network-scope 10.100.10.1
```



CHAPTER 7

# Remote Access IPsec VPNs


- Overview of Remote Access IPsec VPNs, on page 193
- Licensing Requirements for AnyConnect VPN Module of Cisco Secure Client, on page 195
- Restrictions for Remote Access IPsec VPN, on page 195
- Configure Remote Access IPsec VPNs, on page 195
- VPN Authentication Using Post-Quantum Pre-Shared Keys, on page 202
- Configuration Examples for Remote Access IPsec VPNs, on page 207
- Configuration Examples for Standards-Based IPSec IKEv2 Remote Access VPN in Multicontext Mode, on page 208
- Configuration Examples for Secure Client IPSec IKEv2 Remote Access VPN in Multicontext Mode, on page 209
- Feature History for Remote Access VPNs, on page 210


# 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 198.

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 196
- Configure an Address Pool, on page 197
- Add a User, on page 198
- Create an IKEv1 Transform Set or IKEv2 Proposal, on page 198
- Define a Tunnel Group, on page 199
- Create a Dynamic Crypto Map, on page 201
- Create a Crypto Map Entry to Use the Dynamic Crypto Map, on page 201

# 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 204.
- 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 205.
- 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 10: 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 204 |
| 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 205 |

| Step | Action | More Information |
|---|---|---|
| 4 | Configure the ASA tunnel group. | Configure VPN Authentication on ASA Using Post-Quantum Pre-Shared Keys, on page 205 |
| 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.<br>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 202 and Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 203.

### 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 202 and Guidelines and Limitations for Using Post-Quantum Pre-Shared Keys for VPN Authentication, on page 203.

### 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

# 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.

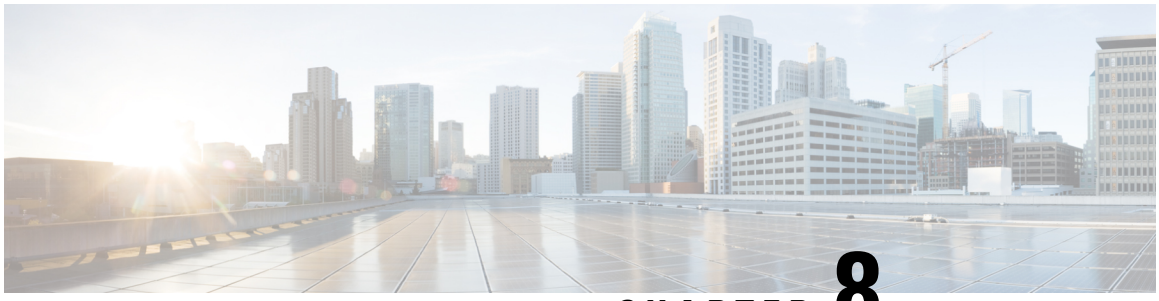
```
hostname/CTX3(config)#crypto ikev2 enable outside client-services port 443
hostname/CTX3(config)#crypto ikev2 remote-access trustpoint ASDM_TrustPoint0
hostname/CTX3(config)#crypto dynamic-map SYSTEM_DEFAULT_CRYPTO_MAP 65535 set ikev2
ipsec-proposal AES256 AES192 AES 3DES DES
hostname/CTX3(config)#crypto 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. |

| Feature Name | Releases | Feature Information |
|---|---|---|
| Automatic mobike support for remote access VPNs. | 9.8(1) | Added Mobile IKE (mobike) support for IPsec IKEv2 RA VPNs. Mobike is always on.<br>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.<br>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. |
| Deprecations of IKE/IPsec encryption and integrity/PRF ciphers<br>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):<br>• 3DES encryption<br>• DES encryption<br>• MD5 integrity<br>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). |



CHAPTER 8

# LAN-to-LAN IPsec VPNs

A LAN-to-LAN VPN connects networks in different geographic locations.

You can create LAN-to-LAN IPsec connections with Cisco peers and with third-party peers that comply with all relevant standards. These peers can have any mix of inside and outside addresses using IPv4 and IPv6 addressing.

ASA does not allow locally sourced traffic other than ping to go over the VPN tunnel.

This chapter describes how to build a LAN-to-LAN VPN connection.


- Summary of the Configuration, on page 213
- Guidelines for Site-to-Site VPNs, on page 214
- Configure Site-to-Site VPN in Multi-Context Mode, on page 214
- Configure Interfaces, on page 215
- Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface, on page 216
- Create an IKEv1 Transform Set, on page 222
- Create an IKEv2 Proposal, on page 223
- Configure an ACL, on page 224
- Define a Tunnel Group, on page 225
- Create a Crypto Map and Applying It To an Interface, on page 226
- Overview of Dynamic Site-to-Site VPNs, on page 229


# Summary of the Configuration

This section provides a summary of the example LAN-to-LAN configuration this chapter describes. Later sections provide step-by-step instructions.

```
hostname(config)# interface ethernet0/0
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)# nameif outside
hostname(config-if)# no shutdown
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)# encryption aes
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)# group 2
hostname(config-ikev1-policy)# lifetime 43200
hostname(config)# crypto ikev1 enable outside
hostname(config)# crypto ikev2 policy 1
```

```
hostname(config-ikev2-policy)# # encryption aes
hostname(config-ikev2-policy)# group 2
hostname(config-ikev12-policy)# prf sha
hostname(config-ikev2-policy)# lifetime 43200
hostname(config)# crypto ikev2 enable outside
hostname(config)#  crypto ipsec ikev1 transform-set FirstSet esp-aes esp-sha-hmac
hostname(config)# crypto ipsec ikev2 ipsec-proposal secure
hostname(config-ipsec-proposal)# protocol esp encryption aes
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0 150.150.0.0
 255.255.0.0
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1 pre-shared-key 44kkaol59636jnfx
hostname(config)# crypto map abcmap 1 match address l2l_list
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)# crypto map abcmap 1 set ikev1 transform-set FirstSet
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)# crypto map abcmap interface outside
hostname(config)# write memory
```

# Guidelines for Site-to-Site VPNs

- When you configure policy-based site-to-site VPNs on the ASA, ensure that outbound ESP traffic (protocol 50) is explicitly allowed in any cloud or network security rules. If this traffic is blocked, VPN packets can be dropped between ASA peers even when they appear on the outside interfaces of both devices.

  For example, in IBM Cloud security rules, add an outbound rule similar to:

  ```
  Protocol: ESP (50)
  Source: Any
  Destination: Any
  Action: Allow
  ```

  Use packet capture on the ASA outside interface to view the ESP packets.

# Configure Site-to-Site VPN in Multi-Context Mode

Follow these steps to allow site-to-site support in multi-mode. By performing these steps, you can see how resource allocation breaks down.

### Procedure

**Step 1** To configure the VPN in multi-mode, configure a resource class and choose VPN licenses as part of the allowed resource. The "Configuring a Class for Resource Management" provides these configuration steps. The following is an example configuration:

```
class ctx1
 limit-resource VPN Burst Other 100
 limit-resource VPN Other 1000
```

**Step 2** Configure a context and make it a member of the configured class that allows VPN licenses. The following is an example configuration:

```
context context1
 member ctx1
 allocate-interface GigabitEthernet3/0.2
 allocate-interface GigabitEthernet3/1.2
 allocate-interface Management0/0
 config-url disk0:/sm_s2s_ik1_ip4_no_webvpn.txt
 join-failover-group 1
```

**Step 3** Configure connection profiles, policies, crypto maps, and so on, just as you would with single context VPN configuration of site-to-site VPN.

# Configure Interfaces

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.

**Note** The ASA's outside interface address (for both IPv4/IPv6) cannot overlap with the private side address space.

### Procedure

**Step 1** To enter Interface configuration mode, in global configuration mode enter the **interface** command with the default name of the interface to configure. In the following example the interface is ethernet0.

```
hostname(config)# interface ethernet0/0
hostname(config-if)#
```

**Step 2** To set the IP address and subnet mask for the interface, enter the **ip address** command. In the following example the IP address is 10.10.4.100 and the subnet mask is 255.255.0.0.

```
hostname(config-if)# ip address 10.10.4.100 255.255.0.0
hostname(config-if)#
```

**Step 3** To name the interface, enter the **nameif** command, maximum of 48 characters. You cannot change this name after you set it. In the following example the name of the ethernet0 interface is outside.

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

**Step 4** To enable the interface, enter the **no** version of the **shutdown** command. By default, interfaces are disabled.

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

**Step 5** To save your changes, enter the **write memory** command:

```
hostname(config-if)# write memory
hostname(config-if)#
```

**Step 6** To configure a second interface, use the same procedure.

# Configure ISAKMP Policy and Enable ISAKMP on the Outside Interface

ISAKMP is the negotiation protocol that lets two hosts agree on how to build an IPsec security association (SA). It provides a common framework for agreeing on the format of SA attributes. This includes negotiating with the peer about the SA, and modifying or deleting the SA. ISAKMP separates negotiation into two phases: Phase 1 and Phase 2. Phase 1 creates the first tunnel, which protects later ISAKMP negotiation messages. Phase 2 creates the tunnel that protects data.

IKE uses ISAKMP to setup the SA for IPsec to use. IKE creates the cryptographic keys used to authenticate peers.

The ASA supports IKEv1 for connections from the legacy Cisco VPN client, and IKEv2 for the AnyConnect VPN client.

To set the terms of the ISAKMP negotiations, you create an IKE policy, which includes the following:

- The authentication type required of the IKEv1 peer, either RSA signature using certificates or preshared key (PSK).
- 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 determine the strength of the encryption-key-determination algorithm. The ASA uses this algorithm to derive the encryption and hash keys.
- For IKEv2, a separate pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption.
- A limit to the time the ASA uses an encryption key before replacing it.

With IKEv1 policies, for each parameter, you set one value. For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

The following sections provide procedures for creating IKEv1 and IKEv2 policies and enabling them on an interface:

• Configure ISAKMP Policies for IKEv1 Connections, on page 217
• Configure ISAKMP Policies for IKEv2 Connections, on page 218

# Configure ISAKMP Policies for IKEv1 Connections

To configure ISAKMP policies for IKEv1 connections, use the **crypto ikev1 policy** priority command to enter IKEv1 policy configuration mode where you can configure the IKEv1 parameters.

### Procedure

**Step 1** Enter IPsec IKEv1 policy configuration mode. For example:

```
hostname(config)# crypto ikev1 policy 1
hostname(config-ikev1-policy)#
```

**Step 2** Set the authentication method. The following example configures a preshared key:

```
hostname(config-ikev1-policy)# authentication pre-share
hostname(config-ikev1-policy)#
```

**Step 3** Set the encryption method. The following example configures :

```
hostname(config-ikev1-policy)# encryption aes
hostname(config-ikev1-policy)#
```

**Step 4** Set the HMAC method. The following example configures SHA-1:

```
hostname(config-ikev1-policy)# hash sha
hostname(config-ikev1-policy)#
```

**Step 5** Set the Diffie-Hellman group. The following example configures Group 14:

```
hostname(config-ikev1-policy)# group 14
hostname(config-ikev1-policy)#
```

**Step 6** Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):

```
hostname(config-ikev1-policy)# lifetime 43200
hostname(config-ikev1-policy)#
```

**Step 7** Enable IKEv1 on the interface named outside in either single or multiple context mode:

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

**Step 8** To save your changes, enter the **write memory** command:

```
hostname(config)# write memory
hostname(config)#
```

# Configure ISAKMP Policies for IKEv2 Connections

To configure ISAKMP policies for IKEv2 connections, use the **crypto ikev2 policy** priority command to enter IKEv2 policy configuration mode where you can configure the IKEv2 parameters.

### Procedure

**Step 1** Enter IPsec IKEv2 policy configuration mode. For example:

```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)#
```

**Step 2** Set the encryption method. The following example configures AES :

```
hostname(config-ikev2-policy)# encryption aes
hostname(config-ikev2-policy)#
```

**Step 3** Set the Diffie-Hellman group. The following example configures Group 15:

```
hostname(config-ikev2-policy)# group 15
hostname(config-ikev2-policy)#
```

**Step 4** Set the pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption. The following example configures SHA-1 (an HMAC variant):

```
hostname(config-ikev12-policy)# prf sha
hostname(config-ikev2-policy)#
```

**Step 5** Set the encryption key lifetime. The following example configures 43,200 seconds (12 hours):

```
hostname(config-ikev2-policy)# lifetime seconds 43200
hostname(config-ikev2-policy)#
```

**Step 6** Enable IKEv2 on the interface named outside:

```
hostname(config)# crypto ikev2 enable outside
hostname(config)#
```

**Step 7** To save your changes, enter the **write memory** command:

```
hostname(config)# write memory
hostname(config)#
```

# Multiple Key Exchanges for IKEv2

IKEv2 uses Diffie-Hellman (DH) groups to establish a shared secret between an initiator and a responder. IKEv2 supports additional key exchanges to secure the IPsec communication from quantum computer attacks. Each exchange uses different DH groups. The computed shared secret for the SA setup is a combination of all the keys derived from each exchange. An IKE SA is established after the multiple key exchanges between the IKE peers.

ASA uses seven new transform types for multiple key exchanges:

- Additional Key Exchange 1 (IANA value 6)
- Additional Key Exchange 2 (IANA value 7)
- Additional Key Exchange 3 (IANA value 8)
- Additional Key Exchange 4 (IANA value 9)
- Additional Key Exchange 5 (IANA value 10)
- Additional Key Exchange 6 (IANA value 11)
- Additional Key Exchange 7 (IANA value 12)

You can configure a maximum of seven multiple key exchanges. For each additional key exchange that you configure, you must specify the DH groups. ASA encrypts the intermediate key exchanges using the keys derived from the previous exchange. If the initiator and responder peers do not agree on a DH group, the negotiation fails and a **NO_PROPOSAL_CHOSEN** error notification is sent to the initiator. You can also configure the transform as none. If you choose none, the key exchange does not happen.

For an initiator, if the key exchange method is configured as **none** for an additional key exchange *n*:

- Responder can choose key exchange method as **none** for the additional key exchange *n*.
- Additional key exchange is optional.

For a successful proposal negotiation, all the transforms in the initiator's proposal must match with the transforms in the responder.

In the following example for an initiator:

```
crypto ikev2 policy 1
encryption aes
integrity sha256
group 14
prf sha256
lifetime seconds 120
additional-key-exchange 5
key-exchange-method none
```

Responder must have an additional-key-exchange 5 for the proposal to match.

If the peer does not support additional key exchange, one of the following occurs:

- If the initiator has another IKEv2 proposal that matches with the responder's proposal, an IKEv2 SA is established.
- The peer treats any additional key exchange transform type in the IKE_SA_INIT exchange message as unknown transform type and skips the proposals. The negotiation fails and a **NO_PROPOSAL_CHOSEN** error notification is sent to the initiator.

For more information about this feature, see RFC 9242.

## Guidelines and Limitations for IKEv2 Multiple Key Exchanges

- You can have a maximum of seven multiple key exchanges.
- You cannot use the same DH group in the succeeding key exchanges.

For this feature, ASA does not support:

- IKEv1
- Combination of classical key exchange and post-quantum algorithm-based key exchange.
- Remote access VPN. Only site-to-site VPNs support IKEv2 multiple key exchanges.
- Clustering

## Configure Multiple Key Exchanges for IKEv2

This configuration is optional, you can perform the configuration if you want to secure the IPsec communication from quantum computer attacks.

### Before you begin

- Review the guidelines and limitations. For more information, see Guidelines and Limitations for IKEv2 Multiple Key Exchanges, on page 220.
- Configure the encryption algorithm, hash algorithm, authentication method, and SA lifetime for the IKEv2 policy. For more information, see Configure IKEv1 and IKEv2 Policies, on page 7.

### Procedure

**Step 1** Create an IKEv2 policy.

**crypto ikev2 policy** *policy_index*

The prompt displays the IKEv2 policy configuration mode.

**Example:**

```
hostname(config)# crypto ikev2 policy 1
```

**Step 2** Configure an additional key exchange transform for the IKEv2 policy.

**additional-key-exchange** *<1-7>*

The prompt displays the IKEv2 policy additional key exchange configuration mode. You can configure a maximum of seven key exchange transforms for a policy.

**Example:**

```
hostname(config-ikev2-policy)# additional-key-exchange 1
```

**Step 3** Configure a key exchange method by defining one or more DH groups for the additional key exchange transform.

**key-exchange-method** <*DH_group*>

Specify the DH group as 14, 15, 16, 19, 20, 21, or 31. You can also configure the transform as none. If you choose none, the key exchange does not happen.

**Example:**

```
hostname(config-ikev2-policy-ake)# key-exchange-method 21 31
```

**Step 4** Repeat steps 2 and 3 to configure multiple key exchanges for the IKEv2 policy.

```
hostname(config)# crypto ikev2 policy 1
hostname(config-ikev2-policy)# additional-key-exchange 1
hostname(config-ikev2-policy-ake)# key-exchange-method 21 31
hostname(config-ikev2-policy)# additional-key-exchange 2
hostname(config-ikev2-policy-ake)# key-exchange-method 20 21
hostname(config-ikev2-policy)# additional-key-exchange 3
hostname(config-ikev2-policy-ake)# key-exchange-method 19 20 none
...
```

### What to do next

Verify the configuration. For more information, see Verify IKEv2 Multiple Key Exchanges Configurations, on page 221.

## Verify IKEv2 Multiple Key Exchanges Configurations

Use the following show commands to view or verify the IKEv2 multiple key exchanges configurations:

- **show running-config crypto ikev2**

```
crypto ikev2 policy 1
encryption aes
integrity sha256
group 14
prf sha256
lifetime seconds 120
additional-key-exchange 1
key-exchange-method 21 31
additional-key-exchange 2
key-exchange-method 20 21
...
```

- **show crypto ikev2 sa detail**

```
IKEv2 SAs:
Session-id:4, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id Local Remote fvrf/ivrf Status Role
```

```
41567725 192.168.15.1/500 192.168.15.2/500 READY INITIATOR
Encr: AES-CBC, keysize: 128, Hash: SHA96, DH Grp:14, Auth sign: PSK, Auth verify: PSK
Additional Key Exchange Group: AKE1: 31 AKE2: 21 AKE3: 20 AKE4: 19 AKE5: 16 AKE6: 15
AKE7: 14
Life/Active Time: 120/5 sec
Session-id: 4
Status Description: Negotiation done
Local spi: 6BB6B7BFA0BAADF4 Remote spi: 7030C7xxx xxxxxxE9DBDE77EB
Local id: 192.168.15.1
Remote id: 192.168.15.2
Local req mess id: 9 Remote req mess id: 0
Local next mess id: 9 Remote next mess id: 0
Local req queued: 9 Remote req queued: 0
Local window: 1 Remote window: 1
DPD configured for 10 seconds, retry 2
NAT-T is not detected
IKEv2 Fragmentation Configured MTU: 576 bytes, Overhead: 28 bytes, Effective MTU: 548
bytes
Parent SA Extended Status:
Delete in progress: FALSE
Marked for delete: FALSE
Child sa: local selector 20.0.0.0/0 - 20.0.0.255/65535
remote selector 30.0.0.0/0 - 30.0.0.255/65535
ESP spi in/out: 0x4a7d5da2/0x56a28fa8
AH spi in/out: 0x0/0x0
CPI in/out: 0x0/0x0
Encr: AES-CBC, keysize: 128, esp_hmac: SHA96
ah_hmac: None, comp: IPCOMP_NONE, mode tunnel
```

# Create an IKEv1 Transform Set

An IKEv1 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.

The table below lists valid encryption and authentication methods.

***Table 11: Valid Encryption and Authentication Methods***

| Valid Encryption Methods | Valid Authentication Methods |
|---|---|
| | esp-sha-hmac (default) |
| esp-aes (128-bit encryption) (default) | |
| esp-aes-192 | |
| esp-aes-256 | |
| esp-null | |

Tunnel Mode is the usual way to implement IPsec between two ASAs that are connected over an untrusted network, such as the public Internet. Tunnel mode is the default and requires no configuration.

To configure a transform set, perform the following site-to-site tasks in either single or multiple context mode:

**Procedure**

**Step 1** In global configuration mode enter the **crypto ipsec ikev1 transform-set** command. The following example configures a transform set with the name FirstSet, esp-aes encryption , and esp-sha-hmac authentication. The syntax is as follows:

esp-sha-hmac (default)

**crypto ipsec ikev1 transform-set** *transform-set-nameencryption-method authentication-method*

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

**Step 2** Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

# Create an IKEv2 Proposal

For IKEv2, you can configure multiple encryption and authentication types, and multiple integrity algorithms for a single policy. The ASA orders the settings from the most secure to the least secure and negotiates with the peer using that order. This allows you to potentially send a single proposal to convey all the allowed transforms instead of the need to send each allowed combination as with IKEv1.

The table below lists valid IKEv2 encryption and authentication methods.

***Table 12: Valid IKEv2 Encryption and Integrity Methods***

| Valid Encryption Methods | Valid Integrity Methods |
|---|---|
| | sha (default) |
| aes (default) - AES with a 128-bit key. | |
| aes-192 | |
| aes-256 | |

**Note** When ASA acts as an IKEv2 responder and receives multiple proposals in the initial packet, it ignores unknown transform types. The device then evaluates the remaining proposals. If a matching proposal is found, the VPN connection is established. If no match is found, the device drops the connection.

To configure an IKEv2 proposal, perform the following tasks in either single or multiple context mode:

**Procedure**

**Step 1** In global configuration mode, use the **crypto ipsec ikev2 ipsec-proposal** command to enter ipsec proposal configuration mode where you can specify multiple encryption and integrity types for the proposal. In this example, secure is the name of the proposal:

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

**Step 2** Then enter a protocol and encryption types. ESP is the only supported protocol. For example:

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

hostname(config-ipsec-proposal)#
```

**Step 3** Enter an integrity type. For example:

```
hostname(config-ipsec-proposal)# protocol esp integrity sha-1
hostname(config-ipsec-proposal)#
```

**Step 4** Save your changes.

# Configure an ACL

The ASA uses access control lists to control network access. By default, the adaptive security appliance denies all traffic. You need to configure an ACL that permits traffic. For more information, see "Information About Access Control Lists" in the general operations configuration guide.

The ACLs that you configure for this LAN-to-LAN VPN control connections are based on the source and translated destination IP addresses and, optionally, ports. Configure ACLs that mirror each other on both sides of the connection.

An ACL for VPN traffic uses the translated address.

**Note** For more information on configuring an ACL with a VPN filter, see the Specify a VLAN for Remote Access or Apply a Unified Access Control Rule to the Group Policy, on page 139.

**Procedure**

**Step 1** Enter the **access-list extended** command.

**access-list** *listname* **extended permit ip** *source-ipaddress source-netmask destination-ipaddress destination-netmask*

The following example configures an ACL named l2l_list that lets traffic from IP addresses in the 192.168.0.0 network travel to the 150.150.0.0 network.

```
hostname(config)# access-list l2l_list extended permit ip 192.168.0.0 255.255.0.0 150.150.0.0
 255.255.0.0
hostname(config)#
```

**Step 2** Configure an ACL for the ASA on the other side of the connection that mirrors the ACL.

Subnets that are defined in an ACL in a crypto map, or in two different crypto ACLs that are attached to the same crypto map, should not overlap.

In the following example, the prompt for the peer is hostname2.

```
hostname2(config)# access-list l2l_list extended permit ip 150.150.0.0 255.255.0.0 192.168.0.0
 255.255.0.0
hostname(config)#
```

# Define a Tunnel Group

A tunnel group is a set of records that contain 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: DefaultRAGroup, which is the default IPsec remote-access tunnel group, and DefaultL2Lgroup, which is the default IPsec LAN-to-LAN tunnel group. You can modify them, but not delete them.

The main difference between IKE versions 1 and 2 lies in terms of the authentication method they allow. IKEv1 allows only one type of authentication at both VPN ends (that is, either preshared key or certificate). However, IKEv2 allows asymmetric authentication methods to be configured (that is, preshared key authentication for the originator but certificate authentication for the responder) using separate local and remote authentication CLIs. Therefore, with IKEv2 you have asymmetric authentication, in which one side authenticates with one credential and the other side uses another credential (either a preshared key or certificate).

You can also create one or more new tunnel groups to suit your environment. 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.

To establish a basic LAN-to-LAN connection, you must set two attributes for a tunnel group:

- Set the connection type to IPsec LAN-to-LAN.
- Configure an authentication method for the IP address (that is, a preshared key for IKEv1 and IKEv2).

## Procedure

**Step 1** To set the connection type to IPsec LAN-to-LAN, enter the **tunnel-group** command.

The syntax is **tunnel-group** *name* **type** *type,* where name is the name you assign to the tunnel group, and type is the type of tunnel. The tunnel types as you enter them in the CLI are:

- **remote-access** (IPsec, SSL, and clientless SSL remote access)
- **ipsec-l2l** (IPsec LAN-to-LAN)

In the following example, the name of the tunnel group is the IP address of the LAN-to-LAN peer, 10.10.4.108.

```
hostname(config)# tunnel-group 10.10.4.108 type ipsec-l2l
hostname(config)#
```

**Note**
LAN-to-LAN tunnel groups that have names that are not IP addresses can be used only if the tunnel authentication method is Digital Certificates and/or the peer is configured to use Aggressive Mode.

**Step 2** To set the authentication method to use a preshared key, enter the ipsec-attributes mode and then enter the **ikev1pre-shared-key** command to create the preshared key. You need to use the same preshared key on both ASAs for this LAN-to-LAN connection.

The key is an alphanumeric string of 1-128 characters.

In the following example, the IKEv1 preshared key is 44kkaol59636jnfx:

```
hostname(config)# tunnel-group 10.10.4.108 ipsec-attributes
hostname(config-tunnel-ipsec)# ikev1-pre-shared-key 44kkaol59636jnfx
```

**Step 3** Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

To verify that the tunnel is up and running, use the **show vpn-sessiondb summary**, **show vpn-sessiondb detail l2l**, or **show crypto ipsec sa** command.

# Create a Crypto Map and Applying It To an Interface

Crypto map entries pull together the various elements of IPsec security associations, including the following:

- Which traffic IPsec should protect, which you define in an ACL.
- Where to send IPsec-protected traffic, by identifying the peer.
- What IPsec security applies to this traffic, which a transform set specifies.
- The local address for IPsec traffic, which you identify by applying the crypto map to an interface.

For IPsec to succeed, both peers must have crypto map entries with compatible configurations. For two crypto map entries to be compatible, they must, at a minimum, meet the following criteria:

- The crypto map entries must contain compatible crypto ACLs (for example, mirror image ACLs). If the responding peer uses dynamic crypto maps, the entries in the ASA crypto ACL must be “permitted” by the peer’s crypto ACL.

- The crypto map entries each must identify the other peer (unless the responding peer is using a dynamic crypto map).
- The crypto map entries must have at least one transform set in common.

If you create more than one crypto map entry for a given interface, use the sequence number (seq-num) of each entry to rank it: the lower the seq-num, the higher the priority. At the interface that has the crypto map set, the ASA evaluates traffic against the entries of higher priority maps first.

If Reverse Route Injection (RRI) is applied to a crypto map, that map must be unique to one interface on the ASA. In other words, the same crypto map cannot be applied to multiple interfaces. If more than one crypto map is applied to multiple interfaces, routes may not be cleaned up correctly. If multiple interfaces require a crypto map, each route must use a uniquely defined map.



**Note** Crypto maps should use consistent address families for peers and next-hops. Ensure thet the RRI routes have matching address types such as IPv4 with IPv4 or IPv6 with IPv6. ASA does not support configurations that mix address families, as it prevents route installation.

Create multiple crypto map entries for a given interface if either of the following conditions exist:

- Different peers handle different data flows.
- You want to apply different IPsec security to different types of traffic (to the same or separate peers), for example, if you want traffic between one set of subnets to be authenticated, and traffic between another set of subnets to be both authenticated and encrypted. In this case, define the different types of traffic in two separate ACLs, and create a separate crypto map entry for each crypto ACL.

### Applying a Crypto Map on Multiple Interfaces

For a dual ISP, you can apply a crypto map to the external and backup interfaces on the ASA. The originate-only option is not available when you use this configuration. You must use Virtual Tunnel Interface (VTI) if you need this redundancy.

When you use a crypto map on multiple interfaces:

- You must have a routing protocol or route tracking.
- Ensure that the remote side also uses routing protocols.
- You must carefully choose multiple interfaces for the same crypto map as ASA allows a connection from a remote site on the interface with the less preferred route.

To create a crypto map and apply it to the outside interface in global configuration mode, perform the following steps in either single or multiple context mode:

## Procedure

**Step 1** To assign an ACL to a crypto map entry, enter the **crypto map match address** command.

The syntax is **crypto map** map-name seq-num **match address** aclname. In the following example the map name is abcmap, the sequence number is 1, and the ACL name is **`l2l_list`**.

```
hostname(config)# crypto map abcmap 1 match address l2l_list
hostname(config)#
```

**Step 2** To identify the peer (s) for the IPsec connection, enter the **crypto map set peer** command.

The syntax is **crypto map** map-name seq-num **set peer** {ip_address1 | hostname1}[... ip_address10 | hostname10]. In the following example the peer name is 10.10.4.108.

```
hostname(config)# crypto map abcmap 1 set peer 10.10.4.108
hostname(config)#
```

**Step 3** To specify an IKEv1 transform set for a crypto map entry, enter the **crypto map ikev1 set transform-set** command.

The syntax is **crypto map** map-name seq-num **ikev1 set transform-set** transform-set-name. In the following example, the transform set name is FirstSet.

```
hostname(config)# crypto map abcmap 1 set transform-set FirstSet
hostname(config)#
```

**Step 4** To specify an IKEv2 proposal for a crypto map entry, enter the crypto map ikev2 set ipsec-proposal command:

The syntax is **crypto map** map-name seq-num set **ikev2 ipsec-proposal proposal-name**. In the following example, the proposal name is secure.

With the **crypto map** command, you can specify multiple IPsec proposals for a single map index. In that case, multiple proposals are transmitted to the IKEv2 peer as part of the negotiation, and the order of the proposals is determined by the administrator upon the ordering of the crypto map entry.

**Note**
If combined mode (AES-GCM/GMAC) and normal mode (all others) algorithms exist in the IPsec proposal, then you cannot send a single proposal to the peer. You must have at least two proposals in this case, one for combined mode and one for normal mode algorithms.

```
hostname(config)# crypto map abcmap 1 set ikev2 ipsec-proposal secure
hostname(config)#
```

# Apply Crypto Maps to Interfaces

You must apply a crypto map set to each interface through which IPsec traffic travels. The ASA supports IPsec on all interfaces. Applying the crypto map set to an interface instructs the ASA to evaluate all interface traffic against the crypto map set and to use the specified policy during connection or security association negotiations.

Binding a crypto map to an interface also initializes the runtime data structures, such as the security association database and the security policy database. When you later modify a crypto map in any way, the ASA automatically applies the changes to the running configuration. It drops any existing connections and reestablishes them after applying the new crypto map.

To apply the configured crypto map to the outside interface, perform the following steps:

**Procedure**

**Step 1** Enter the **crypto map interface** command. The syntax is **crypto map** map-name **interface** interface-name.

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

**Step 2** Save your changes.

```
hostname(config)# write memory
hostname(config)#
```

# Overview of Dynamic Site-to-Site VPNs

In a dynamic site-to-site VPN, loopback interfaces are used as the source and destination for the VPN tunnels. You can use these interfaces to move site-to-site peers between security gateways without updating the peers. When the loopback address propagates through routing protocols, ASA seamlessly redirects traffic from remote peers to the new security gateway cluster without changing configuration on the peers.

**Note** This feature does not work with an Evaluation license.

### Benefits of Using Loopback Interfaces in Dynamic Site-to-Site VPNs

- Redundancy: VPN connectivity persists during physical link or interface failures because a loopback interface remains reachable through multiple physical interfaces. In a cluster, if the node with the loopback interface fails, the interface and the sessions shift to the alternate node.
- Dynamic Loopback Address Reassignment: VPN connectivity is preserved during the reassignment of the loopback address to a different security gateway cluster without requiring changes to remote peers.
- Dynamic Path Selection: VPN connectivity is optimized as routing protocols dynamically select the best path between sites, improving both performance and reliability.

# Prerequisites for using Dynamic VPNs with Loopback Interfaces

### General Prerequisites

This feature is supported only for:

- Cisco Secure Firewall 4200 Series Version 9.24.1
- Layer 2 clustering
- Static crypto maps attached with dynamic crypto maps.

### License Prerequisites

This feature requires these licenses:

- Base license with strong encryption.
- Carrier license for distributed mode VPN.

# Configure Dynamic Site-to-Site VPN Using a Loopback Interface

### Before you begin

Ensure that you review Prerequisites for using Dynamic VPNs with Loopback Interfaces, on page 229.

### Procedure

**Step 1** Configure the outside interface using the **interface** command:

a) Configure the name of the loopback interface using the **nameif** command.
b) Configure the security level using the **security-level** command.
c) Configure the IP address of the interface using the **ip address** command.

**Example:**

```
hostname(config)# interface ethernet0/0
hostname(config-if)#nameif outside
hostname(config-if)#security-level 0
hostname(config-if)#ip address 192.0.2.17 255.255.255.0
```

**Step 2** Configure a layer 2 loopback interface using the **interface** command:

a) Configure the description using the **description** command.
b) Configure the name of the loopback interface using the **nameif** command.
c) Configure the IP address of the loopback interface using the **ip address** command.

**Example:**

```
hostname(config)# interface Loopback2
hostname(config-if)#description Loopback to terminate Group 2
hostname(config-if)#nameif LB2
hostname(config-if)#ip address 209.165.201.1 255.255.255.252
```

**Step 3** Configure an IKEv2 IPsec proposal using the **crypto ipsec ikev2 ipsec-proposal** *proposal tag* command:

a) Configure the description using the **description** command.
b) Configure the encryption protocol using the **protocol esp encryption** command.
c) Configure the encryption and integrity protocols using the **protocol esp integrity** command.

**Example:**

```
hostname(config)#crypto ipsec ikev2 ipsec-proposal ESP-AES-SHA
hostname(config-ipsec-proposal)#protocol esp encryption aes
hostname(config-ipsec-proposal)#protocol esp integrity sha-256
```

**Step 4** Configure an IKEv2 policy using the **crypto ikev2 policy** *policy_index*command:

a) Configure the encryption protocol using the **protocol esp encryption** command.
b) Configure the encryption and integrity protocols using the **protocol esp integrity** command.

c) Configure the Diffie-Hellman group using the **group** command.
d) Configure the pseudo-random function (PRF) used as the algorithm to derive keying material and hashing operations required for the IKEv2 tunnel encryption using the **prf** command.
e) Configure the encryption key lifetime using the **lifetime** command.

**Example:**

```
hostname(config)#crypto ikev2 policy 1
hostname(config-ikev2-policy)#protocol esp encryption aes-256
hostname(config-ikev2-policy)#protocol esp integrity sha
hostname(config-ikev2-policy)#group 5
hostname(config-ikev2-policy)#prf sha
hostname(config-ikev2-policy)#lifetime seconds 86400
```

**Step 5** Configure a dynamic crypto map:

a) Configure a dynamic crypto map and specify an IKEv2 transform set for the map using the **crypto dynamic-map** *dynamic-map-name dynamic-sequence-num* **set ikev2 ipsec-proposal** *transform-set-name1* command.
b) Enable reverse route injection for any connection based on this crypto map entry using the **crypto dynamic-map** *dynamic-map-name dynamic-sequence-num* **set reverse-route** command.

**Example:**

```
hostname(config)#crypto dynamic-map dmap 1 set ikev2 ipsec-proposal ESP-AES-SHA
hostname(config)#crypto dynamic-map dmap 1 set reverse-route
```

**Step 6** Configure a static crypto map:

a) Add the dynamic crypto map set into a static crypto map set using the **crypto map** *map-name sequence-num* **ipsec-isakmp dynamic** *dynamic-map-name* command.
b) Apply the static crypto maps to the loopback interface using the **crypto map** *map-name* **interface** *loopback_interface* command.

**Example:**

```
hostname(config)#crypto map vpn 1 ipsec-isakmp dynamic dmap
hostname(config)#crypto map vpn interface LB2
```

**Step 7** Configure the default LAN-to-LAN tunnel group:

a) Configure the IPsec IKEv2 attributes of the default LAN-to-LAN tunnel group using the **tunnel-group DefaultL2LGroup ipsec-attributes** command.
b) Configure the pre-shared key (PSK) for authenticating the remote peer using the **ikev2 remote-authentication pre-shared-key** *key* command.
c) Configure the pre-shared key for authenticating the local device using the **ikev2 local-authentication pre-shared-key** *key* command.

**Example:**

```
hostname(config)#tunnel-group DefaultL2LGroup ipsec-attributes
hostname(config-tunnel-ipsec)#ikev2 remote-authentication pre-shared-key ****
hostname(config-tunnel-ipsec)#ikev2 local-authentication pre-shared-key ****
```

**Step 8** Enable IKEv2 on the loopback interface using the **crypto ikev2 enable** *loopback_interface* command:

**Example:**

```
hostname(config)#crypto ikev2 enable LB2
```

**Step 9** Configure a routing protocol to advertise the loopback interface.

**Example:**

Sample configuration for OSPF:

```
hostname(config)#router ospf 1
hostname(config-router)#network 203.0.113.0 255.255.255.0 area 0
hostname(config-router)#network 209.165.201.1 255.255.255.252 area 0
hostname(config-router)#log-adj-changes
hostname(config-router)#redistribute connected

hostname(config)#interface outside
hostname(config-interface)#ospf cost 1
hostname(config-interface)#ospf message-digest-key 1 md5 *****
hostname(config-interface)#ospf authentication message-digest
```

# Verify Dynamic Site-to-Site VPN Configurations

Use the following show commands to verify the dynamic site-to-site VPN (using a loopback interface) configurations:

### show vpn-sessiondb det l2l

```
asa-node2/data-node# show vpn-sessiondb det l2l

Session Type: LAN-to-LAN Detailed

Connection   : DefaultL2LGroup
Index        : 399                    IP Addr      : <Peer-IP>
Protocol     : IKEv2 IPsec
Encryption   : IKEv2: (1)AES128  IPsec: (1)AES128
Hashing      : IKEv2: (1)SHA256  IPsec: (1)SHA256
Bytes Tx     : 58680                  Bytes Rx     : 86152
Login Time   : 09:59:41 EDT Tue Apr 8 2025
Duration     : 0h:01m:21s
Session State: Cluster Owner (backup is asa-node1)

IKEv2 Tunnels: 1
IPsec Tunnels: 1

IKEv2:
  Tunnel ID    : 399.1
  UDP Src Port : 500                    UDP Dst Port : 500
  Rem Auth Mode: preSharedKeys
  Loc Auth Mode: preSharedKeys
  Encryption   : AES128                 Hashing      : SHA256
  Rekey Int (T): 86400 Seconds          Rekey Left(T): 86319 Seconds
  PRF          : SHA256                 D/H Group    : 14
  Filter Name  : trace

IPsec:
  Tunnel ID    : 399.2
  Local Addr   : 209.165.201.1 255.255.255.0/0/0
  Remote Addr  : 192.0.2.20 255.255.255.0/0/0
  Encryption   : AES128                 Hashing      : SHA256
  Encapsulation: Tunnel
  Rekey Int (T): 28800 Seconds          Rekey Left(T): 28715 Seconds
  Idle Time Out: 30 Minutes             Idle TO Left : 29 Minutes
  Bytes Tx     : 58680                  Bytes Rx     : 86152
  Pkts Tx      : 978
```

### show crypto ikev2 sa

```
asa-node2/data-node# show crypto ikev2 sa
IKEv2 SAs:
Session-id:399, Status:UP-ACTIVE, IKE count:1, CHILD count:1
Tunnel-id Local                          Remote
fvrf/ivrf     Status         Role
   724781 209.165.201.1/500           192.0.2.20/500
Global/Global      READY    RESPONDER

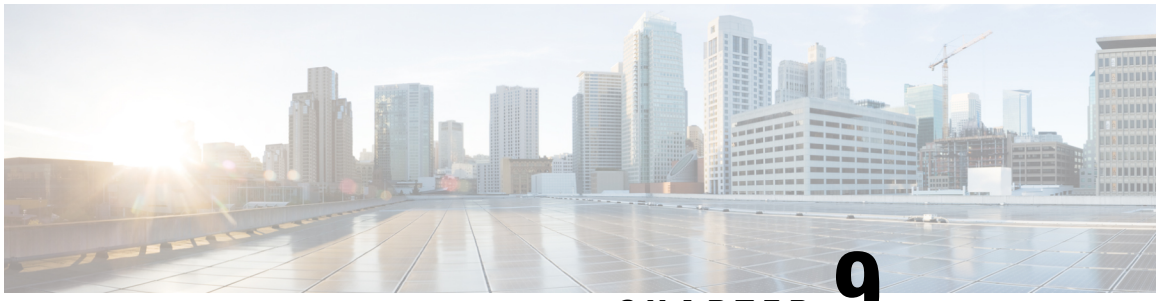
   ...
```

### show crypto ipsec sa

```
asa-node2/data-node# show crypto ipsec sa

interface: LB2
    Crypto map tag: dyn-loop1, seq num: 65535, local addr: 209.165.201.1

   ...
```



CHAPTER 9

# AnyConnect VPN Client Connections

This section describes how to configure AnyConnect VPN Client Connections.


- About the Secure Client VPN Client, on page 235
- Licensing Requirements for Secure Client, on page 236
- Configure Secure Client Connections, on page 236
- SAML 2.0, on page 255
- Monitor Secure Client Connections, on page 265
- Log Off AnyConnect VPN Sessions, on page 266
- Feature History for Secure Client Connections, on page 267


## About the Secure Client VPN Client

The Secure Client provides secure SSL and IPsec/IKEv2 connections to the ASA for remote users. Without a previously-installed client, remote users enter the IP address in their browser of an interface configured to accept SSL or IPsec/IKEv2 VPN connections. Unless the ASA is configured to redirect http:// requests to https://, users must enter the URL in the form https://<*address*>.

After entering the URL, the browser connects to that interface and displays the login screen. If the user satisfies the login and authentication, and the ASA identifies the user as requiring the client, it downloads the client that matches the operating system of the remote computer. After downloading, the client installs and configures itself, establishes a secure SSL or IPsec/IKEv2 connection and either remains or uninstalls itself (depending on the configuration) when the connection terminates.

In the case of a previously installed client, when the user authenticates, the ASA examines the revision of the client, and upgrades the client as necessary.

When the client negotiates an SSL VPN connection with the ASA, it connects using Transport Layer Security (TLS), and optionally, Datagram Transport Layer Security (DTLS). DTLS avoids latency and bandwidth problems associated with some SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

The Secure Client can be downloaded from the ASA, or it can be installed manually on the remote PC by the system administrator. For more information about installing the client manually, see the appropriate release of the Cisco AnyConnect Secure Mobility Configuration Guide .

The ASA downloads the client based on the group policy or username attributes of the user establishing the connection. You can configure the ASA to automatically download the client, or you can configure it to

prompt the remote user about whether to download the client. In the latter case, if the user does not respond, you can configure the ASA to either download the client after a timeout period or present the login page.

### Requirements for Secure Client

For the requirements of endpoint computers running the Secure Client, see the appropriate release of the Cisco AnyConnect Secure Mobility Release Notes.

### Guidelines and Limitations for Secure Client

- The ASA does not verify remote HTTPS certificates.
- Supported in single or multiple context mode. AnyConnect Apex license is required for remote-access VPN in multi-context mode. Although ASA does not specifically recognize an AnyConnect Apex license, it enforces licenses characteristics of an Apex license such as AnyConnect Premium licensed to the platform limit, Secure Client for mobile, Secure Client for Cisco VPN phone, and advanced endpoint assessment. Shared licensing, AnyConnect Essentials, failover license aggregation, and flex/time-based licenses are not supported.
- Issuing commands such as **curl** against the RA VPN headend is not directly supported, and might not have desirable results. For example, the headend does not respond to HTTP HEAD requests.
- When hardware VPN phones such as the Cisco 88xx series use Secure Client, they can experience a reconnection despite having DTLS up and Dead Peer Detection (DPD) configured.
- When a client connects to Secure Client, the IP address of the client before and after the connection changes. ASA supports this behavior.

# Licensing Requirements for Secure Client

**Note** This feature is not available on No Payload Encryption models.

VPN Licenses require an AnyConnect Plus or Apex license, available separately. See Cisco ASA Series Feature Licenses for maximum values per model.

If you start a clientless SSL VPN session and then start the Secure Client session from the portal, 1 session is used in total. However, if you start the Secure Client first (from a standalone client, for example) and then log into the clientless SSL VPN portal, then 2 sessions are used.

# Configure Secure Client Connections

This section describes prerequisites, restrictions, and detailed tasks to configure the ASA to accept AnyConnect VPN client connections.

## Configure the ASA to Web-Deploy the Client

The section describes the steps to configure the ASA to web-deploy the Secure Client.

### Before you begin

Copy the client image package to the ASA using TFTP or another method.



**Note** Even though the clientless VPN feature is disabled on ASA, when you use a web browser to access AnyConnect webdeploy (https://x.x.x.x<ASA IP address>, the VPN session on the ASA is counted as clientless.

## Procedure

**Step 1** Identify a file on flash as the Secure Client package file.

The ASA expands the file in cache memory for downloading to remote PCs. If you have multiple clients, assign an order to the client images with the order argument.

The ASA downloads portions of each client in the order you specify until it matches the operating system of the remote PC. Therefore, assign the lowest number to the image used by the most commonly-encountered operating system.

**anyconnect image** *filename order*

**Example:**

```
hostname(config-webvpn)# anyconnect image
anyconnect-win-2.3.0254-k9.pkg 1
hostname(config-webvpn)# anyconnect image
anyconnect-macosx-i386-2.3.0254-k9.pkg 2
hostname(config-webvpn)# anyconnect image
anyconnect-linux-2.3.0254-k9.pkg 3
```

**Note**
You must issue the **anyconnect enable** command after configuring the Secure Client images with the **anyconnect image** command. If you do not enable Secure Client, it will not operate as expected, and **show webvpn anyconnect** considers the SSL VPN client as not enabled rather than listing the installed Secure Client packages.

**Step 2** Enable SSL on an interface for clientless or Secure Client SSL connections.

**enable** *interface*

**Example:**

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

**Step 3** Without issuing this command, Secure Client does not function as expected, and a **show webvpn anyconnect** command returns that the "SSL VPN is not enabled," instead of listing the installed Secure Client packages.

**anyconnect enable**

**Step 4** (Optional) Create an address pool. You can use another method of address assignment, such as DHCP and/or user-assigned addressing.

**ip local pool** *poolname startaddr-endaddr* **mask** *mask*

**Example:**

```
hostname(config)# ip local pool vpn_users 209.165.200.225-209.165.200.254
mask 255.255.255.224
```

**Step 5** Assign an address pool to a tunnel group.

**address-pool** *poolname*

**Example:**

```
hostname(config)# tunnel-group telecommuters general-attributes
hostname(config-tunnel-general)# address-pool vpn_users
```

**Step 6** Assign a default group policy to the tunnel group.

**default-group-policy** *name*

```
hostname(config-tunnel-general)# default-group-policy sales
```

**Step 7** Enable the display of the tunnel-group list on the clientless portal and Secure Client GUI login page. The list of aliases is defined by the *group-alias name enable* command.

**group-alias** *name* **enable**

**Example:**

```
hostname(config)# tunnel-group telecommuters webvpn-attributes
hostname(config-tunnel-webvpn)# group-alias sales_department enable
```

**Step 8** Specify the Secure Clients as a permitted VPN tunneling protocol for the group or user.

**tunnel-group-list enable**

**Example:**

```
hostname(config)# webvpn
hostname(config-webvpn)# tunnel-group-list enable
```

**Step 9** Specify SSL as a permitted VPN tunneling protocol for the group or user. You can also specify additional protocols. For more information, see the vpn-tunnel-protocol command in the command reference.

**vpn-tunnel-protocol**

**Example:**

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# vpn-tunnel-protocol
```

### What to do next

For more information about assigning users to group policies, see Chapter 6, Configuring Connection Profiles, Group Policies, and Users.

# Enable Permanent Client Installation

Enabling permanent client installation disables the automatic uninstalling feature of the client. The client remains installed on the remote computer for subsequent connections, reducing the connection time for the remote user.

To enable permanent client installation for a specific group or user, use the anyconnect keep-installer command from group-policy or username webvpn modes.

The default is that permanent installation of the client is enabled. The client remains on the remote computer at the end of the session. The following example configures the existing group-policy *sales* to remove the client on the remote computer at the end of the session:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-policy)# anyconnect keep-installer installed none
```

# Configure DTLS

Datagram Transport Layer Security (DTLS) allows the Secure Client establishing an SSL VPN connection to use two simultaneous tunnels—an SSL tunnel and a DTLS tunnel. Using DTLS avoids latency and bandwidth problems associated with SSL connections and improves the performance of real-time applications that are sensitive to packet delays.

### Before you begin

See, Configure Advanced SSL Settings, on page 84 to configure DTLS on this headend, and which version of DTLS is used.

In order for DTLS to fall back to a TLS connection, Dead Peer Detection (DPD) must be enabled. If you do not enable DPD, and the DTLS connection experiences a problem, the connection terminates instead of falling back to TLS. For more information on DPD, see Configure Dead Peer Detection, on page 250.

### Procedure

**Step 1** Specify DTLS options for Secure Client VPN connections:

a) Enable SSL and DTLS on the interface in webvpn mode.

By default, DTLS is enabled when SSL VPN access is enabled on an interface.

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
```

Disable DTLS for all Secure Client users with the **enable** *interface* **tls-only** command in webvpn configuration mode.

If you disable DTLS, SSL VPN connections connect with an SSL VPN tunnel only.

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside tls-only
```

b) Configure the ports for SSL and DTLS using the **port** and **dtls port** commands.

```
hostname(config)# webvpn
hostname(config-webvpn)# enable outside
hostname(config-webvpn)# port 555
hostname(config-webvpn)# dtls port 556
```

**Step 2** Specify DTLS options for specific group policies.

a) Enable DTLS for specific groups or users with the **anyconnect ssl dtls** command in group policy webvpn or username webvpn configuration mode.

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl dtls enable
```

b) If desired, enable DTLS compression using the anyconnect dtls compression command.

```
hostname(config-group-webvpn)# anyconnect dtls compression lzs
```

# Prompt Remote Users

### Procedure

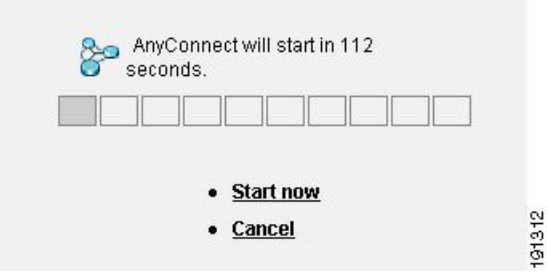
You can enable the ASA to prompt remote SSL VPN client users to download the client with the **anyconnect ask** command from group policy webvpn or username webvpn configuration modes:

[**no**] **anyconnect ask** {**none** | **enable** [**default** {**webvpn** | } **timeout** *value*]}

- **anyconnect enable** prompts the remote user to download the client or go to the clientless portal page and waits indefinitely for user response.
- **anyconnect ask enable default** immediately downloads the client.
- **anyconnect ask enable default webvpn** immediately goes to the portal page.
- **anyconnect ask enable default timeout** value prompts the remote user to download the client or go to the clientless portal page and waits the duration of *value* before taking the default action—downloading the client.
- **anyconnect ask enable default clientless timeout** value prompts the remote user to download the client or go to the clientless portal page, and waits the duration of *value* before taking the default action—displaying the clientless portal page.

The figure below shows the prompt displayed to remote users when either **default anyconnect timeout** *value or* **default webvpn timeout** *value* is configured:

*Figure 6: Prompt Displayed to Remote Users for SSL VPN Client Download*




### Example

The following example configures the ASA to prompt the user to download the client or go to the clientless portal page and wait *10 seconds for a response* before downloading the client:

```
hostname(config-group-webvpn)# anyconnect ask enable default anyconnect timeout
 10
```

# Enable Secure Client Profile Downloads

You enable Secure Client features in the Secure Client profiles—XML files that contain configuration settings for the core client with its VPN functionality and for the optional client modules. The ASA deploys the profiles during Secure Client installation and updates. Users cannot manage or modify profiles.

The file downloaded to the client is of the format: <*profile_name*>.xml.

You can configure a profile using the Secure Client profile editor, a convenient GUI-based configuration tool launched from ASDM or ISE. The Secure Client software package for Windows includes the editor, which activates when you load the client package on the chosen headend device and specify it as an Secure Client image.

We also provide a standalone version of the profile editor for Windows that you can use as an alternative to the profile editor integrated with ASDM or ISE. If you are predeploying the client, you can use the standalone profile editor to create profiles for the VPN service and other modules that you deploy to computers using your software management system.

For more information on the Secure Client and its Profile Editor, see the appropriate release of the Cisco AnyConnect Secure Mobility Configuration Guide .

**Note** The Secure Client protocol defaults to SSL. To enable IPsec IKEv2, you must configure the IKEv2 settings on the ASA and also configure IKEv2 as the primary protocol in the client profile. The IKEv2enabled profile must be deployed to the endpoint computer; otherwise the client attempts to connect using SSL.

### Procedure

**Step 1** Use the profile editor from ASDM/ISE or the standalone profile editor to create a profile.

**Step 2** Load the profile file into flash memory on the ASA using tftp or another method.

**Step 3** Use the **anyconnect profiles** command from webvpn configuration mode to identify the file as a client profile to load into cache memory.

**Example:**

The following example specifies the files sales_hosts.xml and engineering_hosts.xml as profiles:

```
asa1(config-webvpn)# anyconnect profiles sales
disk0:/sales_hosts.xml
asa1(config-webvpn)# anyconnect profiles engineering
disk0:/engineering_hosts.xml
```

The profiles are now available to group policies.

View the profiles loaded in cache memory using the **dir cache:stc/profiles** command:

```
hostname(config-webvpn)# dir cache:/stc/profiles

Directory of cache:stc/profiles/

0      ----  774         11:54:41 Nov 22 2006  engineering.xml
0      ----  774         11:54:29 Nov 22 2006  sales.xml

2428928 bytes total (18219008 bytes free)
hostname(config-webvpn)#
```

**Step 4** Enter group policy webvpn configuration mode and specify a client profile for a group policy with the **anyconnect profiles** command:

**Example:**

You can enter the client profiles value command followed by a question mark (**?**) to view the available profiles. For example:

```
asa1(config-group-webvpn)# anyconnect profiles value ?

config-group-webvpn mode commands/options:
Available configured profile packages: engineering sales
```

The next example configures the group policy to use the profile *sales* with the client profile type *vpn*:

```
asa1(config-group-webvpn)# anyconnect profiles value sales type vpn
asa1(config-group-webvpn)#
```

# Enable Secure Client Deferred Upgrade

Deferred Upgrade allows the Secure Client user to delay download of a client upgrade. When a client update is available, Secure Client opens a dialog asking the user if they would like to update, or to defer the upgrade. This upgrade dialog will not appear unless you have AutoUpdate set to *Enabled* in the Secure Client profile setting.

Deferred Upgrade is enabled by adding custom attribute types and named values to the ASA; then referencing and configuring those attributes in a group policy.

The following custom attributes support Deferred Upgrade:

***Table 13: Custom Attributes for Deferred Upgrade***

| Custom Attribute Type | Valid Values | Default Value | Notes |
|---|---|---|---|
| DeferredUpdateAllowed | true false | false | True enables deferred update. If deferred update is disabled (false), the settings below are ignored. |
| DeferredUpdateMinimumVersion | x.y.z | 0.0.0 | Minimum version of Secure Client that must be installed for updates to be deferrable.<br>The minimum version check applies to all modules enabled on the headend. If any enabled module (including VPN) is not installed or does not meet the minimum version, then the connection is not eligible for deferred update.<br>If this attribute is not specified, then a deferral prompt is displayed (or auto-dismissed) regardless of the version installed on the endpoint. |
| DeferredUpdateDismissTimeout | 0-300 (seconds) | none (disabled) | Number of seconds that the deferred upgrade prompt is displayed before being dismissed automatically. This attribute only applies when a deferred update prompt is to be displayed (the minimum version attribute is evaluated first).<br>If this attribute is missing, then the auto-dismiss feature is disabled, and a dialog is displayed (if required) until the user responds.<br>Setting this attribute to zero allows automatic deferral or upgrade to be forced based on:<br>• The installed version and the value of DeferredUpdateMinimumVersion.<br>• The value of DeferredUpdateDismissResponse. |
| DeferredUpdateDismissResponse | defer update | update | Action to take when DeferredUpdateDismissTimeout occurs. |

**Procedure**

**Step 1** Create the custom attribute types with the **anyconnnect-custom-attr** command in webvpn configuration mode:

[**no**] **anyconnect-custom-attr** *attr-type* [**description** *description* ]

**Example:**

The following example shows how to add the custom attribute types DeferredUpdateAllowed and DeferredUpdateDismissTimeout:

```
hostame(config-webvpn)# anyconnect-custom-attr DeferredUpdateAllowed
description Indicates if the deferred update feature is enabled or not
hostame(config-webvpn)# anyconnect-custom-attr DeferredUpdateDismissTimeout
```

**Step 2** Add named values for custom attributes with the **anyconnect-custom-data** command in global configuration mode. For attributes with long values, you can provide a duplicate entry, and it allows concatenation. However, with a duplicate configuration entry, the Defer Update dialog will not appear, and a user cannot defer the upgrade; instead, the upgrade happens automatically.

[**no**] **anyconnect-custom-data** *attr-type attr-name attr-value*

**Example:**

The following example shows how to add a named value for the custom attribute type DeferredUpdateDismissTimeout and for enabling DeferredUpdateAllowed:

```
hostname(config)# anyconnect-custom-data DeferredUpdateDismissTimeout
def-timeout 150
hostname(config)# anyconnect-custom-data DeferredUpdateAllowed
def-allowed true
```

**Step 3** Add or remove the custom attribute named values to a group policy using the **anyconnect-custom** command:

- **anyconnect-custom** *attr-type* **value** *attr-name*
- **anyconnect-custom** *attr-type* **none**
- **no anyconnect-custom** *attr-type*

**Example:**

The following example shows how to enable Deferred Update for the group policy named sales and set the timeout to 150 seconds:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# anyconnect-custom DeferredUpdateAllowed
value def-allowed
hostname(config-group-policy)# anyconnect-custom DeferredUpdateDismissTimeout
value def-timeout
```

# Enable DSCP Preservation

By setting another custom attribute, you can control Differentiated Services Code Point (DSCP) on Windows or OS X platforms for DTLS connections only. Enabling DSCP preservation allows devices to prioritize latency sensitive traffic; the router takes into account whether this is set and marks prioritized traffic to improve outbound connection quality.

**Procedure**

**Step 1** Create the custom attribute types with the **anyconnect-custom-attr**command in webvpn configuration mode:

**[no] anyconnect-custom-attr DSCPPreservationAllowed description Set to control Differentiated Services Code Point (DSCP) on Windows or OS X platforms for DTLS connections only.**

**Step 2** Add named values for custom attributes with the **anyconnect-custom-data** command in global configuration mode:

**[no] anyconnect-custom-data DSCPPreservationAllowed true**

**Note**
By default, Secure Client performs DSCP preservation (true). To disable it, set the custom attributes to false on the headend and reinitiate the connection.

# Enable Additional Secure Client Features

To minimize download time, the client only requests downloads (from the ASA or ISE) of the core modules that it needs. As additional features become available for the Secure Client, you need to update the remote clients in order for them to use the features.

To enable new features, you must specify the new module names using the **anyconnect modules** command from group policy webvpn or username webvpn configuration mode:

[**no**]**anyconnect modules** {**none** | **value** *string*}

Separate multiple strings with commas.

# Enable Start Before Logon

Start Before Logon (SBL) allows login scripts, password caching, drive mapping, and more, for the Secure Client installed on a Windows PC. For SBL, you must enable the ASA to download the module which enables graphical identification and authentication (GINA) for the Secure Client. The following procedure shows how to enable SBL:

**Procedure**

**Step 1** Enable the ASA to download the GINA module for VPN connection to specific groups or users using the **anyconnect modules** *vpngina* command from group policy webvpn or username webvpn configuration modes.

**Example:**

In the following example, the user enters group-policy attributes mode for the group policy *telecommuters*, enters webvpn configuration mode for the group policy, and specifies the string *vpngina*:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostame(config-group-webvpn)#anyconnect modules value vpngina
```

**Step 2** Retrieve a copy of the client profiles file (AnyConnectProfile.tmpl).

**Step 3** Edit the profiles file to specify that SBL is enabled. The example below shows the relevant portion of the profiles file (AnyConnectProfile.tmpl) for Windows:

```
<Configuration>
    <ClientInitialization>
        <UseStartBeforeLogon>false</UseStartBeforeLogon>
    </ClientInitialization>
```

The <UseStartBeforeLogon> tag determines whether the client uses SBL. To turn SBL on, replace *false* with *true*. The example below shows the tag with SBL turned on:

```
<ClientInitialization>
        <UseStartBeforeLogon>true</UseStartBeforeLogon>
    </ClientInitialization>
```

**Step 4** Save the changes to AnyConnectProfile.tmpl and update the profile file for the group or user on the ASA using the **profile** command from webvpn configuration mode. For example:

```
asa1(config-webvpn)#anyconnect profiles sales disk0:/sales_hosts.xml
```

# Translating Languages for Secure Client User Messages

The ASA provides language translation for the portal and screens displayed to users that initiate browser-based, Clientless SSL VPN connections, as well as the interface displayed to Cisco AnyConnect VPN Client users.

This section describes how to configure the ASA to translate these user messages.

## Understand Language Translation

Functional areas and their messages that are visible to remote users are organized into translation domains. All messages displayed on the user interface of the Cisco AnyConnect VPN Client are located in the Secure Client domain.

The software image package for the ASA includes a translation table template for the Secure Client domain. You can export the template, which creates an XML file of the template at the URL you provide. The message fields in this file are empty. You can edit the messages and import the template to create a new translation table object that resides in flash memory.

You can also export an existing translation table. The XML file created displays the messages you edited previously. Reimporting this XML file with the same language name creates an new version of the translation table object, overwriting previous messages. Changes to the translation table for the Secure Client domain are immediately visible to Secure Client users.

## Create Translation Tables

The following procedure describes how to create translation tables for the Secure Client domain:

### Procedure

**Step 1** Export a translation table template to a computer with the **export webvpn translation-table** command from privileged EXEC mode.

In the following example, the **show import webvpn translation-table** command shows available translation table templates and tables.

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
customization
AnyConnect

PortForwarder
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
```

Then the user exports the translation table for the Secure Client translation domain. The filename of the XML file created is named *client* and contains empty message fields:

```
hostname# export webvpn translation-table AnyConnect
template tftp://209.165.200.225/client
```

In the next example, the user exports a translation table named *zh*, which was previously imported from a template. zh is the abbreviation by Microsoft Internet Explorer for the Chinese language.

```
hostname# export webvpn translation-table customization
language zh tftp://209.165.200.225/chinese_client
```

**Step 2** Edit the Translation Table XML file. The following example shows a portion of the Secure Client template. The end of this output includes a message ID field (msgid) and a message string field (msgstr) for the message *Connected*, which is displayed on the Secure Client GUI when the client establishes a VPN connection. The complete template contains many pairs of message fields:

```
# SOME DESCRIPTIVE TITLE.
# Copyright (C) YEAR THE PACKAGE'S COPYRIGHT HOLDER
# This file is distributed under the same license as the PACKAGE package.
# FIRST AUTHOR <EMAIL@ADDRESS>, YEAR.
#
#, fuzzy
msgid ""
msgstr ""
"Project-Id-Version: PACKAGE VERSION\n"
"Report-Msgid-Bugs-To: \n"
"POT-Creation-Date: 2006-11-01 16:39-0700\n"
"PO-Revision-Date: YEAR-MO-DA HO:MI+ZONE\n"
"Last-Translator: FULL NAME <EMAIL@ADDRESS>\n"
```

```
"Language-Team: LANGUAGE <LL@li.org>\n"
"MIME-Version: 1.0\n"
"Content-Type: text/plain; charset=CHARSET\n"
"Content-Transfer-Encoding: 8bit\n"

#: C:\cygwin\home\<user>\cvc\main\Api\AgentIfc.cpp:23
#: C:\cygwin\home\<user>\cvc\main\Api\check\AgentIfc.cpp:22
#: C:\cygwin\home\<user>\cvc\main\Api\save\AgentIfc.cpp:23
#: C:\cygwin\home\<user>\cvc\main\Api\save\AgentIfc.cpp~:20
#: C:\cygwin\home\<user>\cvc\main\Api\save\older\AgentIfc.cpp:22
msgid "Connected"
msgstr ""
```

The msgid contains the default translation. The msgstr that follows msgid provides the translation. To create a translation, enter the translated text between the quotes of the msgstr string. For example, to translate the message “Connected” with a Spanish translation, insert the Spanish text between the quotes:

```
msgid "Connected"
msgstr "Conectado"
```

Be sure to save the file.

**Step 3** Import the translation table using the **import webvpn translation-table** command from privileged EXEC mode. Be sure to specify the name of the new translation table with the abbreviation for the language that is compatible with the browser.

In the following example, the XML file is imported *es-us*—the abbreviation used by Microsoft Internet Explorer for Spanish spoken in the United States.

```
hostname# import webvpn translation-table AnyConnect
language es-us tftp://209.165.200.225/client
hostname# !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
hostname# show import webvpn translation-table
Translation Tables' Templates:
AnyConnect
PortForwarder

customization
keepout
url-list
webvpn
Citrix-plugin
RPC-plugin
Telnet-SSH-plugin
VNC-plugin

Translation Tables:
es-us AnyConnect
```

## Remove Translation Tables

If you no longer need a translation table, you can remove it.

**Procedure**

**Step 1** List the existing translation tables.

In the following example, the **show import webvpn translation-table** command shows available translation table templates and tables. Various tables are available for French (fr), Japanese (ja), and Russian (ru).

```
hostname# show import webvpn translation-table
Translation Tables' Templates:
  AnyConnect
  PortForwarder
  banners
  csd
  customization
  url-list
  webvpn
Translation Tables:
  fr                    PortForwarder
  fr                    AnyConnect
  fr                    customization
  fr                    webvpn
  ja                    PortForwarder
  ja                    AnyConnect
  ja                    customization
  ja                    webvpn
  ru                    PortForwarder
  ru                    customization
  ru                    webvpn
```

**Step 2** Remove the unwanted translation table.

**revert webvpn translation-table** *translationdomain* **language** *language*

Where *translationdomain* is the domain listed on the right in the Translation Tables listing shown above, and *language* is the 2-character language name.

You must remove each table individually. You cannot remove all of the tables for a given language with one command.

For example, to remove the French translation table for Secure Client:

```
ciscoasa# revert webvpn translation-table anyconnect language fr
ciscoasa#
```

# Configuring Advanced Secure Client SSL Features

The following section describes advanced features that fine-tune Secure Client SSL VPN connections.

## Enable Rekey

When the ASA and the Secure Client perform a rekey on an SSL VPN connection, they renegotiate the crypto keys and initialization vectors, increasing the security of the connection.

To enable the client to perform a rekey on an SSL VPN connection for a specific group or user, use the **anyconnect ssl** rekey command from group-policy or username webvpn modes.

[no]anyconnect ssl rekey {**method** {**new-tunnel** | **none** | **ssl**} | **time** *minutes*}

- **method new-tunnel** specifies that the client establishes a new tunnel during rekey.
- **method ssl** specifies that the client establishes a new tunnel during rekey.
- **method none** disables rekey.
- **time** *minutes* specifies the number of minutes from the start of the session, or from the last rekey, until the rekey takes place, from 1 to 10080 (1 week).

**Note** Configuring the rekey method as **ssl** or **new-tunnel** specifies that the client establishes a new tunnel during rekey instead of the SSL renegotiation taking place during the rekey. See the command reference for a history of the **anyconnect ssl rekey** command.

In the following example, the client is configured to renegotiate with SSL during rekey, which takes place 30 minutes after the session begins, for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl rekey method ssl
hostname(config-group-webvpn)# anyconnect ssl rekey time 30
```

## Configure Dead Peer Detection

Dead Peer Detection (DPD) ensures that the ASA (gateway) or the client can quickly detect a condition where the peer is not responding, and the connection has failed. To enable dead peer detection (DPD) and set the frequency with which either the Secure Client or the ASA gateway performs DPD.

**Note** When the connection is interrupted on the client-side, ASA does not drop the Secure Client sesssion because of DPD or keepalive unconditionally. DPD is only triggered by ASA when some data flow is there from ASA to client. Once DPD is triggered, it makes three retries for each child session (SSL/DTLS) before tearing them down.

If data flow is not there, DPD is not triggered, ASA has a hardcoded 5-minute TCP inactivity timeout that automatically closes SSL/DTLS tunnel connections when no data or keepalive packets flow for exactly 5 minutes, regardless of the configured VPN idle timeout setting. After the child sessions are teared down, **vpn-idle-timeout** command is only responsible for controlling the maximum time for a parent session. For further details on DPD, keepalives, and timeout attributes, see Answer AnyConnect FAQ - Tunnels, DPDs, and Inactivity Timer.

### Before you begin

- This feature applies to connectivity between the ASA gateway and the Secure Client SSL VPN Client only. It does not work with IPsec since DPD is based on the standards implementation that does not allow padding.

- If you enable DTLS, enable Dead Peer Detection (DPD) also. DPD enables a failed DTLS connection to fallback to TLS. Otherwise, the connection terminates.
- When DPD is enabled on the ASA, you can use the Optimal MTU (OMTU) function to find the largest endpoint MTU at which the client can successfully pass DTLS packets. Implement OMTU by sending a padded DPD packet to the maximum MTU. If a correct echo of the payload is received from the head end, the MTU size is accepted. Otherwise, the MTU is reduced, and the probe is sent again until the minimum MTU allowed for the protocol is reached.

### Procedure

**Step 1** Go to the desired group policy.

Enter group policy or username webvpn mode:

```
hostname(config)# group-policy group-policy-name  attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)#
```

Or,

```
hostname# username username attributes
hostname(config-username)# webvpn
hostname (config-username-webvpn #
```

**Step 2** Set Gateway Side Detection.

Use the [**no**] **anyconnect dpd-interval** {[**gateway** {*seconds* | **none**}] command.

The gateway refers to the ASA. You enable DPD and specify the interval with which the ASA waits for any packet from the client as a range of from 30 (default) to 3600 seconds (1 hour). A value of 300 is recommended. If no packets are received within that interval, the ASA performs the DPD test with three attempts at the same interval. If the ASA does not receive a response from the client, it tears down the TLS/DTLS tunnel.

**Note**
Specifying **none** disables the DPD testing that the ASA performs. Use **no anyconnect dpd-interval** to remove this command from the configuration.

Specifying **none** disables the DPD testing that the ASA performs. Use **no anyconnect dpd-interval** to remove this command from the configuration.

**Step 3** Set Client Side Detection.

Use the [**no**] **anyconnect dpd-interval** {[**client** {*seconds* | **none**}]} command.

The client refers to the Secure Client. You enable DPD and specify the frequency with which the client performs the DPD test as a range of from 30 (default) to 3600 seconds (1 hour). A value of 30 seconds is recommended.

Specifying **client none** disables DPD performed by the client. Use **no anyconnect dpd-interval** to remove this command from the configuration.

### Example

The following example sets the frequency of DPD performed by the ASA to 30 seconds, and the frequency of DPD performed by the client set to 10 seconds for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect dpd-interval gateway 30
hostname(config-group-webvpn)# anyconnect dpd-interval client 10
```

## Enable Keepalive

You can adjust the frequency of keepalive messages to ensure that an SSL VPN connection through a proxy, firewall, or NAT device remains open, even if the device limits the time that the connection can be idle. Adjusting the frequency also ensures that the client does not disconnect and reconnect when the remote user is not actively running a socket-based application, such as Microsoft Outlook or Microsoft Internet Explorer.

Keepalives are enabled by default. If you disable keepalives, in the event of a failover, SSL VPN client sessions are not carried over to the standby device.

To set the frequency of keepalive messages, use the **keepalive** command from group-policy webvpn or username webvpn configuration mode: Use the **no** form of the command to remove the command from the configuration and cause the value to be inherited:

[**no**] **anyconnect ssl keepalive** {**none** | *seconds*}

- **none** disables client keepalive messages.
- *seconds* enables the client to send keepalive messages, and specifies the frequency of the messages in the range of 15 to 600 seconds.

In the following example, the ASA is configured to enable the client to send keepalive messages with a frequency of 300 seconds (5 minutes), for the existing group-policy *sales*:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect ssl keepalive 300
```

## Use Compression

Compression increases the communications performance between the ASA and the client by reducing the size of the packets being transferred for low-bandwidth connections. By default, compression for all SSL VPN connections is enabled on the ASA, both at the global level and for specific groups or users.



**Note** When implementing compression on broadband connections, you must carefully consider the fact that compression relies on loss-less connectivity. This is the main reason that it is not enabled by default on broadband connections.

Compression must be turned-on globally using the **compression** command from global configuration mode, and then it can be set for specific groups or users with the **anyconnect ssl compression** command in group-policy and username webvpn modes.

### Changing Compression Globally

To change the global compression settings, use the anyconnect ssl **compression** command from global configuration mode. To remove the command from the configuration, use the **no** form of the command.

In the following example, compression is disabled for all SSL VPN connections globally:

```
hostname(config)# no compression
```

### Changing Compression for Groups and Users

To change compression for a specific group or user, use the anyconnect ssl compression command in the group-policy and username webvpn modes:

[**no**] **anyconnect ssl compression** {**deflate** | **none**}

By default, for groups and users, SSL compression is set to *deflate* (enabled).

To remove the **anyconnect ssl compression** command from the configuration and cause the value to be inherited from the global setting, use the **no** form of the command:

In the following example, compression is disabled for the group-policy sales:

```
hostname(config)# group-policy sales attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# no anyconnect ssl compression none
```

## Adjust MTU Size

You can adjust the MTU size (from 576 to 1406 bytes) for SSL VPN connections established by the client with the **anyconnect mtu** command from group policy webvpn or username webvpn configuration mode:

[**no**] **anyconnect mtu** *size*

This command affects only the Secure Client. The legacy Cisco SSL VPN Client () is not capable of adjusting to different MTU sizes. Also, client connections established in SSL and those established in SSL with DTLS are impacted by this command.

The default for this command in the default group policy is **no anyconnect mtu**. The MTU size is adjusted automatically based on the MTU of the interface that the connection uses, minus the IP/UDP/DTLS overhead.

You may receive an "MTU configuration sent from the secure gateway is too small" message, for example, when running the ISE Posture AnyConnect module. If you enter **anyconnect mtu 1200** along with **anyconnect ssl df-bit-ignore disable**, you can avoid these system scan errors.

### Example

The following example configures the MTU size to 1200 bytes for the group policy telecommuters:

```
hostname(config)# group-policy telecommuters attributes
hostname(config-group-policy)# webvpn
hostname(config-group-webvpn)# anyconnect mtu 1200
```

# Update Secure Client Images

You can update the client images on the ASA at any time using this procedure.

**Note** For optimal security, performance, and manageability of your VPN infrastructure, we recommend that you regularly remove outdated Secure Client images from your firewall, keeping only the latest required versions to prevent configuration conflicts.

**Procedure**

**Step 1** Copy the new client images to the ASA using the **copy** command from privileged EXEC mode, or using another method.

**Step 2** If the new client image files have the same filenames as the files already loaded, reenter the **anyconnect image** command that is in the configuration. If the new filenames are different, uninstall the old files using the [**no**]**anyconnect image**image command. Then use the **anyconnect image** command to assign an order to the images and cause the ASA to load the new images.

# Enable IPv6 VPN Access

If you want to configure IPv6 access, you must use the command-line interface. Release 9.0(x) of the ASA adds support for IPv6 VPN connections to its outside interface using SSL and IKEv2/IPsec protocols.

You enable IPv6 access using the **ipv6 enable** command as part of enabling SSL VPN connections. The following is an example for an IPv6 connection that enables IPv6 on the outside interface:

```
hostname(config)# interface GigabitEthernet0/0
hostname(config-if)# ipv6 enable
```

To enable IPV6 SSL VPN, do the following general actions:

1. Enable IPv6 on the outside interface.
2. Enable IPv6 and an IPv6 address on the inside interface.
3. Configure an IPv6 address local pool for client assigned IP Addresses.
4. Configure an IPv6 tunnel default gateway.

**Procedure**

**Step 1** Configure Interfaces:

```
interface GigabitEthernet0/0
 nameif outside
 security-level 0
 ip address 192.168.0.1 255.255.255.0
 ipv6 enable      ; Needed for IPv6.
!
interface GigabitEthernet0/1
 nameif inside
```

```
security-level 100
ip address 10.10.0.1 255.255.0.0
ipv6 address 2001:DB8::1/32        ; Needed for IPv6.
ipv6 enable      ; Needed for IPv6.
```

**Step 2** Configure an 'ipv6 local pool' (used for IPv6 address assignment):

```
ipv6 local pool ipv6pool 2001:DB8:1:1::5/32 100     ; Use your IPv6 prefix here
```

**Note**
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.

**Step 3** Add the ipv6 address pool to your tunnel group policy (or group-policy):

```
tunnel-group YourTunGrp1 general-attributes  ipv6-address-pool ipv6pool
```

**Note**
You must also configure an IPv4 address pool here as well (using the 'address-pool' command).

**Step 4** Configure an IPv6 tunnel default gateway:

```
ipv6 route inside ::/0 X:X:X:X::X tunneled
```

# SAML 2.0

The ASA supports SAML 2.0 so that the VPN end users will be able to input their credentials only one time when they switch between SAAS applications outside of the private network.

For instance, an enterprise customer has enabled PingIdentity as their SAML Identity Provider (IdP) and has accounts on Rally, Salesforce, Oracle OEM, Microsoft ADFS, onelogin, or Dropbox which have been SAML 2.0 SSO enabled. When you configure the ASA to support SAML 2.0 SSO as a Service Provider (SP), end users are able to sign in once and have access to all these services.

AnyConnect SAML support was added to allow an AnyConnect 4.4 client to access SAAS-based applications using SAML 2.0. AnyConnect 4.6 introduced an enhanced version of SAML integration with an embedded browser which replaced the native (external) browser integration from previous releases. The new enhanced version with embedded browser required you to upgrade to AnyConnect 4.6 (or later) and ASA 9.7.1.24 (or later), 9.8.2.28 (or later), or 9.9.2.1 (or later).

ASA release 9.17.1/ASDM release 7.17.1 introduced support for AnyConnect VPN SAML external browser with AnyConnect 4.10.04065 (or later). When you use SAML as the primary authentication method for the AnyConnect VPN connection profile, you can choose for the Secure Client to use a local browser, instead of the Secure Client embedded browser, when performing web authentication. With this feature, Secure Client supports WebAuthN and any other SAML-based web authentication options, such as Single Sign On, biometric authentication, or other enhanced methods that are unavailable with the embedded browser. For SAML external browser use, you must perform the configuration described here: Configure Default OS Browser for SAML Authentication, on page 262.

The ASA is SP enabled when SAML is configured as the authentication method for a tunnel group, the default tunnel group or any other. The VPN user initiates Single sign-on by accessing an enabled ASA or the SAML IdP. Each of these scenarios is described below.

### SAML SP-initiated SSO

When the end user initiates login by accessing the ASA, sign-on behavior proceeds as follows:

1. When the VPN user accesses or chooses a SAML enabled tunnel group, the end user will be redirected to the SAML IdP for authentication. The user will be prompted unless the user access the group-url directly, in which case the redirect is silent.

   The ASA generates a SAML Authentication Request, which the browser redirects to the SAML IdP.

2. The IdP challenges the end user for credential and the end user logs in. The entered credentials must satisfy the IdP authentication configuration.

3. The IdP Response is sent back to the browser and posted to the ASA's sign-in URL. The ASA verifies the response to complete the login.

### SAML IdP-initiated SSL

When the user initiates login by accessing the IdP, sign-on behavior proceeds as follows:

1. An end user accesses the IdP. The IdP challenges the end user for credentials according to the IdP's authentication configuration. The end user submits credentials and logs in to the IdP.

2. In general, the end user gets a list of SAML enabled services that have been configured with the IdP. The end user chooses the ASA.

3. A SAML response is sent back to the browser, and posted to the ASA sign-in URL. The ASA verifies the response to complete the login.

### Circle of Trust

The trust relationship between the ASA and the SAML Identity Provider is established through configured certificates (ASA trustpoints).

The trust relationship between the end user and SAML Identity Provider is established through the authentication configured on IdP.

### SAML Timeouts

In SAML assertion, there are NotBefore and NotOnOrAfter as follows: <saml:Conditions NotBefore="2015-03-10T19:47:41Z" NotOnOrAfter="2015-03-10T20:47:41Z">

A SAML timeout configured on the ASA will override NotOnOrAfter if the sum of NotBefore and timeout is earlier than NotOnOrAfter. If NotBefore + timeout is later than NotOnOrAfter, then NotOnOrAfter will take effect.

The timeout should be very short to prevent the assertion from being re-used after the timeout. You must synchronize your ASA's Network Time Protocol (NTP) server with the IdP NTP server in order to use the SAML feature.

### Support in Private Network

SAML 2.0-based service provider IdP is supported in a private network. When the SAML IdP is deployed in the private cloud, ASA and other SAML-enabled services are in peer positions, and all in the private network. With the ASA as a gateway between the user and services, authentication on IdP is handled with a restricted anonymous webvpn session, and all traffic between IdP and the user is translated. When the user logs in, the ASA modifies the session with the corresponding attributes and stores the IdP sessions. Then you can use service provider on the private network without entering credentials again.

The SAML IdP *NameID* attribute determines the user's username and is used for authorization, accounting, and VPN session database.



**Note** You cannot exchange authentication information between private and public networks. If you use the same IdP for both internal and external service providers, you must authenticate separately. Internal-only IdP cannot be used with external services: external-only IdP cannot be used with service providers in the private network.

# Guidelines and Limitations for SAML 2.0

- ASA supports the following signatures for SAML authentication:
  - SHA1 with RSA and HMAC
  - SHA2 with RSA and HMAC
- ASA supports SAML 2.0 Redirect-POST binding , which is supported by all SAML IdPs.
- The ASA functions as a SAML SP only. It cannot act as an Identity Provider in gateway mode or peer mode.
- This SAML SSO SP feature is a mutual exclusion authentication method. It cannot be used with AAA and certificate together.
- Features that are based on username/password authentication, certificate authentication, and KCD are not supported. For instance, username/password pre-filling feature, form-based Auto sign-on, Macro Substitution based Auto sign-on, KCD SSO, and so on.
- ASA supports VPN load balancing with AnyConnect SAML authentication.
- While using Safari for SAML authentication, ensure that you have Safari update 14.1.2 or higher.
- ASA administrators need to ensure clock synchronization between the ASA and the SAML IdP for proper handling of authentication assertions and proper timeout behavior.
- ASA administrators have the responsibility to maintain a valid signing certificate on both ASA and IdP considering the following:
  - The IdP signing certificate is mandatory when configuring an IdP on the ASA.
  - The ASA does not do a revocation check on the signing certificate received from the IdP.
- In SAML assertions, there are NotBefore and NotOnOrAfter conditions. The ASA SAML configured **timeout** interacts with these conditions as follows:
  - Timeout overrides NotOnOrAfter if the sum of NotBefore and timeout is earlier than NotOnOrAfter.

  - If NotBefore + timeout is later than NotOnOrAfter, then NotOnOrAfter takes effect.
  - If the NotBefore attribute is absent, the ASA denies the login request. If the NotOnOrAfter attribute is absent and SAML timeout is not set, ASA denies the login request.

- ASA does not work with Duo in a deployment using an internal SAML, which forces the ASA to proxy for the client to authenticate, due to the FQDN change that occurs during challenge/response for Two-factor authentication (push, code, password).
- Untrusted server certificates are not allowed in the embedded browser.
- The embedded browser SAML integration is not supported in CLI or SBL modes.
- SAML authentication established in a web browser is not shared with AnyConnect and vice versa.
- Depending on the configuration, various methods are used when connecting to the headend with the embedded browser. For example, while AnyConnect might prefer an IPv4 connection over an IPv6 connection, the embedded browser might prefer IPv6, or vice versa. Similarly, AnyConnect may fall back to no proxy after trying proxy and getting a failure, while the embedded browser may stop navigation after trying proxy and getting a failure.
- You must synchronize your ASA's Network Time Protocol (NTP) server with the IdP NTP server in order to use the SAML feature.
- The VPN Wizard on ASDM does not currently support SAML configurations.
- You cannot access internal servers with SSO after logging in using an internal IdP.
- The SAML IdP NameID attribute determines the user's username and is used for authorization, accounting, and VPN session database.
- SAML is not supported in the Multicontext mode.
- Multiple attributes received with a SAML assertion are not supported.
- Chromebooks does not support Secure Client SAML with external browser authentication.
- Ensure that the IdP includes the Relay state parameter in the SAML response as received in the corresponding SAML request.

# Configure a SAML 2.0 Identity Provider (IdP)

### Before you begin

Get the Sign-in and Sign-out URLs for your SAML (IdP) provider. You can get the URLs from the provider's website, or they may provide that information in a metadata file.

### Procedure

**Step 1** Create a SAML identity provider in webvpn config mode and enter saml-idp sub-mode under webvpn.

[**no**] **saml idp** *idp-entityID*

*idp-entityID*— The SAML IdP entityID must contain 4 to 128 characters.

To remove a SAML IdP, use the **no** form of this command.

**Step 2** Configure the IdP URLs.

**url** [**sign-in** | **sign-out**] *value*

*value* —This is the URL for signing into the IdP or the URL for redirecting to when signing out of the IdP. The **sign-in** URL is required, the **sign-out** URL is optional The url value must contain 4 to 500 characters.

**Step 3** (Optional) Configures the base URL of the SAML service provider for VPN authentication. This URL is used in SAML metadata, which is provided to third-party IdPs, so that IdPs can redirect endpoint users back to the ASA.

**base-url** *URL*

This URL is provided to third-party IdPs to redirect end users back to the ASA.

When base-url is configured, we use it as the base URL of the AssertionConsumerService and SingleLogoutService attribute in **show saml metadata**.

When base-url is not configured, the URL is determined by the ASA's hostname and domain-name. For example, we use `https://ssl-vpn.cisco.com` when hostname is `ssl-vpn` and domain-name is `cisco.com`.

An error occurs if neither base-url nor the hostname/domain-name are configured when entering **show saml metadata**.

**Step 4** Configure trustpoints between the IdP and SP (ASA).

**trustpoint idp** *trustpoint-name1* [*trustpoint-name2*]

**trustpoint sp** *trustpoint-name1*

**idp**—Specifies the trustpoint that contains the IdP certificate for the ASA to verify SAML assertions.

You can now configure two trustpoints for a SAML IdP on a device. This feature allows you to gracefully transition to a new Identity Provider (IdP) certificate without any loss of service. You do not need to open a maintenance window to simultaneously update all ASAs and the IdP with the same certificate. When the new IdP certificate is enabled on the IdP, the device automatically detects the new certificate. You can safely delete the original trustpoint after the transition.

**sp** —Specifies the trustpoint that contains the ASA (SP)'s certificate for the IdP to verify ASA's signature or encrypted SAML assertion.

*trustpoint-name*—Must be a previously configured trustpoint.

**Step 5** (Optional) Configure SAML timeout.

**timeout assertion** *timeout-in-seconds*

If specified, this configuration overrides NotOnOrAfter if the sum of NotBefore and timeout-in-seconds is earlier than NotOnOrAfter.

If not specified, NotBefore and NotOnOrAfter in the assertion is used to determine the validity.

**Note**
For a tunnel group with existing SAML IdP configured, any changes to the saml idp CLI under webvpn are only applied to the tunnel group when SAML is re-enabled for that particular tunnel group. After you configure the timeout, the updated timeout takes effect only after re-issuing the saml identity-provider CLI in the tunnel group webvpn-attributes.

**Step 6** (Optional) Enable or disable (default setting) the signature in SAML request.

**signature <value>**

**Note**
With the upgrade to SSO 2.5.1, the default signing method changes from SHA1 to SHA256, and you can configure which signing method option you prefer by entering the *value* rsa-sha1, rsa-sha256, rsa-sha384, or rsa-sha512.

**Step 7** (Optional) To set the flag determining that the IdP is an internal network, use the **internal** command. The ASA will then work in a gateway mode.

**Step 8** Use **show webvpn saml idp** to view the configuration.

**Step 9** Use **force re-authentication** to cause the identity provider to authenticate directly rather than rely on a previous security context when a SAML authentication request occurs. This setting is the default; therefore, to disable, use **no force re-authentication**.

### Example

The following example configures an IdP named salesforce_idp and uses preconfigured trustpoints:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)#saml idp salesforce_idp

ciscoasa(config-webvpn-saml-idp)#url sign-in
https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
ciscoasa(config-webvpn-saml-idp)#url sign-out
https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect

ciscoasa(config-webvpn-saml-idp)#trustpoint idp salesforce_trustpoint salesforce_trustpoint2
ciscoasa(config-webvpn-saml-idp)#trustpoint sp asa_trustpoint

ciscoasa(config)#show webvpn saml idp
saml idp salesforce_idp
url sign-in https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
url sign-out https://asa-dev-ed.my.salesforce.com/idp/endpoint/HttpRedirect
trustpoint idp salesforce_trustpoint salesforce_trustpoint2
trustpoint  sp asa_trustpoint
```

The following web page shows an example of how to get URLs for Onelogin,

https://onelogin.zendesk.com/hc/en-us/articles/202767260-Configuring-SAML-for-Clarizen

The following web page is an example of how to use metadata to find the URLs from OneLogin.

http://onlinehelp.tableau.com/current/online/en-us/saml_config_onelogin.htm

### What to do next

Apply SAML authentication to connection profiles, as described in Configure ASA as a SAML 2.0 Service Provider (SP), on page 261.

# Configure ASA as a SAML 2.0 Service Provider (SP)

### Before you begin

The IdP must have been previously configured. See Configure a SAML 2.0 Identity Provider (IdP), on page 258.

### Procedure

**Step 1** In tunnel-group webvpn sub-mode, use the saml identity-provider command to assign an IdP.

**saml identity-provider** *idp-entityID*

*idp-entityID*—Must be one of the existing IdPs previously configured.

To disable SAML SP, use the **no** form of this command.

**Step 2** Enable the SAML authentication method.

**authentication saml**

**Step 3** (Optional) Configure the SAML IdP trustpoint for the tunnel group.

**saml idp-trustpoint** *trustpoint-name* [*trustpoint-name2*]

If your IdP, for example Azure IdP, has multiple applications but shares the same SAML entity ID, and each application has its own certificate. You can use the above command associated with a given application to override the main webvpn saml configuration.

### Example

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# tunnel-group-list enable
ciscoasa(config)# tunnel-group cloud_idp_onelogin type remote-access
ciscoasa(config)# tunnel-group cloud_idp_onelogin webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# group-alias cloud_idp enable
ciscoasa(config-tunnel-webvpn)# saml identity-provider
https://app.onelogin.com/saml/metadata/462950
```

Configure an IdP trustpoint for a tunnel group:

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# tunnel-group-list enable
ciscoasa(config)# tunnel-group partner-saml type remote-access
ciscoasa(config)# tunnel-group partner-saml webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# saml identity-provider https://sts.windows.net/123-3456-7890/
ciscoasa(config-tunnel-webvpn)# saml idp-trustpoint azure-partner-idp-tp
```

# Configure Default OS Browser for SAML Authentication

Specify whether or not AnyConnect should handle the SSO authentication process using the platform's native browser (the operating system's default browser) or using the browser that is embedded in AnyConnect.

You must download the AnyConnect external browser package (Example, *external-sso-4.10.04065-webdeploy-k9.pkg*) and upload it to ASA. You can then choose the SAML login method (AnyConnect's embedded browser or the operating system's default browser) for SAML authentication. This bundle is a script that allows the VPN client to launch the default OS web browser for authentication purposes, and is independent of the operating system, browser, and VPN Client version. As long as the feature is enabled, the VPN client version and the external browser package version file do not need to match.

Choosing the default operating system browser enables single sign-on (SSO) between your VPN authentication and other corporate logins. Choose this option if you want to support web authentication methods, such as biometric authentication, that cannot be performed in the VPN client's embedded browser. Before selecting the operating system's browser, you must upload a package that can be run in the browser to enable web authentication.

### Procedure

**Step 1** In webvpn sub-mode, use the anyconnect external-browser-pkg command to enable AnyConnect SAML authentication through the operating system's default browser.

**anyconnect external-browser-pkg** *path*

To disable the operating system's default browser for SAML authentication, use the **no** form of this command.

**Step 2** In tunnel-group webvpn sub-mode, use the external-browser command to enable AnyConnect SAML authentication through the operating system's default browser.

**external-browser enable** *idp-entityID*

To disable the operating system's default browser for SAML authentication, use the **no** form of this command.

### Example

This example selects the path for the AnyConnect external browser package and enables an external browser (the operating system's default browser) for SAML authentication.

```
asa(config-webvpn)# anyconnect external-browser-pkg flashshow :
asa(config)# tunnel-group SAML webvpn-attributes
asa(config-tunnel-webvpn)# external-browser enable
asa(config-tunnel-webvpn)#
```

# Configure Certificate and SAML Authentication

You can configure certificate and SAML authentication for SAML-based connection profiles to validate customer owned assets without profiling for a particular file/registry key. SAML based authentications can

be tied to sanctioned assets and/or users. You can use a single certificate or multiple certificates with SAML for authentication.

When the Secure Client initiates a connection, ASA or FTD will request and authenticate one or more certificates from the endpoint before SAML authentication is performed.

Once SAML authentication is complete, the SAML and certificate username can be

Once SAML authentication is complete, the SAML and certificate username can be compared before proceeding to the authorization phase.

### Before you begin

Ensure that you configure required SAML settings before configuration Certificate and SAML authentication:

- Get the Sign-in and Sign-out URLs for your SAML (IdP) provider. You can get the URLs from the provider's website, or they may provide that information in a metadata file.
- Configure SAML identity provider and trustpoint settings. See Configure Certificate and SAML Authentication, on page 262

## Procedure

**Step 1** To configure certificate and SAML authentication, enter tunnel-group webvpn-attributes mode by entering the following command. The prompt changes to indicate the mode change:

```
hostname(config)# tunnel-group tunnel-group-name webvpn-attributes
hostname(config-tunnel-webvpn)#
```

**Step 2** To specify the authentication method to use, enter the following command:

```
hostname(config-tunnel-webvpn)#authentication authentication_method
```

For example, The following command allows both SAML and certificate authentication:

```
hostname(config-tunnel-webvpn)#authentication saml certificate
```

The following command allows certificate and SAML authentication:

```
hostname(config-tunnel-webvpn)#authentication certificate saml
```

The following command allows both multiple certificate and SAML authentication:

```
hostname(config-tunnel-webvpn)#authentication multiple-certificate saml
```

**Step 3** Add or edit a connection profile and then select **Basic** connection profile attribute settings.

**Step 4** To specify the authentication method for certificate and SAML authentication, select SAML and certificate or Multiple certificates and SAML from the drop-down.

### Example

The following example configures multiple certificates and SAML authentication for the sales_group connection profile:

```
ciscoasa(config)# tunnel-group sales_group webvpn
ciscoasa(config-tunnel-webvpn)#authentication multiple-certificate saml
```

# Example SAML 2.0 and Onelogin

Follow this example using your third party SAML 2.0 IdP in place of the Onelogin information and naming.

1. Set time synchronization between the IdP and the ASA(SP).

   ```
   ciscoasa(config)# ntp server 209.244.0.4
   ```

2. Obtain the IdP's SAML metadata from the IdP following procedures provided by your third party IdP.

3. Import the IdP's signing certificate into a trustpoint.

   ```
   ciscoasa(config)# crypto ca trustpoint onelogin
   ciscoasa(config-ca-trustpoint)# enrollment terminal
   ciscoasa(config-ca-trustpoint)# no ca-check
   ciscoasa(config-ca-trustpoint)# crypto ca authenticate onelogin
   Enter the base 64 encoded CA certificate.
   End with the word "quit" on a line by itself
   quit
   INFO: Certificate has the following attributes:
   Fingerprint:     85de3781 07388f5b d92d9d14 1e22a549
   Do you accept this certificate? [yes/no]: yes
   Trustpoint CA certificate accepted.
   % Certificate successfully imported
   ```

4. Import the SP (ASA) signing PKCS12 into a trustpoint

   ```
   ciscoasa(config)# crypto ca import asa_saml_sp pkcs12 password
   Enter the base 64 encoded pkcs12.
   End with the word "quit" on a line by itself:
   quit
   INFO: Import PKCS12 operation completed successfully
   ```

5. Add a SAML IdP:

   ```
   ciscoasa(config-webvpn)# saml idp https://app.onelogin.com/saml/metadata/462950
   ```

6. Configure attributes under saml-idp sub-mode:

   Configure the IdP sign-in URL and sign-ou URL:

   ```
   ciscoasa(config-webvpn-saml-idp)# url sign-in
   https://ross.onelogin.com/trust/saml2/http-post/sso/462950
   ciscoasa(config-webvpn-saml-idp)# url sign-out
   https://ross.onelogin.com/trust/saml2/http-redirect/slo/462950
   ```

   Configure the IdP trustpoint and the SP trustpoint

   ```
   ciscoasa(config-webvpn-saml-idp)# trustpoint idp onelogin
   ciscoasa(config-webvpn-saml-idp)# trustpoint sp asa_saml_sp
   ```

   Configure the Clientless VPN base URL, SAML request signature and SAML assertion timout:

```
ciscoasa(config-webvpn-saml-idp)# base-url https://172.23.34.222
ciscoasa(config-webvpn-saml-idp)# signature
ciscoasa(config-webvpn-saml-idp)# timeout assertion 7200
```

7. Configure an IdP for a tunnel group and enable SAML authentication.

```
ciscoasa(config)# webvpn
ciscoasa(config-webvpn)# tunnel-group-list enable
ciscoasa(config)# tunnel-group cloud_idp_onelogin type remote-access
ciscoasa(config)# tunnel-group cloud_idp_onelogin webvpn-attributes
ciscoasa(config-tunnel-webvpn)# authentication saml
ciscoasa(config-tunnel-webvpn)# group-alias cloud_idp enable
ciscoasa(config-tunnel-webvpn)# saml identity-provider
https://app.onelogin.com/saml/metadata/462950
```

8. Show the ASA's SAML SP metadata:

   You can get the ASA's SAML SP metadata from `https://172.23.34.222/saml/sp/metadata/cloud_idp_onelogin`. In the URL, `cloud_idp_onelogin` is the tunnel group name.

9. Configure a SAML SP on your third party IdP following procedures provided by your third party IdP.

## Troubleshooting SAML 2.0

Use **debug webvpn saml***value* to debug SAML 2.0 behavior. The following SAML messages will be displayed depending on the *value* :

- 8—errors
- 16—warnings and errors
- 128 or 255—debug, warnings, and errors

# Monitor Secure Client Connections

To view information about active sessions, use the **show vpn-sessiondb** command:

| Command | Purpose |
|---|---|
| **show vpn-sessiondb** | Displays information about active sessions. |
| **vpn-sessiondb logoff** | Logs off VPN sessions. |
| **show vpn-sessiondb anyconnect** | Enhances the VPN session summary to show OSPFv3 session |
| **show vpn-sessiondb ratio encryption** | Shows the number of tunnels and percentages for the Suite B alg as AES-GCM-128, AES-GCM-192, AES-GCM-256, AES-GM so on). |

**Note** **AnyConnect Parent Tunnel**

AnyConnect parent tunnels do not have assigned IP addresses.

This is the main session that is created during the negotiation in order to set up the session token that is necessary in case a reconnect is needed due to network connectivity issues or hibernation. Based on the connection mechanism, the Cisco Adaptive Security Appliance (ASA) lists the session as Clientless (Weblaunch via the Portal) or Parent (Standalone AnyConnect).

AnyConnect parent represents the session when the client is not actively connected. Effectively, it works similar to a cookie, in that it is a database entry on the ASA that maps to the connection from a particular client. If the client sleeps/hibernates, the tunnels (IPsec/Internet Key Exchange (IKE)/ Transport Layer Security (TLS)/Datagram Transport Layer Security (DTLS) protocols) are torn down, but the Parent remains until the idle timer or maximum connect time takes effect. This allows the user to reconnect without reauthenticating.

### Example

The Inactivity field shows the elapsed time since an Secure Client session lost connectivity. If the session is active, 00:00m:00s appears in this field.

```
hostname# show vpn-sessiondb

Session Type: SSL VPN Client

Username     : lee
Index        : 1                      IP Addr      : 209.165.200.232
Protocol     : SSL VPN Client         Encryption   : 3DES
Hashing      : SHA1                   Auth Mode    : userPassword
TCP Dst Port : 443                    TCP Src Port : 54230
Bytes Tx     : 20178                  Bytes Rx     : 8662
Pkts Tx      : 27                     Pkts Rx      : 19
Client Ver   : Cisco STC 1.1.0.117
Client Type  : Internet Explorer
Group        : DfltGrpPolicy
Login Time   : 14:32:03 UTC Wed Mar 20 2007
Duration     : 0h:00m:04s
Inactivity    : 0h:00m:04s
Filter Name  :

hostname# vpn-sessiondb logoff
INFO: Number of sessions of type "" logged off : 1

hostname# vpn-sessiondb logoff name tester
Do you want to logoff the VPN session(s)? [confirm]
INFO: Number of sessions with name "tester" logged off : 1
```

# Log Off AnyConnect VPN Sessions

To log off all VPN sessions, use the **vpn-sessiondb logoff** command in global configuration mode:

The following example logs off all VPN sessions:

```
hostname# vpn-sessiondb logoff
INFO: Number of sessions of type "" logged off : 1
```

You can log off individual sessions using either the name argument or the index argument:

**vpn-sessiondb logoff name** *name*
**vpn-sessiondb logoff index** *index*

The sessions that have been inactive the longest time are marked as idle (and are automatically logged off) so that license capacity is not reached and new users can log in. If the session resumes at a later time, it is removed from the inactive list.

You can find both the username and the index number (established by the order of the client images) in the output of the **show vpn-sessiondb anyconnect** command. The following examples shows the username *lee* and index number *1*.

```
hostname# show vpn-sessiondb anyconnect

Session Type: AnyConnect

Username     : lee                    Index        : 1
Assigned IP  : 192.168.246.1           Public IP    : 10.139.1.2
Protocol     : AnyConnect-Parent SSL-Tunnel DTLS-Tunnel
License      : AnyConnect Premium
Encryption   : RC4 AES128             Hashing      : SHA1
Bytes Tx     : 11079                  Bytes Rx     : 4942
Group Policy : EngPolicy            Tunnel Group : EngGroup
Login Time   : 15:25:13 EST Fri Jan 28 2011
Duration     : 0h:00m:15s
Inactivity   : 0h:00m:00s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none
```

The following example terminates the session using the **name** option of the **vpn-session-db logoff** command:

```
hostname# vpn-sessiondb logoff name lee
Do you want to logoff the VPN session(s)? [confirm]
INFO: Number of sessions with name "lee" logged off : 1

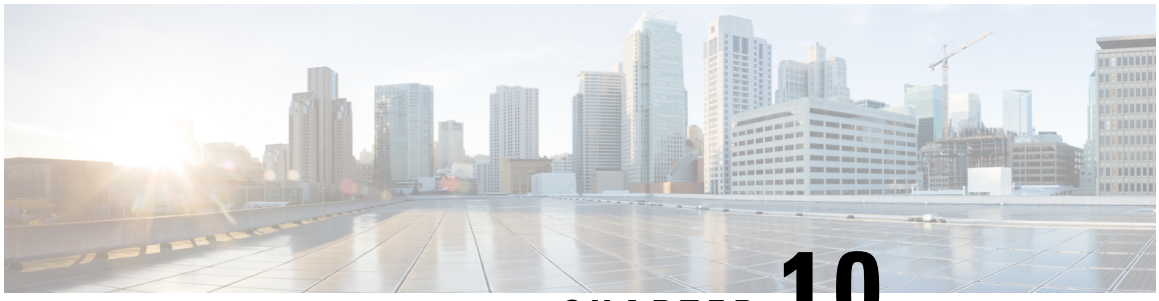
hostname#
```

# Feature History for Secure Client Connections

The following table lists the release history for this feature.

***Table 14: Feature History for Secure Client Connections***

| Feature Name | Releases | Feature Information |
|---|---|---|
| Secure Client Connections | 7.2(1) | The following commands were introduced or modified: authenticati authentication ms-chap-v1, authentication ms-chap-v2, authenticati tunnel hello, vpn-tunnel-protocol l2tp-ipsec. |
| IPsec IKEv2 | 8.4(1) | IKEv2 was added to support IPsec IKEv2 connections for Secure C LAN-to-LAN. |



CHAPTER 10

# Secure Client HostScan

The AnyConnect Posture Module provides the Secure Client the ability to identify the operating system, anti-malware and firewall software installed on the host. The HostScan application gathers this information. Posture assessment requires HostScan to be installed on the host.


- Prerequisites for HostScan/Secure Firewall Posture, on page 269
- Licensing for HostScan, on page 269
- HostScan Packaging, on page 270
- Install or Upgrade HostScan/Secure Firewall Posture, on page 270
- Enable or Disable HostScan, on page 271
- View the HostScan/Secure Firewall Posture Version Enabled on the ASA, on page 272
- Uninstall HostScan/Secure Firewall Posture, on page 272
- Assign Secure Client Feature Modules to Group Policies, on page 273
- HostScan/Secure Firewall Posture Related Documentation, on page 274


## Prerequisites for HostScan/Secure Firewall Posture

The Secure Client with the Secure Firewall Posture/HostScan module requires these minimum ASA components:

- ASA 8.4
- ASDM 6.4

You must install Secure Firewall Posture/HostScan to use the SCEP authentication feature.

Refer to Supported VPN Platforms, Cisco ASA Series for what operating systems are supported for Secure Firewall Posture/HostScan installation.

## Licensing for HostScan

These are the Secure Client licensing requirements for the HostScan:

- AnyConnect Apex
- AnyConnect VPN Only

# HostScan Packaging

You can load the HostScan package on to the ASA as a standalone package: **hostscan-version.pkg**. This file contains the HostScan software as well as the HostScan library and support charts.

# Install or Upgrade HostScan/Secure Firewall Posture

Use this procedure to install or upgrade the HostScan or Secure Firewall Posture package and enable it using the command line interface for the ASA.

### Before you begin



**Note** If you are attempting to upgrade to HostScan version 4.6.x or later from a 4.3.x version or earlier, you will receive an error message due to the fact that all existing AV/AS/FW DAP policies and LUA script(s) that you have previously established are incompatible with HostScan 4.6.x or greater.

There is a one time migration procedure that must be done to adapt your configuration. This procedure involves leaving this dialog box to migrate your configuration to be comptaible with HostScan 4.4.x before saving this configuration. Abort this procedure and refer to the Secure Client HostScan 4.3.x to 4.6.x Migration Guide for detailed instructions. Briefly, migration involves navigating to the ASDM DAP policy page to review and manually deleting the incompatible AV/AS/FW attributes, and then reviewing and rewriting LUA scripts.

- Log on to the ASA and enter global configuration mode. In global configuration mode, the ASA displays this prompt: hostname(config)#
- Upload the secure-firewall-posture-*version*-k9.pkg to the ASA. If you are using HostScan 4.x version, you should upload the hostscan_*version*-k9.pkg file.

### Procedure

**Step 1** Enter webvpn configuration mode.

**Example:**

```
hostname(config)# webvpn
```

**Step 2** Open ASDM and choose **Configuration > Remote Access VPN > Posture (for Secure Firewall) > Posture Image**. If you are using the HostScan 4.x version, the path will be **Configuration > Remote Access VPN > Secure Desktop Manager > Host Scan Image**.

**Step 3** Specify the path to the package you want to designate as the HostScan/Secure Firewall Posture image. You can specify a standalone package or the Secure Client package.

***hostscan image*** *path*

**Example:**

If you are using the HostScan 4.x version,

```
ASAName(webvpn)#hostscan image disk0:/hostscan_4.10.06081.pkg
```

If you are using the Secure Firewall Posture 5.x version,

```
ASAName(webvpn)#hostscan image disk0:/secure-firewall-posture5.0.00556.pkg
```

**Step 4** Enable the HostScan/Secure Firewall Posture image you designated in the previous step.

**Example:**

```
ASAName(webvpn)#hostscan enable
```

**Step 5** Save the running configuration to flash. After successfully saving the new configuration to flash memory, you receive the message [OK].

**Example:**

```
hostname(webvpn)# write memory
```

**Step 6**

# Enable or Disable HostScan

These commands enable or disable an installed HostScan image using the command line interface of the ASA.

### Before you begin

Log on to the ASA and enter global configuration mode. In global configuration mode, the ASA displays this prompt: hostname(config)#

### Procedure

**Step 1** Enter webvpn configuration mode.

**Example:**

**webvpn**

**Step 2** Enable the standalone HostScan image if it has not been uninstalled from your ASA.

**hostscan enable**

**Step 3** Disable HostScan for all installed HostScan packages.

**Note**
Before you uninstall the enabled HostScan image, you must first disable HostScan using this command.

**no hostscan enable**

# View the HostScan/Secure Firewall Posture Version Enabled on the ASA

Use this procedure to determine the enabled HostScan/Secure Firewall Posture version using ASA's command line interface.

### Before you begin

Log on to the ASA and enter privileged exec mode. In privileged exec mode, the ASA displays this prompt: hostname#

### Procedure

Show the version of HostScan/Secure Firewall Posture enabled on the ASA.

**show webvpn hostscan**

# Uninstall HostScan/Secure Firewall Posture

Uninstalling HostScan/Secure Firewall Posture package removes it from view on the ASDM interface and prevents the ASA from deploying it even when it is enabled. Uninstalling HostScan/Secure Firewall Posture does not delete the package from the flash drive.

### Before you begin

Log on to the ASA and enter global configuration mode. In global configuration mode, the ASA displays this prompt: hostname(config)#.

### Procedure

**Step 1** Enter webvpn configuration mode.

**webvpn**

**Step 2** Disable the HostScan/Secure Firewall Posture image you want to uninstall.

**no hostscanenable**

**Step 3** Specify the path to the HostScan/Secure Firewall Posture image you want to uninstall. A standalone package may have been designated as the HostScan/Secure Firewall Posture package.

**no hostscan image** *path*

**Example:**

If you are using the HostScan 4.x version,

```
ASAName(webvpn)#hostscan image disk0:/hostscan_4.10.06081-k9.pkg
```

If you are using the Secure Firewall Posture 5.x version,

```
ASAName(webvpn)#hostscan image disk0:/secure-firewall-posture-5.0.00556-k9.pkg
```

**Step 4** Save the running configuration to flash.After successfully saving the new configuration to flash memory, you receive the message [OK].

**write memory**

# Assign Secure Client Feature Modules to Group Policies

This procedure associates Secure Client feature modules with a group policy. When VPN users connect to the ASA, the ASA downloads and installs these Secure Client feature modules to their endpoint computer.

### Before you begin

Log on to the ASA and enter global configuration mode. In global configuration mode, the ASA displays this prompt: hostname(config)#

### Procedure

**Step 1** Adds an internal group policy for Network Client Access

**group-policy** *name* **internal**

**Example:**

```
hostname(config)# group-policy PostureModuleGroup internal
```

**Step 2** Edit the new group policy. After entering the command, you receive the prompt for group policy configuration mode, hostname(config-group-policy)#.

**group-policy** *name* **attributes**

**Example:**

```
hostname(config)# group-policy PostureModuleGroup attributes
```

**Step 3** Enter group policy webvpn configuration mode. After you enter the command, the ASA returns this prompt: hostname(config-group-webvpn)#

**webvpn**

**Step 4** Configure the group policy to download the Secure Client feature modules for all users in the group.

**anyconnect modules value** *Secure Firewall Module Name*

The value of the anyconnect module command can contain one or more of the following values. When specifying more than one module, separate the values with a comma:

| value | Secure Firewall Module/Feature Name |
|---|---|
| dart | Secure Client DART (Diagnostics and Reporting Tool) |
| vpngina | Secure Client SBL (Start Before Logon) |
| posture | Secure Firewall Posture/HostScan |
| nam | Secure Client Network Access Manager |
| none | Used by itself to remove all AnyConnect modules from the group policy. |
| profileMgmt | Secure Client Management Tunnel VPN |

**Example:**

```
hostname(config-group-webvpn)# anyconnect modules value websecurity,telemetry,posture
```

To remove one of the modules, re-send the command specifying only the module values you want to keep. For example, this command removes the websecurity module:

```
hostname(config-group-webvpn)# anyconnect modules value telemetry,posture
```

**Step 5** Save the running configuration to flash.

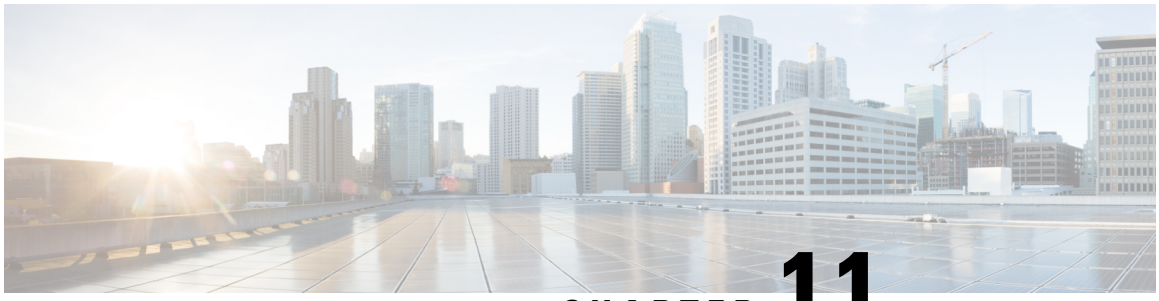
After successfully saving the new configuration to flash memory, you receive the message [OK] and the ASA returns you to this prompt hostname(config-group-webvpn)#

**write memory**

# HostScan/Secure Firewall Posture Related Documentation

Once HostScan/Secure Firewall Posture gathers the posture credentials from the endpoint computer, you will need to understand subjects like configuring dynamic access policies and using LUA expressions to make use of the information.

These topics are covered in detail in these documents: Cisco Adaptive Security Device Manager Configuration Guides. See also the Cisco *Secure Client (including AnyConnect) Administrator Guide* for more information about how HostScan/Secure Firewall Posture works with Secure Client.



CHAPTER 11

# Virtual Tunnel Interface

This chapter describes how to configure a VTI tunnel.


- About Virtual Tunnel Interfaces, on page 275
- Guidelines for Virtual Tunnel Interfaces, on page 276
- Create a VTI Tunnel, on page 279
- Feature History for Virtual Tunnel Interface, on page 288


## About Virtual Tunnel Interfaces

ASA supports a logical interface called the Virtual Tunnel Interface (VTI). As an alternative to policy-based VPN, you can create a VPN tunnel between peers using VTIs. VTIs support route-based VPN with IPsec profiles attached to the end of each tunnel. You can use dynamic or static routes. Egressing traffic from the VTI is encrypted and sent to the peer, and the associated SA decrypts the ingress traffic to the VTI.

Using VTI does away with the requirement of configuring static crypto map access lists and mapping them to interfaces. You no longer have to track all remote subnets and include them in the crypto map access list. Deployments become easier, and having static VTI which supports route-based VPN with dynamic routing protocol also satisfies many requirements of a virtual private cloud.

### Static VTI

You can use static VTI configurations for site-to-site connectivity in which a tunnel is always-on between two sites. For a static VTI interface, you must define a physical interface as a tunnel source. You can associate a maximum of 1024 VTIs per device. To create a static VTI interface, see Add a VTI Interface, on page 282.

### Dynamic VTI

Dynamic VTI provides highly secure and scalable connectivity for site-to-site VPNs. Dynamic VTI eases the configuration of peers for large enterprise hub and spoke deployments. A single dynamic VTI can replace several static VTI configurations on the hub. You can add new spokes to a hub without changing the hub configuration. Dynamic VTI replaces dynamic crypto maps and the dynamic hub-and-spoke method for establishing tunnels. In the management center, dynamic VTI supports only the hub and spoke topology.

Dynamic VTI uses a virtual template for dynamic instantiation and management of IPsec interfaces. The virtual template dynamically generates the virtual access interface that is unique for each VPN session. Dynamic VTI supports multiple IPsec security associations and accepts multiple IPsec selectors proposed by the spoke. Dynamic VTI also supports dynamic (DHCP) spokes. To create a dynamic VTI interface, see Add a Dynamic VTI Interface, on page 285.

### How Does an ASA Create a Dynamic VTI Tunnel for a VPN Session

1. Create a virtual template on ASA (**interface virtual-Template** *template_number* **type tunnel**).

   You can use this template for multiple VPN sessions.
2. Attach this template to a tunnel group. You can attach a virtual template to multiple tunnel groups.
3. Spoke initiates a tunnel request with the hub.
4. Hub authenticates the spoke.
5. ASA uses the virtual template to dynamically create a virtual access interface on the hub for the VPN session with the spoke.
6. Hub establishes a dynamic VTI tunnel with the spoke using the virtual access interface.
7. Configure the IKEv2 route set interface command to advertise the VTI interface IP over IKEv2 exchanges. This option enables unicast reachability between the VTI interfaces for BGP or path monitoring to work over the tunnel.
8. After the VPN session ends, the tunnel disconnects and the hub deletes the corresponding virtual access interface.

# Guidelines for Virtual Tunnel Interfaces

### Context Mode and Clustering

- Supported in single mode only.
- No support for clustering.

### Firewall Mode

Supported in routed mode only.

### BGP IPv4 and IPv6 Support

Supports IPv4 and IPv6 BGP routing over VTI.

### EIGRP Support

Supports IPv4 and IPv6 EIGRP routing over VTI.

### OSPF IPv4 and IPv6 Support

Supports IPv4 and IPv6 OSPF routing over VTI.

### IPv6 Support

- IPv6 addressed VTIs can be configured.
- Both the tunnel source and the tunnel destination of a VTI can have IPv6 addresses.
- Following combinations of VTI IP (or internal networks IP version) over public IP versions are supported:

- IPv6 over IPv6
- IPv4 over IPv6
- IPv4 over IPv4
- IPv6 over IPv4

- Only static IPv6 address is supported as the tunnel source and destination.
- The tunnel source interface can have IPv6 addresses and you can specify which address to be used as the tunnel endpoint. If you do not specify, by default, the first IPv6 global address in the list is used as the tunnel endpoint.
- You can specify the tunnel mode as IPv6. When specified, the IPv6 traffic can be tunneled through the VTI. However, the tunnel mode can either be IPv4 or IPv6 for a single VTI.

### Bidirectional Forwarding Detection (BFD) for IPsec Tunnel Interface Guidelines

- For a Static VTI interface with a BFD template, configure only one IP address type, either IPv4 or IPv6. If the BFD neighbor does not come up due to dual IP address types, remove and re-add the tunnel interface with a single IP address type.
- Enable BFD only on the tunnel interface for Static VTI interface and on the tunnel group for Dynamic VTI interface. You must not configure BFD neighbor under router BGP for the same overlay neighbor IP address.
- For BFD-enabled tunnels, set the Dead Peer Detection (DPD) interval to 60 seconds or more, depending on the total number of BFD tunnels on the device.
- When you configure a VTI or virtual template interface with IPv6 unnumbered, you cannot assign multiple IPv6 addresses to the referenced interface, if the VTI or virtual template interface has BFD enabled on it.

### General Configuration Guidelines

- If you use dynamic crypto maps and dynamic VTIs in your LAN-to-LAN VPNs, only the dynamic VTI tunnels will come up. This behaviour occurs because both the crypto maps and dynamic VTIs try to use the default tunnel group.

  We recommend that you do one of the following:

  - Migrate your LAN-to-LAN VPNs to dynamic VTIs.
  - Use static crypto maps with their own tunnel-groups.

- VTIs are only configurable in IPsec mode. To terminate GRE tunnels on an ASA is unsupported.
- You can use static, BGP, OSPF or EIGRP IPv4 routes for traffic using the tunnel interface.
- For static and dynamic VTI, ensure that you do not use the borrow IP interface as the tunnel source IP address for any VTI interface.
- The MTU for VTIs is automatically set, according to the underlying physical interface. However, if you change the physical interface MTU after the VTI is enabled, you must disable and reenable the VTI to use the new MTU setting.

- For dynamic VTI, the virtual access interface inherits the MTU from the configured tunnel source interface. If you do not specify the tunnel source interface, the virtual access interface inherits the MTU from the source interface from which ASA accepts the VPN session request.
- For static VTIs, you can configure a maximum of 1024 VTIs on a device. While calculating the VTI count, consider the following:
  - Include nameif subinterfaces to derive the total number of VTIs that can be configured on the device.
  - You cannot configure nameif on member interfaces of a portchannel. Therefore, the tunnel count is reduced by the count of actual main portchannel interfaces alone and not any of its member interfaces.
  - Even if a platform supports more than 1024 interfaces, the VTI count is limited to the number of VLANs configurable on that platform. For example, if a model supports 500 VLANs, then the tunnel count would be 500 minus the number of physical interfaces configured.
- For dynamic VTIs, the maximum number of dynamically-created virtual access interfaces can be 1024 or the total interface limit of the platform, whichever is less.
- VTI supports IKE versions v1, v2, and uses IPsec for sending and receiving data between the tunnel's source and destination.
- If NAT has to be applied, the IKE and ESP packets are encapsulated in the UDP header.
- IKE and IPsec security associations will be re-keyed continuously regardless of data traffic in the tunnel. This ensures that VTI tunnels are always up.
- The tunnel group name must match what the peer sends as its IKEv1 or IKEv2 identity.
- For IKEv1 in site-to-site tunnel groups, you can use names which are not IP addresses, if the tunnel authentication method is digital certificates and/or the peer is configured to use aggressive mode.
- VTI and crypto map configurations can co-exist on the same physical interface, provided the peer address configured in the crypto map and the tunnel destination for the VTI are different.
- Access rules can be applied on a VTI interface to control traffic through VTI.
- ICMP ping is supported between VTI interfaces.
- If the peer device for an IKEv2 site-to-site VPN tunnel sends IKEv2 configuration request payloads, the ASA cannot establish an IKEv2 tunnel with the device. You must disable the config-exchange request on the peer device for the ASA to establish a VPN tunnel with the peer device.
- Dynamic VTI supports HA and IKEv2.

### Default Settings

- By default, all traffic through VTI is encrypted.
- By default, the security level for VTI interfaces is 0. You cannot configure the security level.

### Limitations for VTI

ASA drops Security Group Tag (SGT) frames and packets after VTI decryption.

Dynamic VTI does not support:

- ECMP and VRF
- Clustering
- IKEv1
- QoS

For dynamic VTIs, if a tunnel source is not specified, IKEv2 will be enabled on all interfaces of the device except management-only and failover interfaces.

# Create a VTI Tunnel

To configure a VTI tunnel, create an IPsec proposal (transform set). You will need to create an IPsec profile that references the IPsec proposal, followed by a VTI interface with the IPsec profile. Configure the remote peer with identical IPsec proposal and IPsec profile parameters. SA negotiation will start when all tunnel parameters are configured.

**Note** For the ASA which is a part of both the VPN VTI domains, and has BGP adjacency on the physical interface:

When a state change is triggered due to the interface health check, the routes in the physical interface will be deleted until BGP adjacency is re-established with the new active peer. This behavior does not apply to logical VTI interfaces.

Access control lists can be applied on a VTI interface to control traffic through VTI. To permit any packets that come from an IPsec tunnel without checking ACLs for the source and destination interfaces, enter the sysopt connection permit-vpn command in global configuration mode.

You can use the following command to enable IPsec traffic through the ASA without checking ACLs:

**hostname(config)# sysopt connection permit-vpn**

When an outside interface and VTI interface have the security level of 0, if you have ACL applied on VTI interface, it will not be hit if you do not have same-security-traffic configured.

To configure this feature, use the **same-security-traffic** command in global configuration mode with its **intra-interface** argument.

For more information, see Permitting Intra-Interface Traffic (Hairpinning), on page 68.

### Procedure

**Step 1** Add an IPsec Proposal (Transform Sets).

**Step 2** Add an IPsec Profile.

**Step 3** Add a VTI Tunnel.

# Add an IPsec Proposal (Transform Sets)

A transform set is required to secure traffic in a VTI tunnel. Used as a part of the IPsec profile, it is a set of security protocols and algorithms that protects the traffic in the VPN.

### Before you begin

- You can use either pre-shared key or certificates for authenticating the IKE session associated with a VTI. IKEv2 allows asymmetric authentication methods and keys. For both IKEv1 and IKEv2, you must configure the pre-shared key under the tunnel group used for the VTI.
- For certificate based authentication using IKEv1, you must specify the trustpoint to be used at the initiator. For the responder, you must configure the trustpoint in the tunnel-group command. For IKEv2, you must configure the trustpoint to be used for authentication under the tunnel group command for both initiator and responder.

## Procedure

Add an IKEv1 transform set, or an IKEv2 IPsec proposal to establish the security association.

Add an IKEv1 transform set:

**crypto ipsec ikev1 transform-set** {*transform-set-name* | *encryption* | *authentication*}

**Example:**

```
ciscoasa(config)#crypto ipsec ikev1 transform-set SET1 esp-aes esp-sha-hmac
```

*Encryption* specifies which encryption method protects IPsec data flows:

- esp-aes—Uses AES with a 128-bit key.
- esp-aes-192—Uses AES with a 192-bit key.
- esp-aes-256—Uses AES with a 256-bit key.
- esp-null—No encryption.

*Authentication* specifies which encryption method to protect IPsec data flows:

- esp-md5-hmac—Uses the MD5/HMAC-128 as the hash algorithm.
- esp-sha-hmac—Uses the SHA/HMAC-160 as the hash algorithm.
- esp-none—No HMAC authentication.

Add an IKEv2 IPsec proposal.

**Note**
For the IOS platform, use the **no config-exchange request** command in the IKEv2 profile configuration mode to disable configuration exchange options. See http://www.cisco.com/c/en/us/td/docs/ios-xml/ios/security/a1/sec-a1-cr-book/sec-cr-c2.html#wp3456426280 for more information.

- Specify a name for the IPsec proposal:

  **crypto ipsec ikev2 ipsec-proposal** *IPsec proposal name*

**Example:**

```
ciscoasa(config)#crypto ipsec ikev2 ipsec-proposal SET1
```

- Specify the security parameters in the crypto IPsec ikev2 ipsec-proposal configuration mode:

  **protocol esp {encryption {aes | aes-192 | aes-256 | aes-gcm | aes-gcm-192 | aes-gcm-256 | null} | integrity {sha-1 | sha-256 | sha-384 | sha-512 | null}**

**Example:**

```
ciscoasa(config-ipsec-proposal)#protocol esp encryption aes aes-192
```

# Add an IPsec Profile

An IPsec profile contains the required security protocols and algorithms in the IPsec proposal or transform set that it references. This ensures a secure, logical communication path between two site-to-site VTI VPN peers.

**Procedure**

**Step 1** Set a name for the profile:

**crypto ipsec profile** *name*

**Example:**

```
ciscoasa(config)#crypto ipsec profile PROFILE1
```

**Step 2** Set the IKEv1 or IKEv2 proposal. You can choose either an IKEv1 transform set or an IKEv2 IPsec proposal.

a) Set the IKEv1 transform set.

- To set the IKEv1 proposal, enter the following command in the crypto ipsec profile command sub-mode:

  **set ikev1 transform set** *set_name*

  In this example, SET1 is the IKEv1 proposal set created previously.

  ```
  ciscoasa(config-ipsec-profile)#set ikev1 transform-set SET1
  ```

b) Set the IKEv2 proposal.

- To set the IKEv2 proposal, enter the following command in the crypto ipsec profile command sub-mode:

  **set ikev2 ipsec-proposal** *IPsec_proposal_name*

  In this example, SET1 is the IKEv2 IPsec proposal created previously.

  ```
  ciscoasa(config-ipsec-profile)#set ikev2 ipsec-proposal SET1
  ```

**Step 3** (Optional) Specify the duration of the security association:

**set security-association lifetime** {**seconds** *number* | **kilobytes** {*number* | **unlimited**}}

**Example:**

```
ciscoasa(config-ipsec-profile)#set security-association lifetime
seconds 120 kilobytes 10000
```

**Step 4** (Optional) Configure the end of the VTI tunnel to act only as a responder:

**responder-only**

- You can configure one end of the VTI tunnel to perform only as a responder. The responder-only end will not initiate the tunnel or rekeying.
- If you are using IKEv2, set the duration of the security association lifetime, greater than the lifetime value in the IPsec profile in the initiator end. This is to facilitate successful rekeying by the initiator end and ensure that the tunnels remain up.
- If you are using IKEv1, IOS should always be in responder-only mode since IOS doesn't support continuous channel mode. The ASA becomes the initiator and session and rekeys.
- If the rekey configuration in the initiator end is unknown, remove the responder-only mode to make the SA establishment bi-directional, or configure an infinite IPsec lifetime value in the responder-only end to prevent expiry.

**Step 5** (Optional) Specify the PFS group. Perfect Forward Secrecy (PFS) generates a unique session key for each encrypted exchange. This unique session key protects the exchange from subsequent decryption. To configure PFS, you have to select the Diffie-Hellman key derivation algorithm to use when generating the PFS session key. The key derivation algorithms generate IPsec security association (SA) keys. Each group has a different size modulus. A larger modulus provides higher security, but requires more processing time. You must have matching Diffie-Hellman groups on both peers.

**set pfs** { **group14** }

**Example:**

```
ciscoasa(config-ipsec-profile)# set pfs group14
```

**Step 6** (Optional) Specify a trustpoint that defines the certificate to be used while initiating a VTI tunnel connection.

**set trustpoint** *name*

**Example:**

```
ciscoasa(config-ipsec-profile)#set trustpoint TPVTI
```

**Step 7** (Optional) Enable Reverse Route Injection (RRI) for this IPsec profile and set the reverse route as dynamic.

**set reverse-route** [ dynamic]

**Example:**

```
ciscoasa(config-ipsec-profile)#set reverse-route dynamic
```

# Add a VTI Interface

To create a new VTI interface and establish a VTI tunnel, perform the following steps:

**Note** Implement IP SLA to ensure that the tunnel remains up when a router in the active tunnel is unavailable. See Configure Static Route Tracking in the ASA General Operations Configuration Guide in http://www.cisco.com/go/asa-config.

## Procedure

**Step 1** Create a new tunnel interface:

**interface tunnel** *tunnel_interface_number*

Specify a tunnel ID, from a range of 0 to 10413. Up to 10413 VTI interfaces are supported.

**Example:**

```
ciscoasa(config)#interface tunnel 100
```

**Step 2** Enter the name of the VTI interface.

Enter the following command in the **interface tunnel** command submode:

**nameif** *interface name*

**Example:**

```
ciscoasa(config-if)#nameif vti
```

**Step 3** Enter the IP address of the VTI interface.

**ip address** *IP addressmask*

**Example:**

```
ciscoasa(config-if)#ip address 192.168.1.10 255.255.255.254
```

**Step 4** Enter an IPv4 or IPv6 address of an interface that the virtual template inherits.

You can choose any physical interface or a loopback address configured on the device. All the virtual access interfaces that are cloned from the virtual template will have the same IP address.

**ip unnumbered** *interface-name*

**ipv6 unnumbered** *interface-name*

**Example:**

```
ciscoasa(config-if)#ip unnumbered loopback1
```

**Step 5** Specify the tunnel source interface.

**tunnel source interface** *interface_name*

The source interface can be a physical interface or a loopback interface.

**Example:**

```
ciscoasa(config-if)#tunnel source interface outside
```

**Step 6** Specify the tunnel destination IP address.

**tunnel destination** *ip_address*

**Example:**

```
ciscoasa(config-if)#tunnel destination 10.1.1.1
```

**Step 7** Configure the tunnel with tunnel mode IPsec IPv4.

**tunnel mode ipsec** *ipv4*

**Example:**

```
ciscoasa(config-if)#tunnel mode ipsec ipv4
```

**Step 8** Assign the IPsec profile to tunnel.

**tunnel protection ipsec** *IPsec profile*

**Example:**

```
ciscoasa(config-if)#tunnel protection ipsec Profile1
```

**Step 9** Assign a traffic selector for the static VTI interface.

**tunnel protection policy** *acl_name*

The access list can contain single or multiple list selectors. If you do not configure this command, the static VTI interface proposes any-any selectors, which is the default behavior.

**Example:**

```
ciscoasa(config)# access-list Spoke-to-Hub extended permit ip 209.165.200.225 255.255.255.224
 any
ciscoasa(config-if)# tunnel protection ipsec policy Spoke-to-Hub
```

---

**Example**

Example configuration of a VTI tunnel (with IKEv2) between ASA and an IOS device:

```
ASA:

crypto ikev2 policy 1
 encryption aes-gcm-256
 integrity null
 group 21
 prf sha512
 lifetime seconds 86400
!
crypto ipsec ikev2 ipsec-proposal gcm256
 protocol esp encryption aes-gcm-256
 protocol esp integrity null
!
crypto ipsec profile asa-vti
 set ikev2 ipsec-proposal gcm256
!
interface Tunnel 100
 nameif vti
 ip address 10.10.10.1 255.255.255.254
 tunnel source interface [asa-source-nameif]
```

```
 tunnel destination [router-ip-address]
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile asa-vti
!
tunnel-group [router-ip-address] ipsec-attributes
 ikev2 remote-authentication pre-shared-key cisco
 ikev2 local-authentication pre-shared-key cisco
!
crypto ikev2 enable [asa-interface-name]


IOS:

!
crypto ikev2 proposal asa-vti
 encryption aes-gcm-256
 prf sha512
 group 21
!
crypto ikev2 policy asa-vti
 match address local [router-ip-address]
 proposal asa-vti
!
crypto ikev2 profile asa-vti
 match identity remote address [asa-ip-address] 255.255.255.255
 authentication local pre-share key cisco
 authentication remote pre-share key cisco
 no config-exchange request
!
crypto ipsec transform-set gcm256 esp-gcm 256
!
crypto ipsec profile asa-vti
 set ikev2-profile asa-vti
 set transform-set gcm256
!
interface tunnel 100
 ip address 10.10.10.0 255.255.255.254
 tunnel mode ipsec ipv4
 tunnel source [router-interface]
 tunnel destination [asa-ip-address]
 tunnel protection ipsec profile asa-vti
!
```

# Add a Dynamic VTI Interface

To create a virtual template for dynamic VTI:

**Note** Implement IP SLA to ensure that the tunnel remains up when a router in the active tunnel is unavailable. See "Configure Static Route Tracking" in the ASA General Operations Configuration Guide in http://www.cisco.com/go/asa-config.

### Before you begin

Ensure that you have configured an IPsec profile and an IP unnumbered interface.

### Procedure

**Step 1** Create a new virtual template:

**interface virtual-Template** *template_number* **type tunnel**

*template_number* is the unique number of the virtual template. The range is from 1 to 10413..

The interface template must not be in the shutdown state. Following are the mandatory parameters for the virtual template:

- Interface Name
- Tunnel IPsec Mode
- Tunnel IPsec Profile

**Example:**

```
ciscoasa(config)#interface virtual-Template 101 type tunnel
```

**Step 2** Specify the name of the dynamic VTI virtual template interface.

Enter the following command in the **interface** configuration mode:

**nameif** *interface_name*

ASA dynamically creates virtual access interfaces as <*Virtual_Template_name*>_va<*n*>. For example, if the name of the virtual template is dVTI101, the virtual access interfaces will be dVTI101_va1, dVTI101_va2, and so on. If you want to modify a virtual template, you must shutdown the virtual template using the **shutdown** command.

**Example:**

```
ciscoasa(config-if)#nameif dVTI101
```

**Step 3** Configure an IPv4 or IPv6 address of an interface that the virtual template inherits.

**ip unnumbered** *interface-name*

**ipv6 unnumbered** *interface-name*

The virtual template can inherit the IP address of any physical interface or a loopback address configured on the device. All the virtual access interfaces cloned from the virtual template will have the same IP address.

**Example:**

```
ciscoasa(config-if)#ip unnumbered loopback1
```

**Step 4** (Optional) Specify the tunnel source interface.

**tunnel source interface** *interface_name*

The source interface can be a physical interface or a loopback interface.

The ASA accepts VPN session requests only from the interface configured as the tunnel source IP address. If you do not specify this interface, ASA accepts VPN session requests received from any interface. The

virtual access interface inherits the MTU from the configured tunnel source interface. If you do not enable the above option, the virtual access interface inherits the MTU from the source interface from which ASA accepts the VPN session request.

**Example:**

```
ciscoasa(config-if)#tunnel source interface outside1
```

**Step 5** Specify the tunnel protection mode as IPv4 or IPv6.

**tunnel mode ipsec** {**ipv4** | **ipv6**}

**Example:**

```
ciscoasa(config-if)#tunnel mode ipsec ipv4
```

**Step 6** Assign an IPsec profile to the tunnel.

**tunnel protection ipsec profile** *ipsec_profile*

This IPsec profile configures the IPSec/IKE parameters required to negotiate the exchange.

**Example:**

```
ciscoasa(config-if)#tunnel protection ipsec profile Profile1
```

**Step 7** Attach the virtual template to a tunnel group.

**tunnel-group** *tunnel_group_name* **type** *type*

**tunnel-group** *tunnel_group_name* **ipsec-attributes**

**virtual-template** *template_number*

You can attach the same virtual template to multiple tunnel groups. ASA uses the virtual template to create individual virtual access interfaces for each VPN session.

**Example:**

```
ciscoasa(config)#tunnel-group DVTI_spoke1 type ipsec-l2l
ciscoasa(config)#tunnel-group DVTI_spoke1 ipsec-attributes
ciscoasa(config-tunnel-ipsec)#virtual-template 101
```

**Step 8** Enable dynamic routing for the tunnel group.

**tunnel-group** *tunnel_group_name* **ipsec-attributes**

**ikev2 route accept any**

**ikev2 route set interface**

The **ikev2 route accept any** command allows ASA to accept any tunnel interface IP addresses received during IKEv2 exchanges. By default, this option is enabled.

The **ikev2 route set interface** command allows ASA to send the tunnel interface IP address during IKEv2 exchanges. This option enables unicast reachability between the VTI interfaces for BGP to work over the tunnel.

Dynamic routing is enabled for the tunnel groups using BGP/OSPF/EIGRP. After you configure the virtual template, you must configure a routing policy to route the dynamic VTI traffic between the devices over the VTI tunnel. You must also configure an access control rule to allow encrypted traffic.

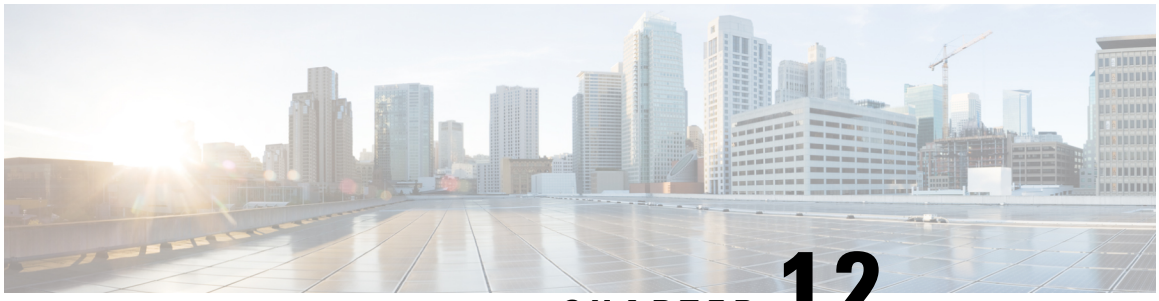
**Example:**

```
ciscoasa(config)#tunnel-group DVTI_spoke1 ipsec-attributes
ciscoasa(config-tunnel-ipsec)#ikev2 route set interface
ciscoasa(config-tunnel-ipsec)#ikev2 route accept any
```

# Feature History for Virtual Tunnel Interface

| Feature Name | Releases | Feature Information |
|---|---|---|
| Dynamic Virtual Tunnel Interface support | 9.19(1) | You can create a dynamic VTI and use it to configure a route-based site-to-site VPN in a hub and spoke topology. Dynamic VTI eases the configuration of peers for large enterprise hub and spoke deployments. A single dynamic VTI can replace several static VTI configurations on the hub. You can add new spokes to a hub without changing the hub configuration.<br>New/Modified commands: **interface virtual-Template, ip unnumbered, ipv6 unnumbered, tunnel protection ipsec policy** |
| OSPF IPv4 and IPv6 support | 9.19(1) | Supports OSPF IPv4 and IPv6 routing protocol over a VTI. |
| EIGRP support | 9.19(1) | Supports EIGRP IPv4 and IPv6 routing protocol over a VTI. |
| Loopback interface support for static and dynamic VTIs | 9.19(1) | You can now set a loopback interface as the source interface for a VTI. Support has also been added to inherit the IP address from a loopback interface instead of a statically configured IP address. The loopback interface helps to overcome path failures. If an interface goes down, you can access all interfaces through the IP address assigned to the loopback interface.<br>New/Modified commands: **tunnel source interface**, **ip unnumbered**, **ipv6 unnumbered** |
| Local tunnel ID support | 9.17(1) | ASA supports unique local tunnel ID that allows ASA to have multiple IPsec tunnel behind a NAT to connect to Cisco Umbrella Secure Internet Gateway (SIG). The local identity is used to configure a unique identity per IKEv2 tunnel, instead of a global identity for all the tunnels.<br>New/Modified commands: **local-identity-from-cryptomap** , |
| Support for IPv6 on Static VTI | 9.16(1) | ASA supports IPv6 addresses in Virtual Tunnel Interfaces (VTI) configurations.<br>A VTI tunnel source interface can have an IPv6 address, which you can configure to use as the tunnel endpoint. If the tunnel source interface has multiple IPv6 addresses, you can specify which address to be used, else the first IPv6 global address in the list is used by default.<br>The tunnel mode can be either IPv4 or IPv6, but it must be the same as IP address type configured on VTI for the tunnel to be active. An IPv6 address can be assigned to the tunnel source or the tunnel destination interface in a VTI.<br>New/Modified commands: **tunnel source interface**, **tunnel destination**, **tunnel mode** |

| Feature Name | Releases | Feature Information |
|---|---|---|
| Support for 1024 VTI interfaces per device | 9.16(1) | The number of maximum VTIs to be configured on a device has been increased from 100 to 1024.<br>Even if a platform supports more than 1024 interfaces, the VTI count is limited to the number of VLANs configurable on that platform. For example, ASA 5510 supports 100 VLANs, the tunnel count would be 100 minus the number of physical interfaces configured.<br>New/Modified commands: None |
| DHCP Relay Server Support on VTI | 9.14(1) | ASA allows VTI interfaces to be configured as DHCP relay server connecting interfaces.<br>We modified the following commands: **dhcprelay server** *ip_address vti_ifc_name*. |
| Support for IKEv2, certificate based authentication, and ACL in VTI | 9.8.(1) | Virtual Tunnel Interface (VTI) now supports BGP (static VTI). You can now use IKEv2 in standalone and high availability modes. You can use certificate based authentication by setting up a trustpoint in the IPsec profile. You can also apply access lists on VTI using access-group commands to filter ingress traffic.<br>We introduced the following command in the IPsec profile configuration mode: **set trustpoint**. |
| Virtual Tunnel Interface (VTI) support | 9.7.(1) | The ASA is enhanced with a new logical interface called Virtual Tunnel Interface (VTI), used to represent a VPN tunnel to a peer. This supports route based VPN with IPsec profiles attached to each end of the tunnel. Using VTI does away with the need to configure static crypto map access lists and map them to interfaces.<br>We introduced the following commands: **crypto ipsec profile, interface tunnel, responder-only, set ikev1 transform-set, set pfs, set security-association lifetime, tunnel destination, tunnel mode ipsec, tunnel protection ipsec profile, tunnel source interface**. |



CHAPTER 12

# Configure an External AAA Server for VPN


- About External AAA Servers, on page 291
- Guidelines For Using External AAA Servers, on page 292
- Configure Multiple Certificate Authentication, on page 292
- Configure LDAP Authorization for VPN, on page 294
- Active Directory/LDAP VPN Remote Access Authorization Examples, on page 307


## About External AAA Servers

This ASA can be configured to use an external LDAP, RADIUS, or TACACS+ server to support Authentication, Authorization, and Accounting (AAA) for the ASA. The external AAA server enforces configured permissions and attributes. Before you configure the ASA to use an external server, you must configure the external AAA server with the correct ASA authorization attributes and, from a subset of these attributes, assign specific permissions to individual users.

### Understanding Policy Enforcement of Authorization Attributes

The ASA supports several methods of applying user authorization attributes (also called user entitlements or permissions) to VPN connections. You can configure the ASA to obtain user attributes from any combination of:

- a Dynamic Access Policy (DAP) on the ASA
- an external RADIUS or LDAP authentication and/or authorization server
- a group policy on the ASA

If the ASA receives attributes from all sources, the attributes are evaluated, merged, and applied to the user policy. If there are conflicts between attributes, the DAP attributes take precedence.

The ASA applies attributes in the following order:

1. DAP attributes on the ASA—Introduced in Version 8.0(2), these attributes take precedence over all others. If you set a bookmark or URL list in DAP, it overrides a bookmark or URL list set in the group policy.
2. User attributes on the AAA server—The server returns these attributes after successful user authentication and/or authorization. Do not confuse these with attributes that are set for individual users in the local AAA database on the ASA (User Accounts in ASDM).

3. Group policy configured on the ASA—If a RADIUS server returns the value of the RADIUS CLASS attribute IETF-Class-25 (OU=*group-policy*) for the user, the ASA places the user in the group policy of the same name and enforces any attributes in the group policy that are not returned by the server.

   For LDAP servers, any attribute name can be used to set the group policy for the session. The LDAP attribute map that you configure on the ASA maps the LDAP attribute to the Cisco attribute IETF-Radius-Class.

4. Group policy assigned by the Connection Profile (called tunnel-group in the CLI)—The Connection Profile has the preliminary settings for the connection, and includes a default group policy applied to the user before authentication. All users connecting to the ASA initially belong to this group, which provides any attributes that are missing from the DAP, user attributes returned by the server, or the group policy assigned to the user.

5. Default group policy assigned by the ASA (DfltGrpPolicy)—System default attributes provide any values that are missing from the DAP, user attributes, group policy, or connection profile.

# Guidelines For Using External AAA Servers

The ASA enforces the LDAP attributes based on attribute name, not numeric ID. RADIUS attributes, are enforced by numeric ID, not by name.

For ASDM Version 7.0, LDAP attributes include the cVPN3000 prefix. For ASDM Versions 7.1 and later, this prefix was removed.

LDAP attributes are a subset of the Radius attributes, which are listed in the Radius chapter.

# Configure Multiple Certificate Authentication

You can now validate multiple certificates per session with the Secure Client SSL and IKEv2 client protocols. For example, you can make sure that the issuer name of the machine certificate matches a particular CA and therefore that the device is a corporate-issued device.

The multiple certificates option allows certificate authentication of both the machine and user via certificates. Without this option, you could only do certificate authentication of one or the other, but not both.



**Note** Because multiple certificate authentication requires a machine certificate and a user certificate (or two user certificates), you cannot use Secure Client start before logon (SBL) with this feature.

The pre-fill username field allows a field from the second (user) certificate to be parsed and used for subsequent AAA authentication in a AAA and certificate authenticated connection. The username for both primary and secondary prefill is always retrieved from the second (user) certificate received from the client.

Beginning with 9.14(1), ASA allows you to specify which certificate the primary and secondary username should come from when configuring multiple certificate authentication and using the pre-fill username option for Authentication or Authorization. For information, see Configure Multiple Certificate Username, on page 293

With multiple certificate authentication, two certificates are authenticated: the second (user) certificate received from the client is the one that the pre-fill and username-from-certificate primary and secondary usernames are parsed from.

You can also configure multiple certificate authentication with SAML.

The existing authentication webvpn attributes is modified to include an option for multiple-certificate authentication:

```
tunnel-group <name> webvpn-attributes
authentication {aaa [certificate | multiple-certificate] | multiple-certificate [aaa | saml]
 | saml [certificate | multiple-certificate]}
```

With multiple-certificate authentication, you can make policy decisions based on the fields of a certificate used to authenticate that connection attempt. The user and machine certificate received from the client during multiple-certificate authentication is loaded into DAP to allow policies to be configured based on the field of the certificate. To add multiple certificate authentication using Dynamic Access Policies (DAP) so that you can set up rules to allow or disallow connection attempts, refer to *Add Multiple Certificate Authentication to DAP* in the appropriate release of the ASA VPN ASDM Configuration Guide.

# Configure Multiple Certificate Username

A new command was introduced in ASA 9.14(1) to configure the certificate that ASA must use as the primary and secondary username for authentication or authorization. You can specify whether to use the machine certificate sent in SSL or IKE (first certificate) or the user certificate from client (second certificate) to get the authentication and authorization parameters. This option is available and can be configured for any tunnel groups irrespective of the authentication type (**aaa**, **certificate**, or **multiple-certificate**). However, the configuration takes effect only for Multiple Certificate Authentication (**multiple-certificate** or **aaa multiple-certificate**). When the option is not used for Multiple Certificate Authentication, the second certificate is used by default for authentication or authorization.

### Procedure

**Step 1** Specify whether to use the primary username from the first or second certificate:

**username-from-certificate-choice {first-certificate | second-certificate}**

**Step 2** Specify whether to use the secondary username from the first or second certificate:

**secondary-username-from-certificate-choice {first-certificate | second-certificate}**

**Example:**

```
tunnel-group tg1 webvpn-attributes
authentication aaa multiple-certificate
pre-fill-username client
secondary-pre-fill-username client
tunnel-group tg1 type remote-access
tunnel-group tg1 general-attributes
secondary-authentication-server-group LOCAL
username-from-certificate-choice first-certificate
secondary-username-from-certificate-choice first-certificate
```

# Configure LDAP Authorization for VPN

After LDAP authentication for VPN access has succeeded, the ASA queries the LDAP server, which returns LDAP attributes. These attributes generally include authorization data that applies to the VPN session.

You may require authorization from an LDAP directory server that is separate and distinct from the authentication mechanism. For example, if you use an SDI or certificate server for authentication, no authorization information is passed back. For user authorizations in this case, you can query an LDAP directory after successful authentication, accomplishing authentication and authorization in two steps.

To set up VPN user authorization using LDAP, perform the following steps.

### Procedure

**Step 1** Create a AAA server group.

**aaa-server** *server_group* **protocol** {**ldap** | **nt** | **radius** | **sdi** | **tacacs+**}

**Example:**

```
hostname(config)# aaa-server servergroup1 protocol ldap
hostname(config-aaa-server-group)
```

**Step 2** Create an IPsec remote access tunnel group named remotegrp.

**tunnel-group** *groupname*

**Example:**

```
hostname(config)# tunnel-group remotegrp
```

**Step 3** Associate the server group and the tunnel group.

**tunnel-group** *groupname* **general-attributes**

**Example:**

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

**Step 4** Assigns a new tunnel group to a previously created AAA server group for authorization.

***authorization-server-group*** *group-tag*

**Example:**

```
hostname(config-general)# authorization-server-group ldap_dir_1
```

### Example

The following example shows commands for enabling user authorization with LDAP. The example then creates an IPsec remote access tunnel group named RAVPN and assigns that new tunnel group to the previously created LDAP AAA server group for authorization:

```
hostname(config)# tunnel-group RAVPN type remote-access
hostname(config)# tunnel-group RAVPN general-attributes
hostname(config-general)# authorization-server-group (inside) LDAP
hostname(config-general)#
```

After you complete this configuration work, you can then configure additional LDAP authorization parameters such as a directory password, a starting point for searching a directory, and the scope of a directory search by entering the following commands:

```
hostname(config)# aaa-server LDAP protocol ldap
hostname(config-aaa-server-group)# aaa-server LDAP (inside) host 10.0.2.128
hostname(config-aaa-server-host)# ldap-base-dn DC=AD,DC=LAB,DC=COM
hostname(config-aaa-server-host)# ldap-group-base-dn DC=AD,DC=LAB,DC=COM
hostname(config-aaa-server-host)# ldap-scope subtree
hostname(config-aaa-server-host)# ldap-login-dn AD\cisco
hostname(config-aaa-server-host)# ldap-login-password cisco123
hostname(config-aaa-server-host)# ldap-over-ssl enable
hostname(config-aaa-server-host)# server-type microsoft
```

# Define the ASA LDAP Configuration

This section describes how to define the LDAP AV-pair attribute syntax and includes the following information:

- Supported Cisco Attributes for LDAP Authorization, on page 295
- Cisco AV Pair Attribute Syntax, on page 306
- Cisco AV Pairs ACL Examples, on page 307



**Note** The ASA enforces the LDAP attributes based on attribute name, not numeric ID. RADIUS attributes, on the other hand, are enforced by numeric ID, not by name.

Authorization refers to the process of enforcing permissions or attributes. An LDAP server defined as an authentication or authorization server enforces permissions or attributes if they are configured.

For ASA Version 7.0, LDAP attributes include the cVPN3000 prefix. For software Versions 7.1 and later, this prefix was removed.

## Supported Cisco Attributes for LDAP Authorization

This section provides a complete list of attributes (see ) for the ASA 5500, VPN 3000 concentrator, and PIX 500 series ASAs. The table includes attribute support information for the VPN 3000 concentrator and PIX 500 series ASAs to assist you in configuring networks with a combination of these devices.

*Table 15: ASA Supported Cisco Attributes for LDAP Authorization*

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| Access-Hours | Y | Y | Y | String | Single | Name of the time-range<br>(for example, Business-Hours) |
| Allow-Network-Extension-Mode | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| Authenticated-User-Idle-Timeout | Y | Y | Y | Integer | Single | 1 - 35791394 minutes |
| Authorization-Required | Y | | | Integer | Single | 0 = No<br>1 = Yes |
| Authorization-Type | Y | | | Integer | Single | 0 = None<br>1 = RADIUS<br>2 = LDAP |
| Banner1 | Y | Y | Y | String | Single | Banner string for clientless and client SSL VPN, and IPsec clients. |
| Banner2 | Y | Y | Y | String | Single | Banner string for clientless and client SSL VPN, and IPsec clients. |
| Cisco-AV-Pair | Y | Y | Y | String | Multi | An octet string in the following format:<br>*[Prefix] [Action] [Protocol] [Source] [Source Wildcard Mask] [Destination] [Destination Wildcard Mask] [Established] [Log] [Operator] [Port]*<br>For more information, see the "Cisco AV Pair Attribute Syntax" section." |
| Cisco-IP-Phone-Bypass | Y | Y | Y | Integer | Single | 0 = Disabled<br>1 = Enabled |
| Cisco-LEAP-Bypass | Y | Y | Y | Integer | Single | 0 = Disabled<br>1 = Enabled |
| Client-Intercept-DHCP-Configure-Msg | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| Client-Type-Version-Limiting | Y | Y | Y | String | Single | IPsec VPN client version number string |
| Confidence-Interval | Y | Y | Y | Integer | Single | 10 - 300 seconds |
| DHCP-Network-Scope | Y | Y | Y | String | Single | IP address |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| DN-Field | Y | Y | Y | String | Single | Possible values: UID, OU, O, CN, L, SP, C, EA, T, N, GN, SN, I, GENQ, DNQ, SER, and use-entire-name. |
| Firewall-ACL-In | | Y | Y | String | Single | Access list ID |
| Firewall-ACL-Out | | Y | Y | String | Single | Access list ID |
| Group-Policy | | Y | Y | String | Single | Sets the group policy for the remote access VPN session. For version 8.2 and later, use this attribute instead of IETF-Radius-Class. You can use one of the three following formats:<br>• group policy name<br>• OU= group policy name<br>• OU= group policy name : |
| IE-Proxy-Bypass-Local | | | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IE-Proxy-Exception-List | | | | String | Single | A list of DNS domains. Entries must be separated by the new line character sequence (\n). |
| IE-Proxy-Method | Y | Y | Y | Integer | Single | 1 = Do not modify proxy settings<br>2 = Do not use proxy<br>3 = Auto detect<br>4 = Use ASA setting |
| IE-Proxy-Server | Y | Y | Y | Integer | Single | IP address |
| IETF-Radius-Class | Y | Y | Y | | Single | Sets the group policy for the remote access VPN session. For version 8.2 and later, use this attribute instead of IETF-Radius-Class. You can use one of the three following formats:<br>• group policy name<br>• OU= group policy name<br>• OU= group policy name : |
| IETF-Radius-Filter-Id | Y | Y | Y | String | Single | Access list name that is defined on the ASA. The setting applies to VPN remote access IPsec and SSL VPN clients. |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| IETF-Radius-Framed-IP-Address | Y | Y | Y | String | Single | An IP address. The setting applies to VPN remote access IPsec and SSL VPN clients. |
| IETF-Radius-Framed-IP-Netmask | Y | Y | Y | String | Single | An IP address mask. The setting applies to VPN remote access IPsec and SSL VPN clients. |
| IETF-Radius-Idle-Timeout | Y | Y | Y | Integer | Single | Seconds |
| IETF-Radius-Service-Type | Y | Y | Y | Integer | Single | 1 = Login<br>2 = Framed<br>5 = Remote access<br>6 = Administrative<br>7 = NAS prompt |
| IETF-Radius-Session-Timeout | Y | Y | Y | Integer | Single | Seconds |
| IKE-Keep-Alives | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Allow-Passwd-Store | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Authentication | Y | Y | Y | Integer | Single | 0 = None<br>1 = RADIUS<br>2 = LDAP (authorization only)<br>3 = NT Domain<br>4 = SDI (RSA)<br>5 = Internal<br>6 = RADIUS with Expiry<br>7 = Active Directory |
| IPsec-Auth-On-Rekey | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Backup-Server-List | Y | Y | Y | String | Single | Server addresses (space delimited) |
| IPsec-Backup-Servers | Y | Y | Y | String | Single | 1 = Use client-configured list<br>2 = Disabled and clear client list<br>3 = Use backup server list |
| IPsec-Client-Firewall-Filter-Name | Y | | | String | Single | Specifies the name of the filter to be pushed to the client as firewall policy. |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| IPsec-Client-Firewall-Filter-Optional | Y | Y | Y | Integer | Single | 0 = Required<br>1 = Optional |
| IPsec-Default-Domain | Y | Y | Y | String | Single | Specifies the single default domain name to send to the client (1 - 255 characters). |
| IPsec-Extended-Auth-On-Rekey | | Y | Y | String | Single | String |
| IPsec-IKE-Peer-ID-Check | Y | Y | Y | Integer | Single | 1 = Required<br>2 = If supported by peer certificate<br>3 = Do not check |
| IPsec-IP-Compression | Y | Y | Y | Integer | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Mode-Config | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Over-UDP | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| IPsec-Over-UDP-Port | Y | Y | Y | Integer | Single | 4001 - 49151; The default is 10000. |
| IPsec-Required-Client-Firewall-Capability | Y | Y | Y | Integer | Single | 0 = None<br>1 = Policy defined by remote FW Are-You-There (AYT)<br>2 = Policy pushed CPP<br>4 = Policy from server |
| IPsec-Sec-Association | Y | | | String | Single | Name of the security association |
| IPsec-Split-DNS-Names | Y | Y | Y | String | Single | Specifies the list of secondary domain names to send to the client (1 - 255 characters). |
| IPsec-Split-Tunneling-Policy | Y | Y | Y | Integer | Single | 0 = Tunnel everything<br>1 = Split tunneling<br>2 = Local LAN permitted |
| IPsec-Split-Tunnel-List | Y | Y | Y | String | Single | Specifies the name of the network or access list that describes the split tunnel inclusion list. |
| IPsec-Tunnel-Type | Y | Y | Y | Integer | Single | 1 = LAN-to-LAN<br>2 = Remote access |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| L2TP-Encryption | Y | | | Integer | Single | Bitmap:<br>1 = Encryption required<br>2 = 40 bit<br>4 = 128 bits<br>8 = Stateless-Req<br>15 = 40/128-Encr/Stateless-Req |
| L2TP-MPPC-Compression | Y | | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| MS-Client-Subnet-Mask | Y | Y | Y | String | Single | An IP address |
| PFS-Required | Y | Y | Y | Boolean | Single | 0 = No<br>1 = Yes |
| Port-Forwarding-Name | Y | Y | | String | Single | Name string (for example, “Corporate-Apps”) |
| PPTP-Encryption | Y | | | Intger | Single | Bitmap:<br>1 = Encryption required<br>2 = 40 bit<br>4 = 128 bits<br>8 = Stateless-Req<br>Example:<br>15 = 40/128-Encr/Stateless-Req |
| PPTP-MPPC-Compression | Y | | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| Primary-DNS | Y | Y | Y | String | Single | An IP address |
| Primary-WINS | Y | Y | Y | String | Single | An IP address |
| Privilege-Level | | | | Integer | Single | For usernames, 0 - 15 |
| Required-Client-Firewall-Vendor-Code | Y | Y | Y | Integer | Single | 1 = Cisco Systems (with Cisco Integrated Client)<br>2 = Zone Labs<br>3 = NetworkICE<br>4 = Sygate<br>5 = Cisco Systems (with Cisco Intrusion Prevention Security Agent) |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| Required-Client-Firewall-Description | Y | Y | Y | String | Single | — |
| Required-Client-Firewall-Product-Code | Y | Y | Y | Integer | Single | Cisco Systems Products:<br>1 = Cisco Intrusion Prevention Security Agent or Cisco Integrated Client (CIC)<br>Zone Labs Products:<br>1 = Zone Alarm<br>2 = Zone AlarmPro<br>3 = Zone Labs Integrity<br>NetworkICE Product:<br>1 = BlackIce Defender/Agent<br>Sygate Products:<br>1 = Personal Firewall<br>2 = Personal Firewall Pro<br>3 = Security Agent |
| Require-HW-Client-Auth | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| Require-Individual-User-Auth | Y | Y | Y | Integer | Single | 0 = Disabled<br>1 = Enabled |
| Secondary-DNS | Y | Y | Y | String | Single | An IP address |
| Secondary-WINS | Y | Y | Y | String | Single | An IP address |
| SEP-Card-Assignment | | | | Integer | Single | Not used |
| Simultaneous-Logins | Y | Y | Y | Integer | Single | 0 - 2147483647 |
| Strip-Realm | Y | Y | Y | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| TACACS-Authtype | Y | Y | Y | Integer | Single | — |
| TACACS-Privilege-Level | Y | Y | Y | Integer | Single | — |
| Tunnel-Group-Lock | | Y | Y | String | Single | Name of the tunnel group or “none” |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| Tunneling-Protocols | Y | Y | Y | Integer | Single | 1 = PPTP<br>2 = L2TP<br>4 = IPSec (IKEv1)<br>8 = L2TP/IPSec<br>16 = WebVPN<br>32 = SVC<br>64 = IPsec (IKEv2)<br>8 and 4 are mutually exclusive<br>(0 - 11, 16 - 27, 32 - 43, 48 - 59 are legal values). |
| Use-Client-Address | Y | | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| User-Auth-Server-Name | Y | | | String | Single | IP address or hostname |
| User-Auth-Server-Port | Y | Y | Y | Integer | Single | Port number for server protocol |
| User-Auth-Server-Secret | Y | | | String | Single | Server password |
| WebVPN-ACL-Filters | | Y | | String | Single | Webtype access list name |
| WebVPN-Apply-ACL-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled<br>With Version 8.0 and later, this attribute is not required. |
| WebVPN-Citrix-Support-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled<br>With Version 8.0 and later, this attribute is not required. |
| WebVPN-Enable-functions | | | | Integer | Single | Not used - deprecated |
| WebVPN-Exchange-Server-Address | | | | String | Single | Not used - deprecated |
| WebVPN-Exchange-Server-NETBIOS-Name | | | | String | Single | Not used - deprecated |
| WebVPN-File-Access-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| WebVPN-File-Server-Browsing-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-File-Server-Entry-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-Forwarded-Ports | | Y | | String | Single | Port-forward list name |
| WebVPN-Homepage | Y | Y | | String | Single | A URL such as http://www.example.com |
| WebVPN-Macro-Substitution-Value1 | Y | Y | | String | Single | Refer to Configure an External AAA Server for VPNGuide Chapter in Cisco ASA Series VPN CLI Configuration Guide for examples at the following URL:<br>https://www.cisco.com/c/en/us/support/security/adaptive-security-appliance-asa-software/products-installation-and-configuration-guides-list.html |
| WebVPN-Macro-Substitution-Value2 | Y | Y | | String | Single | Refer to Configure an External AAA Server for VPNGuide Chapter in Cisco ASA Series VPN CLI Configuration Guide for examples at the following URL:<br>https://www.cisco.com/c/en/us/support/security/adaptive-security-appliance-asa-software/products-installation-and-configuration-guides-list.html |
| WebVPN-Port-Forwarding-Auto-Download-Enable | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-Port-Forwarding-Enable | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-Port-Forwarding-Exchange-Proxy-Enable | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-Port-Forwarding-HTTP-Proxy-Enable | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-Single-Sign-On-Server-Name | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-SVC-Client-DPD | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |

| Attribute Name | VPN 3000 | ASA | PIX | Syntax/Type | Single or Multi-valued | Possible Values |
|---|---|---|---|---|---|---|
| WebVPN-SVC-Compression | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-SVC-Enable | Y | Y | | Boolean | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-SVC-Gateway-DPD | Y | Y | | Integer | Single | 0 = Disabled<br>n = Dead peer detection value in seconds (30 - 3600) |
| WebVPN-SVC-Keepalive | Y | Y | | Integer | Single | 0 = Disabled<br>n = Keepalive value in seconds (15 - 600) |
| WebVPN-SVC-KeepEnable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-SVC-Rekey-Method | Y | Y | | Integer | Single | 0 = None<br>1 = SSL<br>2 = New tunnel<br>3 = Any (sets to SSL) |
| WebVPN-SVC-Rekey-Period | Y | Y | | Integer | Single | 0 = Disabled<br>n = Retry period in minutes<br>(4 - 10080) |
| WebVPN-SVC-RequireEnable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-URL-Entry-Enable | Y | Y | | Integer | Single | 0 = Disabled<br>1 = Enabled |
| WebVPN-URL-List | | Y | | String | Single | URL list name |

## URL Types Supported in ACLs

The URL may be a partial URL, contain wildcards for the server, or include a port.

The following URL types are supported.

| | | | |
|---|---|---|---|
| any All URLs | https:// | post:// | ssh:// |
| cifs:// | ica:// | rdp:// | telnet:// |
| citrix:// | imap4:// | rdp2:// | vnc:// |

| | | | |
|---|---|---|---|
| citrixs:// | ftp:// | smart-tunnel:// | |
| http:// | pop3:// | smtp:// | |

## Guidelines for Using Cisco-AV Pairs (ACLs)

- Use Cisco-AV pair entries with the ip:inacl# prefix to enforce access lists for remote IPsec and SSL VPN Client (SVC) tunnels.
- Use Cisco-AV pair entries with the webvpn:inacl# prefix to enforce access lists for SSL VPN clientless (browser-mode) tunnels.
- For webtype ACLs, you do not specify the source because the ASA is the source.

***Table 16: ASA-Supported Tokens***

| Token | Syntax Field | Description |
|---|---|---|
| ip:inacl# *Num* = | N/A (Identifier) | (Where *Num* is a unique integer.) Starts all AV pair access control lists. Enforces access lists for remote IPsec and SSL VPN (SVC) tunnels. |
| webvpn:inacl# *Num* = | N/A (Identifier) | (Where *Num* is a unique integer.) Starts all clientless SSL AV pair access control lists. Enforces access lists for clientless (browser-mode) tunnels. |
| deny | Action | Denies action. (Default) |
| permit | Action | Allows action. |
| icmp | Protocol | Internet Control Message Protocol (ICMP) |
| 1 | Protocol | Internet Control Message Protocol (ICMP) |
| IP | Protocol | Internet Protocol (IP) |
| 0 | Protocol | Internet Protocol (IP) |
| TCP | Protocol | Transmission Control Protocol (TCP) |
| 6 | Protocol | Transmission Control Protocol (TCP) |
| UDP | Protocol | User Datagram Protocol (UDP) |
| 17 | Protocol | User Datagram Protocol (UDP) |
| any | Hostname | Rule applies to any host. |
| host | Hostname | Any alpha-numeric string that denotes a hostname. |
| log | Log | When the event occurs, a filter log message appears. (Same as permit and log or deny and log.) |
| lt | Operator | Less than value |
| gt | Operator | Greater than value |

| Token | Syntax Field | Description |
|---|---|---|
| eq | Operator | Equal to value |
| neq | Operator | Not equal to value |
| range | Operator | Inclusive range. Should be followed by two values. |

## Cisco AV Pair Attribute Syntax

The Cisco Attribute Value (AV) pair (ID Number 26/9/1) can be used to enforce access lists from a RADIUS server (like Cisco ACS), or from an LDAP server via an LDAP attribute map.

The syntax of each Cisco-AV-Pair rule is as follows:

*[Prefix] [Action] [Protocol] [Source] [Source Wildcard Mask] [Destination] [Destination Wildcard Mask] [Established] [Log] [Operator] [Port]*

***Table 17: AV-Pair Attribute Syntax Rules***

| Field | Description |
|---|---|
| Action | Action to perform if the rule matches a deny or a permit. |
| Destination | Network or host that receives the packet. Specify it as an IP address, a hostname, or the **any** keyword. If using an IP address, the source wildcard mask must follow. |
| Destination Wildcard Mask | The wildcard mask that applies to the destination address. |
| Log | Generates a FILTER log message. You must use this keyword to generate events of severity level 9. |
| Operator | Logic operators: greater than, less than, equal to, not equal to. |
| Port | The number of a TCP or UDP port in the range of 0 - 65535. |
| Prefix | A unique identifier for the AV pair (for example: ip:inacl#1= for standard access lists or webvpn:inacl# = for clientless SSL VPN access lists). This field only appears when the filter has been sent as an AV pair. |
| Protocol | Number or name of an IP protocol. Either an integer in the range of 0 - 255 or one of the following keywords: **icmp** , **igmp** , **ip** , **tcp** , **udp** . |
| Source | Network or host that sends the packet. Specify it as an IP address, a hostname, or the **any** keyword. If using an IP address, the source wildcard mask must follow. This field does not apply to Clientless SSL VPN because the ASA has the role of the source or proxy. |
| Source Wildcard Mask | The wildcard mask that applies to the source address. This field does not apply to Clientless SSL VPN because the ASA has the role of the source or proxy. |

## Cisco AV Pairs ACL Examples

This section shows examples of Cisco AV pairs and describes the permit or deny actions that result.



**Note** Each ACL # in inacl# must be unique. However, they do not need to be sequential (for example, 1, 2, 3, 4). That is, they could be 5, 45, 135.

***Table 18: Examples of Cisco AV Pairs and Their Permitting or Denying Action***

| Cisco AV Pair Example | Permitting or Denying Action |
|---|---|
| `ip:inacl#1=deny ip 10.155.10.0 0.0.0.255 10.159.2.0 0.0.0.255 log` | Allows IP traffic between the two hosts using a full tunnel IPsec or SSL VPN client. |
| `ip:inacl#2=permit TCP any host 10.160.0.1 eq 80 log` | Allows TCP traffic from all hosts to the specific host on port 80 only using a full tunnel IPsec or SSL VPN client. |
| `webvpn:inacl#1=permit url http://www.example.comwebvpn:inacl#2=deny url smtp://serverwebvpn:inacl#3=permit url cifs://server/share` | Allows clientlessSSL VPN traffic to the URL specified, denies SMTP traffic to a specific server, and allows file share access (CIFS) to the specified server. |
| `webvpn:inacl#1=permit tcp 10.86.1.2 eq 2222 logwebvpn:inacl#2=deny tcp 10.86.1.2 eq 2323 log` | Denies Telnet access and permits SSH access on non-default ports 2323 and 2222, respectively, or other application traffic flows using these ports for clientless SSL VPN. |
| `webvpn:inacl#1=permit url ssh://10.86.1.2webvpn:inacl#35=permit tcp 10.86.1.5 eq 22 logwebvpn:inacl#48=deny url telnet://10.86.1.2webvpn:inacl#100=deny tcp 10.86.1.6 eq 23` | Allows clientless SSL VPN SSH access to default port 22 and denies Telnet access to port 23, respectively. This example assumes that you are using Telnet or SSH Java plug-ins enforced by these ACLs. |

# Active Directory/LDAP VPN Remote Access Authorization Examples

This section presents example procedures for configuring authentication and authorization on the ASA using the Microsoft Active Directory server. It includes the following topics:


- Policy Enforcement of User-Based Attributes, on page 308
- Enforce Static IP Address Assignment for Secure Client Tunnels, on page 309
- Enforce Dial-in Allow or Deny Access, on page 311
- Enforce Logon Hours and Time-of-Day Rules, on page 313


Other configuration examples available on Cisco.com include the following TechNotes.

- ASA/PIX: Mapping VPN Clients to VPN Group Policies Through LDAP Configuration Example

- PIX/ASA 8.0: Use LDAP Authentication to Assign a Group Policy at Login

# Policy Enforcement of User-Based Attributes

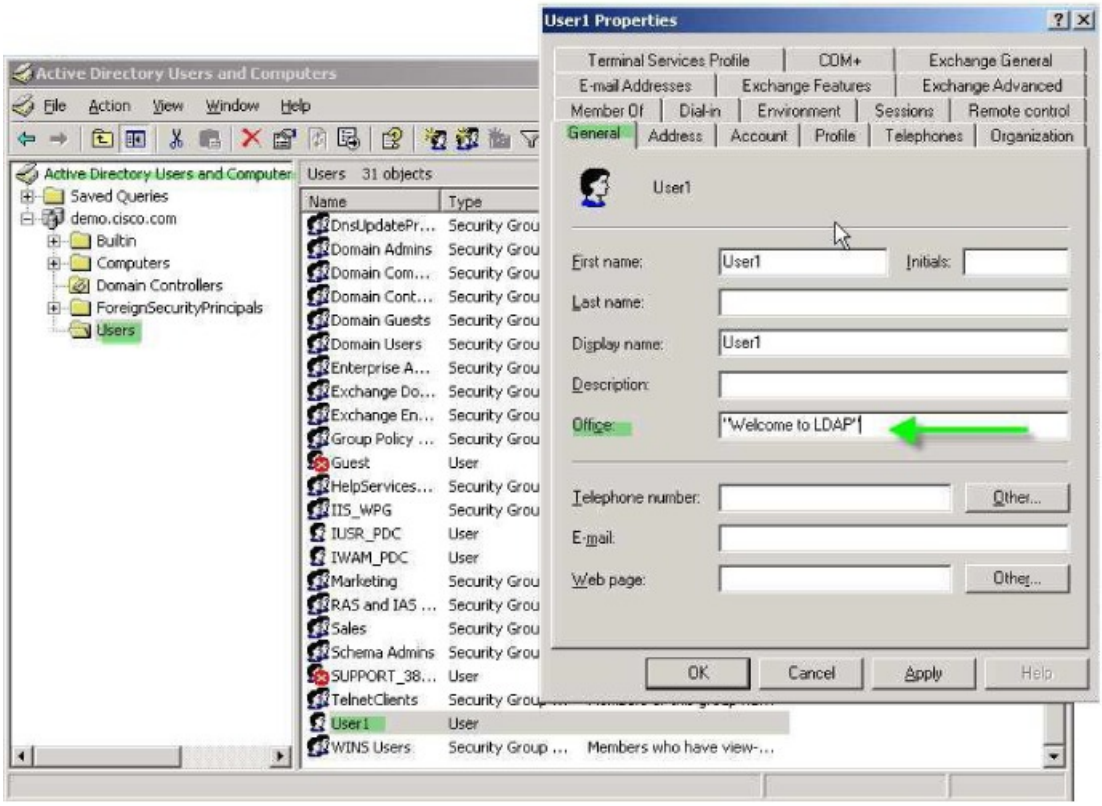
This example displays a simple banner to the user, showing how you can map any standard LDAP attribute to a well-known Vendor-Specific Attribute (VSA), and you can map one or more LDAP attribute(s) to one or more Cisco LDAP attributes. It applies to any connection type, including the IPsec VPN client and Secure Client.

To enforce a simple banner for a user who is configured on an AD LDAP server use the Office field in the General tab to enter the banner text. This field uses the attribute named physicalDeliveryOfficeName. On the ASA, create an attribute map that maps physicalDeliveryOfficeName to the Cisco attribute Banner1.

During authentication, the ASA retrieves the value of physicalDeliveryOfficeName from the server, maps the value to the Cisco attribute Banner1, and displays the banner to the user.

### Procedure

**Step 1** Right-click the username, open the Properties dialog box then the **General** tab and enter banner text in the Office field, which uses the AD/LDAP attribute physicalDeliveryOfficeName.




**Step 2** Create an LDAP attribute map on the ASA.

Create the map Banner and map the AD/LDAP attribute physicalDeliveryOfficeName to the Cisco attribute Banner1:

```
hostname(config)# ldap attribute-map Banner
hostname(config-ldap-attribute-map)# map-name physicalDeliveryOfficeName Banner1
```

**Step 3** Associate the LDAP attribute map to the AAA server.

Enter the aaa server host configuration mode for the host 10.1.1.2 in the AAA server group MS_LDAP, and associate the attribute map Banner that you previously created:

```
hostname(config)# aaa-server MS_LDAP host 10.1.1.2
hostname(config-aaa-server-host)# ldap-attribute-map Banner
```

**Step 4** Test the banner enforcement.

# Enforce Static IP Address Assignment for Secure Client Tunnels

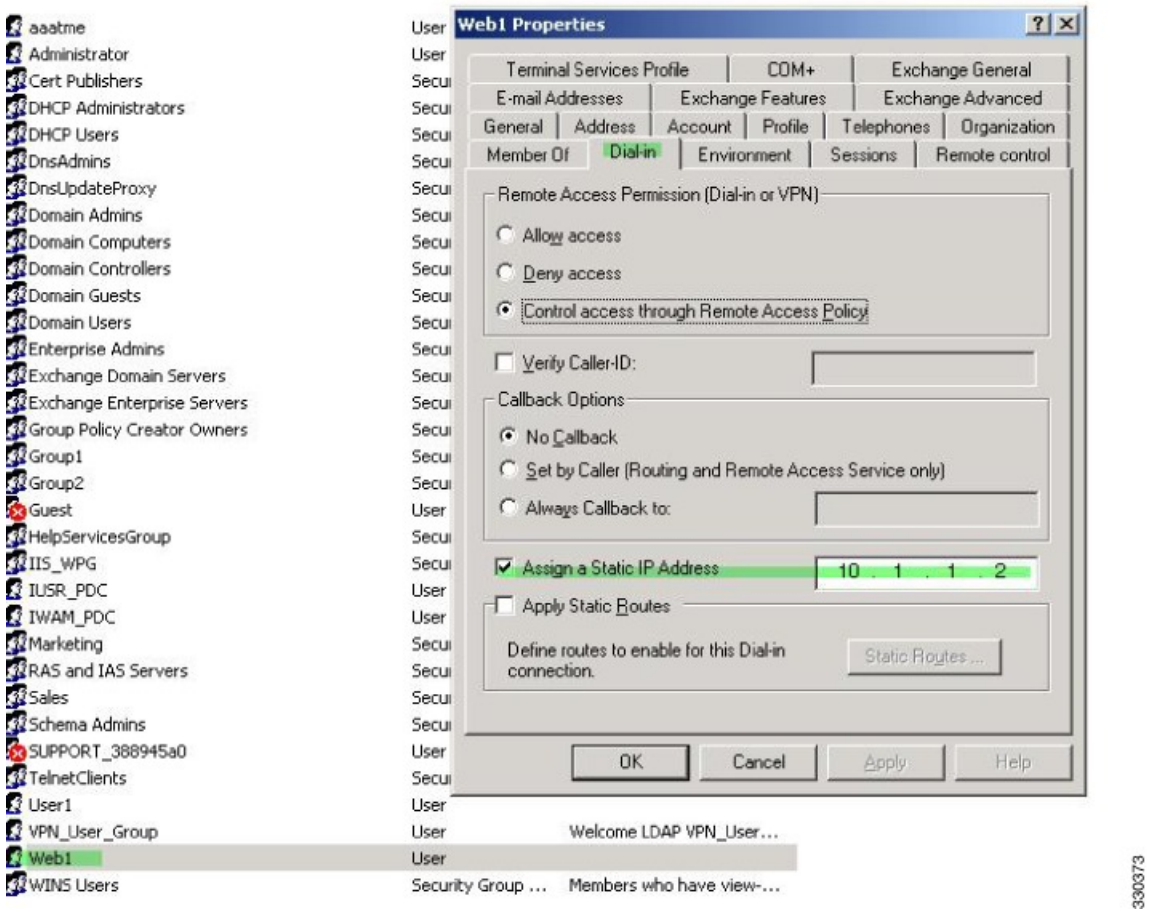
This example applies to full-tunnel clients, such as the IPsec client and the SSL VPN clients.

To enforce static Secure Client static IP assignments configure the Secure Client user Web1 to receive a static IP address, enter the address in the Assign Static IP Address field of the Dialin tab on the AD LDAP server (this field uses the msRADIUSFramedIPAddress attribute), and create an attribute map that maps this attribute to the Cisco attribute IETF-Radius-Framed-IP-Address.

During authentication, the ASA retrieves the value of msRADIUSFramedIPAddress from the server, maps the value to the Cisco attribute IETF-Radius-Framed-IP-Address, and provides the static address to User1.

## Procedure

**Step 1** Right-click the username, open the Properties dialog box then the **Dial-in** tab, check the **Assign Static IP Address** check box, and enter an IP address of 10.1.1.2.




**Step 2** Create an attribute map for the LDAP configuration shown.

Map the AD attribute msRADIUSFramedIPAddress used by the Static Address field to the Cisco attribute IETF-Radius-Framed-IP-Address:

```
hostname(config)# ldap attribute-map static_address
hostname(config-ldap-attribute-map)# map-name msRADIUSFramedIPAddress
IETF-Radius-Framed-IP-Address
```

**Step 3** Associate the LDAP attribute map to the AAA server.

Enter the aaa server host configuration mode for the host 10.1.1.2 in the AAA server group MS_LDAP, and associates the attribute map static_address that you previously created in:

```
hostname(config)# aaa-server MS_LDAP host 10.1.1.2
hostname(config-aaa-server-host)# ldap-attribute-map static_address
```

**Step 4** Verify that the **vpn-address-assignment** command is configured to specify AAA by viewing this part of the configuration:

```
hostname(config)# show run all vpn-addr-assign
vpn-addr-assign aaa   << Make sure this is configured >>
```

```
no vpn-addr-assign dhcp
vpn-addr-assign local
hostname(config)#
```

**Step 5** Establish a connection to the ASA with the Secure Client. Observe that the user receives the IP address configured on the server and mapped to the ASA.

**Step 6** Use the **show vpn-sessiondb svc** command to view the session details and verify the address assigned:

```
hostname# show vpn-sessiondb svc

Session Type: SVC
Username     : web1                   Index        : 31
Assigned IP  : 10.1.1.2              Public IP    : 10.86.181.70
Protocol     : Clientless SSL-Tunnel DTLS-Tunnel
Encryption   : RC4 AES128             Hashing      : SHA1
Bytes Tx     : 304140                 Bytes Rx     : 470506
Group Policy : VPN_User_Group         Tunnel Group : Group1_TunnelGroup
Login Time   : 11:13:05 UTC Tue Aug 28 2007
Duration     : 0h:01m:48s
NAC Result   : Unknown
VLAN Mapping : N/A                    VLAN         : none
```

# Enforce Dial-in Allow or Deny Access

This example creates an LDAP attribute map that specifies the tunneling protocols allowed by the user. You map the allow access and deny access settings on the Dialin tab to the Cisco attribute Tunneling-Protocol, which supports the following bitmap values:

| Value | Tunneling Protocol |
|---|---|
| 1 | PPTP |
| 2 | L2TP |
| 4 | IPsec (IKEv1) |
| 8 | L2TP/IPsec |
| 16 | Clientless SSL |
| 32 | SSL client—Secure Client or SSL VPN client |
| 64 | IPsec (IKEv2) |

[1] (1) IPsec and L2TP over IPsec are not supported simultaneously. Therefore, the values 4 and 8 are mutually exclusive.

[2] (2) See note 1.

Use this attribute to create an Allow Access (TRUE) or a Deny Access (FALSE) condition for the protocols, and enforce the method for which the user is allowed access.

See Tech Note ASA/PIX: Mapping VPN Clients to VPN Group Policies Through LDAP Configuration Example for another example of enforcing dial-in allow access or deny access.

### Procedure

**Step 1** Right-click the username, open the Properties dialog box then the **Dial-in** tab, and click the Allow Access radio button.




**Note**
If you choose the Control access through the Remote Access Policy option, then a value is not returned from the server, and the permissions that are enforced are based on the internal group policy settings of the ASA.

**Step 2** Create an attribute map to allow both an IPsec and Secure Client connection, but deny a clientless SSL connection.

a) Create the map tunneling_protocols:

```
hostname(config)# ldap attribute-map tunneling_protocols
```

b) Map the AD attribute msNPAllowDialin used by the Allow Access setting to the Cisco attribute Tunneling-Protocols:

```
hostname(config-ldap-attribute-map)# map-name msNPAllowDialin Tunneling-Protocols
```

c) Add map values:

```
hostname(config-ldap-attribute-map)# map-value msNPAllowDialin FALSE 48
hostname(config-ldap-attribute-map)# map-value msNPAllowDialin TRUE 4
```

**Step 3** Associate the LDAP attribute map to the AAA server.

a) Enter the aaa server host configuration mode for the host 10.1.1.2 in the AAA server group MS_LDAP:

```
hostname(config)# aaa-server MS_LDAP host 10.1.1.2
```

b) Associates the attribute map tunneling_protocols that you created:

```
hostname(config-aaa-server-host)# ldap-attribute-map tunneling_protocols
```

**Step 4** Verify that the attribute map works as configured.

Try connections using clientless SSL, the user should be informed that an unauthorized connection mechanism was the reason for the failed connection. The IPsec client should connect because IPsec is an allowed tunneling protocol according to the attribute map.

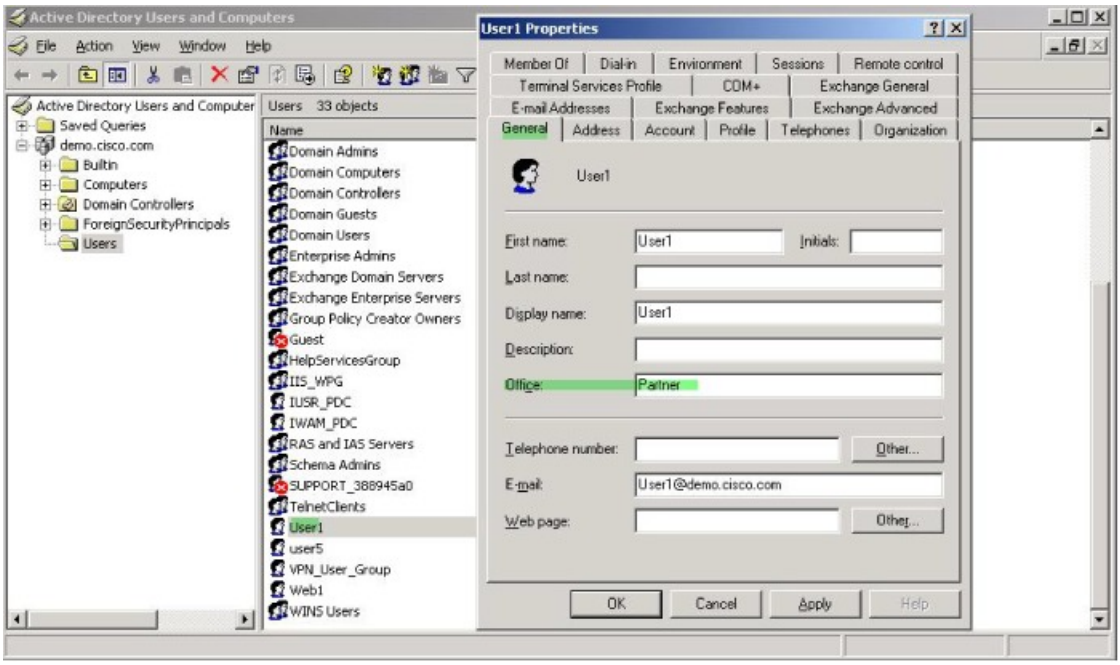
# Enforce Logon Hours and Time-of-Day Rules

The following example shows how to configure and enforce the hours that a clientless SSL user (such as a business partner) is allowed to access the network.

On the AD server, use the Office field to enter the name of the partner, which uses the physicalDeliveryOfficeName attribute. Then we create an attribute map on the ASA to map that attribute to the Cisco attribute Access-Hours. During authentication, the ASA retrieves the value of physicalDeliveryOfficeName and maps it to Access-Hours.

## Procedure

**Step 1** Select the user, right-click **Properties**, and open the **General** tab:




**Step 2** Create an attribute map.

Create the attribute map access_hours and map the AD attribute physicalDeliveryOfficeName used by the Office field to the Cisco attribute Access-Hours.

```
hostname(config)# ldap attribute-map access_hours
hostname(config-ldap-attribute-map)# map-name physicalDeliveryOfficeName Access-Hours
```

**Step 3** Associate the LDAP attribute map to the AAA server.

Enter the aaa server host configuration mode for host 10.1.1.2 in the AAA server group MS_LDAP and associate the attribute map access_hours that you created.

```
hostname(config)# aaa-server MS_LDAP host 10.1.1.2
hostname(config-aaa-server-host)# ldap-attribute-map access_hours
```

**Step 4** Configure time ranges for each value allowed on the server.

Configure Partner access hours from 9am to 5pm Monday through Friday:

```
hostname(config)# time-range Partner
hostname(config-time-range)# periodic weekdays 09:00 to 17:00
```