ACOS 6.0.7
IP Security Configuration Guide

April, 2025

Information in this document is subject to change without notice.

PATENT PROTECTION

A10 Networks, Inc. products are protected by patents in the U.S. and elsewhere. The following website is provided to satisfy the virtual patent marking provisions of various jurisdictions including the virtual patent marking provisions of the America Invents Act. A10 Networks, Inc. products, including all Thunder Series products, are protected by one or more of U.S. patents and patents pending listed at: a10-virtual-patent-marking.

TRADEMARKS

A10 Networks, Inc. trademarks are listed at: a10-trademarks

CONFIDENTIALITY

This document contains confidential materials proprietary to A10 Networks, Inc. This document and information and ideas herein may not be disclosed, copied, reproduced or distributed to anyone outside A10 Networks, Inc. without prior written consent of A10 Networks, Inc.

DISCLAIMER

This document does not create any express or implied warranty about A10 Networks, Inc. or about its products or services, including but not limited to fitness for a particular use and non-infringement. A10 Networks, Inc. has made reasonable efforts to verify that the information contained herein is accurate, but A10 Networks, Inc. assumes no responsibility for its use. All information is provided "as-is." The product specifications and features described in this publication are based on the latest information available; however, specifications are subject to change without notice, and certain features may not be available upon initial product release. Contact A10 Networks, Inc. for current information regarding its products or services. A10 Networks, Inc. products and services are subject to A10 Networks, Inc. standard terms and conditions.

ENVIRONMENTAL CONSIDERATIONS

Some electronic components may possibly contain dangerous substances. For information on specific component types, please contact the manufacturer of that component. Always consult local authorities for regulations regarding proper disposal of electronic components in your area.

FURTHER INFORMATION

For additional information about A10 products, terms and conditions of delivery, and pricing, contact your nearest A10 Networks, Inc. location, which can be found by visiting www.a10networks.com.

Table of Contents

Getting Started

The ACOS software supports Internet Protocol Security (IPsec) Virtual Private Network (VPN). IPsec VPN is a suite of protocols that secures private traffic over a public network. This chapter provides an overview of the IPsec VPN.

To protect communications, organizations need to encrypt data at high speeds and scale-out VPN tunnel capacity on-demand. ACOS uses IPsec VPN to secure high-capacity links at unparalleled speeds.

IPsec, as defined in RFC 4301, provides a means by which to ensure the authenticity, integrity, and confidentiality of data at the network layer of the Open System Interconnection (OSI) stack. IPsec is application-agnostic; any IP application's network traffic is secured without the need for application changes.

NOTE: The current release supports route-based and policy-based IP Encapsulating Security Payload (ESP) protocol (RFC 4303), in network tunnel-mode operation for IPv4 and IPv6 traffic. VPN tunnels are established using the Internet Key Exchange (IKE) protocol version 1 or 2.

The following topics are covered:

Licensing

A license needs to be installed through A10's Global License Manager (GLM) in order to use the ACOS Internet Protocol Security (IPsec) Virtual Private Network (VPN). The A10 Thunder device needs to be connected to the internet in order to periodically validate the license.

NOTE: By default, from ACOC 4.1.4-GR1-P2 release onwards the Convergent Firewall (CFW) platforms requires a license to enable IPsec functionality.

The following topics are covered:

License Activation (Offline Mode)

To configure or import the activation license key using the ACOS **GUI**, running on the ACOS 4.1.4-GR1-P2 or later, perform the following:

- Navigate to **System** > **Admin** tab for entering ACOS activation key from the GUI.
- Click on the **Licensing** tab. The Licensing window is displayed.

Select one of the following ways to enter the activation:

- **Copy/Paste** — Copy the text of the activation key license and paste it into the blank field. Then, click Submit.
- **Upload Text File** — If the activation key license is saved as a text file, you can click Choose File, navigate to the text file, and then click Upload File.
- **Upload License file from Remote Server** — Click Remote Import to import a license file from a remote server.

To configure or import the activation license key using the ACOS **CLI**, running on the ACOS 4.1.4GR1-P2 or later, perform the following:

1. Access the Privileged EXEC (enable) level or any configuration level of the CLI.
2. Save the activation key license file sent by A10 Networks onto a server that can be locally accessed over the network by your appliance.
3. Enter the following command to install the license:

   ```
   import glm-license file-name url
   ```

   The file-name is the name of the activation key license file received from A10 Networks. The URL specifies the file transfer protocol, the username (if required), and directory path.

   You can enter the entire URL on the command line or press **Enter** to display a prompt for each part of the URL. If you enter the entire URL and a password is required, you will still be prompted for the password.

   To enter the entire URL:

   - tftp://host/file
   - ftp://[user@]host[:port]/file
   - scp://[user@]host/file
   - http://[user@]host/file
   - https://[user@]host/file
   - sftp://[user@]host/file
4. Close your CLI session.
5. Open a new CLI session.
6. Access the Privileged EXEC (enable) level or any configuration level of the CLI.
7. Enter the **`show license-info`** command to verify the activation key license installation.

For example,

```
TH7650-1#show license-info
Host ID         : 4BD78D258E47EBA6E0E0458E8CAA9BC9183C72D4
USB ID          : Not Available
Billing Serials: vTh5c3b2c9840000, vTh64420a9070000
Token           : Not Available
Product         : CFW
Platform        : Thunder Series Unified Application Service Gateway
```

```
Burst           : Disabled
GLM Ping Interval In Hours : 24
-------------------------------------------------------------------------------
----------
Enabled Licenses          Expiry Date (UTC)                Notes
-------------------------------------------------------------------------------
----------
SLB                       None
CGN                       None
GSLB                      None
RC                        None
DAF                       None
WAF                       None
SSLI                      None
DCFW                      None
GIFW                      None
URLF                      None
AAM                       None
FP                        None
WEBROOT                   N/A             Requires an additional Webroot
license.
THREATSTOP                N/A             Requires an additional
ThreatSTOP license.
QOSMOS                    N/A             Requires an additional QOSMOS
license.
WEBROOT_TI                N/A             Requires an additional Webroot
Threat Intel license.
CYLANCE                   N/A             Requires an additional Cylance
license.
IPSEC_VPN                 None
```

NOTE: Reboot ACOS after verification to ensure the full functionality of ACOS features.

CLI Example

- Thunder- AX series

1. Log onto the CLI, access the Privileged EXEC level, and display the license host ID:

```
login as: admin
Using keyboard-interactive authentication.
Password:
Last login: Mon May 1 21:23:14 2020 from 172.16.137.157
ACOS system is ready now.
[type ? for help]
TH3030S-AX3>enable
Password:
TH3030S-AX3#show license uid
A0C774C33831F0A5FB9961EA5EDCF31330FB91A6
```

2. Imports and installs the license.

```
TH3030S-AX3#import glm-license IPsec_33395-10.16.21.108.txt use-mgmt-
port scp://
root:password@10.16.21.113/home/deyue/IPsec_33395-10.16.21.108.txt
License successfully updated, please log out and log back in to access
license features
Done.
TH3030S-AX3#
```

3. Close the existing CLI session and open a new CLI session.

   The following **`show license-info`** command verifies license installation and lists the set of licenses and information.

   For example,

```
TH3030S-AX3#show license-info
Host ID         : 49E78D258E47EBA6E0E0458E8CAA9BC9183C7AEB
USB ID          : Not Available
Billing Serials: vTh3e3b2c9840000, vTh64420a9060000
Token           : Not Available
Product         : CFW
Platform        : Thunder Series Unified Application Service Gateway
Burst           : Disabled
GLM Ping Interval In Hours : 24
------------------------------------------------------------------------
-------------
Enabled Licenses          Expiry Date (UTC)                Notes
```

```
--------------------------------------------------------------------------
-------------
SLB                          None
CGN                          None
GSLB                         None
RC                           None
DAF                          None
WAF                          None
SSLI                         None
DCFW                         None
GIFW                         None
URLF                         None
AAM                          None
FP                           None
WEBROOT                      N/A               Requires an additional
Webroot license.
THREATSTOP                   N/A               Requires an additional
ThreatSTOP license.
QOSMOS                       N/A               Requires an additional
QOSMOS license.
WEBROOT_TI                   N/A               Requires an additional
Webroot Threat Intel license.
CYLANCE                      N/A               Requires an additional
Cylance license.
IPSEC_VPN                 None
```

License Activation (Online Mode)

To request and enable the license, perform the following:

1. Configure your ACOS device with a valid domain name server (DNS).

   An example configuration is provided below. Use the `show run ip` command to verify your configuration.

   ```
   ACOS(config)#ip dns primary 8.8.8.8
   ```

2. Configure the user management port interface.

   ```
   ACOS(config)#glm use-mgmt-port
   ```

3. Enable the glm requests.

```
ACOS(config)#glm enable-requests
The QOSMOS license is successfully updated. Please log out and log back
in to the ACOS device to access the license features  vThcee6ed6d60000
IPSEC_VPN License successfully updated, please log out and log back in
to access license features  vThcee6ed6d60000
```

4. Reboot the system to enable the IPsec license.

5. Enter the `#show license-info` command to validate the license information.

```
TH3030S-AX3#show license-info
Host ID        : 49E78D258E47EBA6E0E0458E8CAA9BC9183C7AEB
USB ID         : Not Available
Billing Serials: vTh3e3b2c9840000, vTh64420a9060000
Token          : Not Available
Product        : CFW
Platform       : Thunder Series Unified Application Service Gateway
Burst          : Disabled
GLM Ping Interval In Hours : 24
------------------------------------------------------------------------
-------------
Enabled Licenses          Expiry Date (UTC)                Notes
------------------------------------------------------------------------
-------------
SLB                       None
CGN                       None
GSLB                      None
RC                        None
DAF                       None
WAF                       None
SSLI                      None
DCFW                      None
GIFW                      None
URLF                      None
AAM                       None
FP                        None
WEBROOT                   N/A              Requires an additional
Webroot license.
THREATSTOP                N/A              Requires an additional
ThreatSTOP license.
```

```
QOSMOS                      N/A              Requires an additional
QOSMOS license.
WEBROOT_TI                  N/A              Requires an additional
Webroot Threat Intel license.
CYLANCE                     N/A              Requires an additional
Cylance license.
IPSEC_VPN               None
```

Perpetual License

To deploy IPsec VPN license on a prior version and to upgrade to ACOS 4.1.4-GR1-P2 or later, perform the following:

1. Obtain a license from A10 Networks sales representative.
2. Verify the license validity by signing in to the Global Licensing Manager account.

NOTE: For more information, see the *Obtaining Your Activation Key License* chapter in the Global License Manager to obtain and apply the IPsec license.

3. Take system backup. For more information see Backing Up System Information section of *System Configuration and Administration Guide*.
4. Upgrade ACOS on the device.

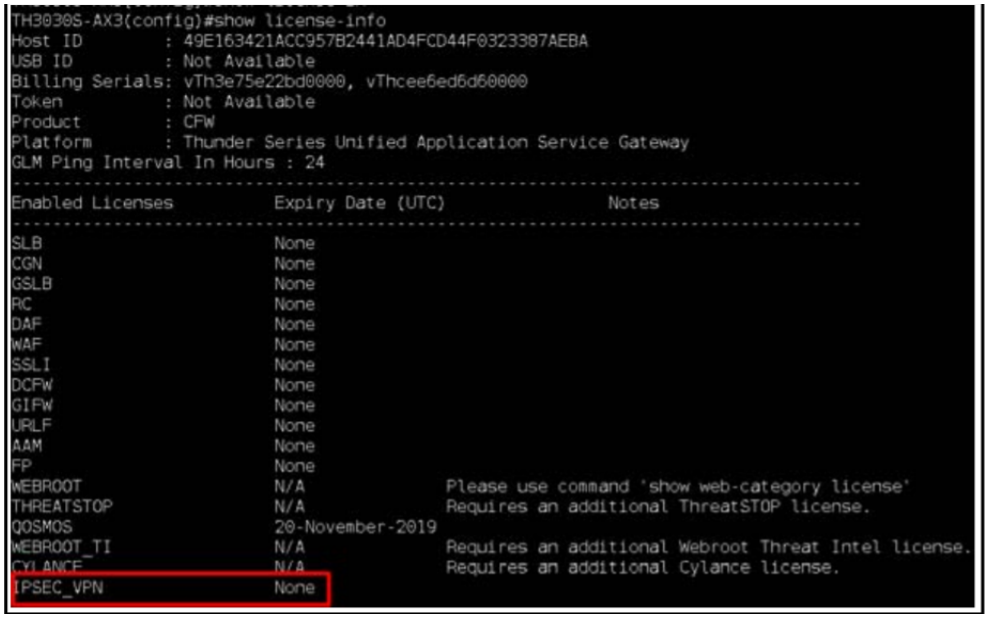
NOTE: For upgrade instructions, see the release notes for the ACOS release to which you plan to upgrade.

5. Reboot the system.

NOTE: After upgrading, upon first boot, DO NOT save the running-config startup-config with `write memory` command.

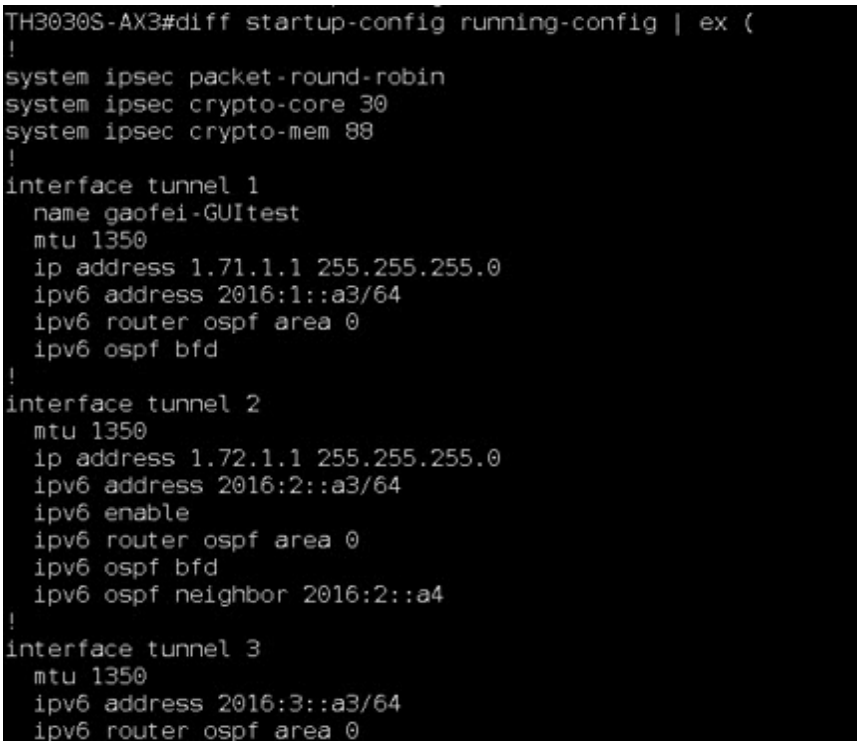
6. Enter the `#sh license-info` command.

Figure 1 : License information



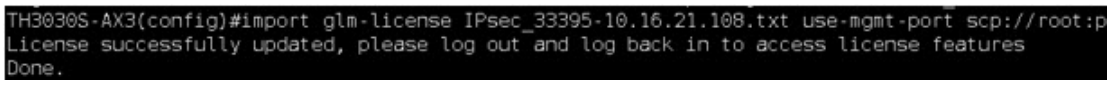
NOTE: After upgrading, at first boot, there will be no IPsec license, and IPsec related configurations are not in running-config as shown below:

Figure 2 : Interface information



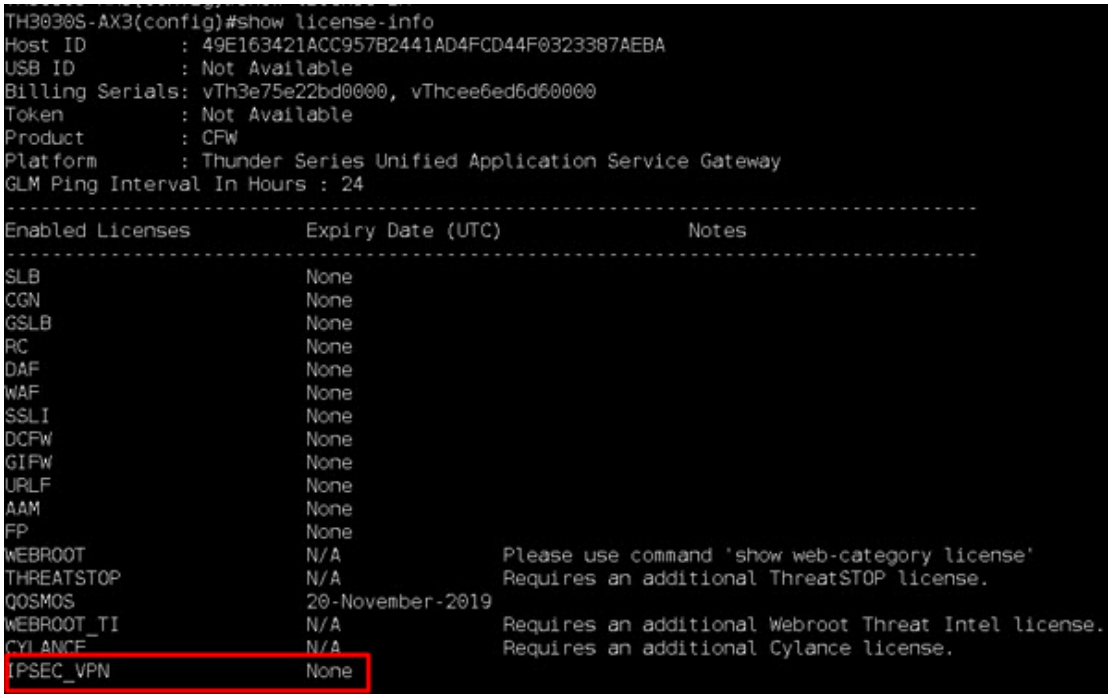
7. Import IPsec license.

Figure 3 : Import IPSec License



8. Validate the IPsec License using the `#show license-info` command.

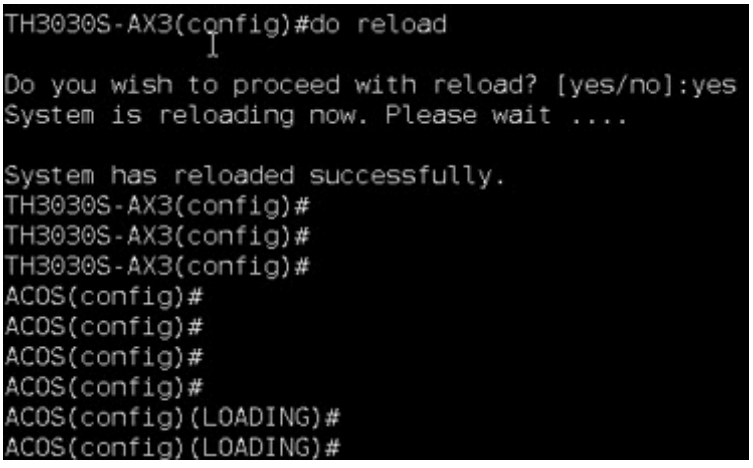
Figure 4 : License Validation



NOTE: The IPsec license is working in “None” status.

9. Reload the system using the `#do reload` command.

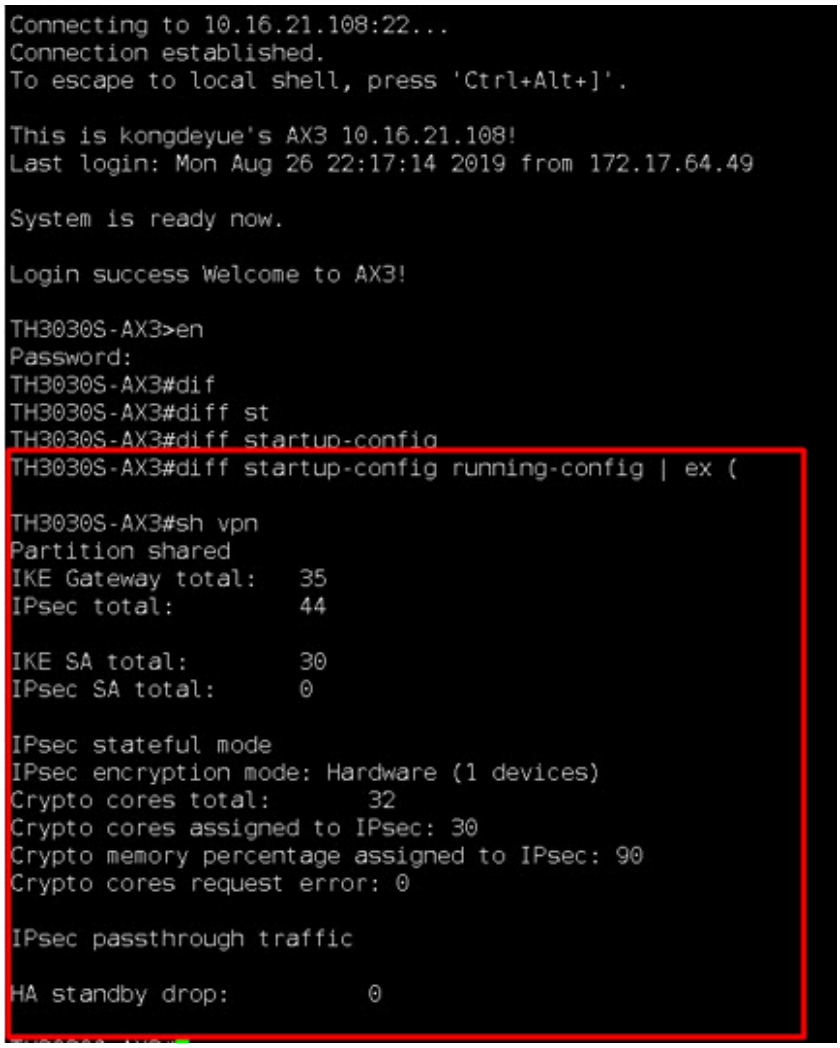
Figure 5 : Reload



NOTE: In TH3030S-AX3, system prompts to save, select “no”, then select “yes” to reload.

10. Validate the IPsec configuration by entering the `#diff startup-config running-config` command.

Figure 6 : Running Configuration



NOTE: All IPsec related configurations are in the running-config, it is the same as startup-config.

Features

Route-Based

Route-based IPsec VPN is when traffic is routed to specific VPN peers through a VPN tunnel based on the dynamic routing information that is configured on the gateways.

Policy-Based

ACOS supports policy-based IPsec VPN by passing the traffic that matches the specified Firewall rule to the specified IPsec VPN tunnel.

Traffic Selectors

Traffic selectors are used to specify the networks, protocols, and ports that pass through the IPsec tunnel. Traffic Selectors perform traffic classification based on the flow listed below:

- Local IP address, protocol port, and IP protocol number
- Remote (peer) IP address, Layer 4 transport protocol port, and IP protocol number

NOTE: If both sides of the IPsec tunnel are ACOS devices then traffic selectors are not needed in route-based VPNs. However, if the other side of the tunnel is not an ACOS device then traffic selectors may need to be configured to avoid a policy mismatch.

Static routes for the remote networks need to be added manually. These routes do not conflict with routing protocols. A maximum of one traffic selector can be configured for each VPN.

NOTE: In the current release, the traffic selector is sort of an identifier that distinguishes IPsec tunnels from each other when they are bound to the same ike-gateway.

NAT Traversal

Network Address Translation Traversal (NAT-T) is required if a device between ACOS and the peer VPN gateway is performing source or destination NAT. In this case, NAT-T must be enabled on both ACOS and the peer VPN gateway, for NAT to function properly between the two VPN gateways. When NAT-T is enabled, the ACOS device encapsulates ESP traffic inside UDP packets before sending them to the peer VPN gateway.

NOTE: When NAT-T is disabled, ESP packets use IP protocol 50. In this case, the NAT device may inappropriately apply NAT to the packets. When NAT-T is enabled, IPsec packets use UDP (protocol 17) port 4500.

Dead Peer Detection

Dead Peer Detection (DPD) is a mechanism that tests idle VPN sessions to detect connectivity issues with peer VPN gateways sooner. Without DPD enabled, the ACOS does not detect that the peer VPN gateway is down until an IPsec SA is about to expire. When the ACOS attempts to re-key, the attempts fail. Prior to this time, the ACOS continues to send packets over the tunnel; however, these packets are never received by the peer.

DPD is disabled by default. You can enable DPD at the VPN gateway configuration level. When configuring DPD, you must specify the following parameters:

- Interval
- Retries

Interval

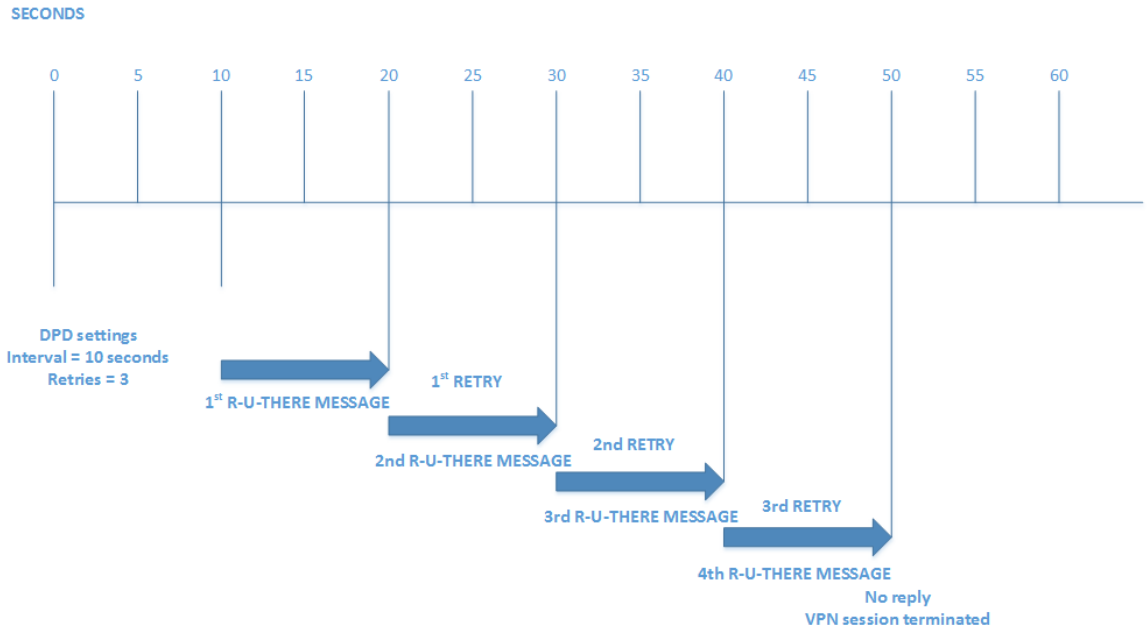
Time between DPD R-U-There messages. The interval can be 1-3600 seconds.

Retries

Maximum number of times the ACOS resends an unanswered R-U-There packet before tearing down the session. When you enable DPD, you can specify 1-10 retries.

There is no default value and both interval and retry need to be configured.

Figure 7 : Dead Peer Detection R-U-THERE Message Timeline



In the above example, the DPD interval is set to 10 seconds and the number of retries is set to 3. The connection is idle for the length of a DPD interval (10 seconds). ACOS sends an R-U-There packet to the remote gateway. ACOS waits for another interval (10 seconds) for a reply.

If there is no reply, ACOS retries by resending the R-U-There message. If there is still no reply before the last retry expires, ACOS deletes the IKE SAs and any IPsec SAs associated with the peer gateway.

Topologies

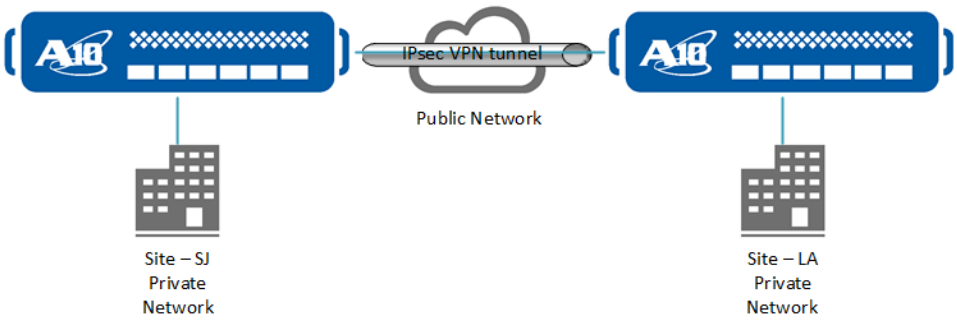
ACOS supports three IPsec VPN topologies.

The following topics are covered:

Site-to-Site

Site-to-site IPsec VPN provides security for network traffic between two private networks. Typically, these private networks are connected over the Internet.

Figure 8 : Site-to-Site IPsec VPN Topology



In this example, Site-SJ and Site-LA act as VPN gateways for two private enterprise networks that are connected via a public network. The IPsec tunnel is a logical entity that securely connects both sites of the enterprise. The tunnel is established by negotiating a set of security parameters between the two sites. These security parameters include:

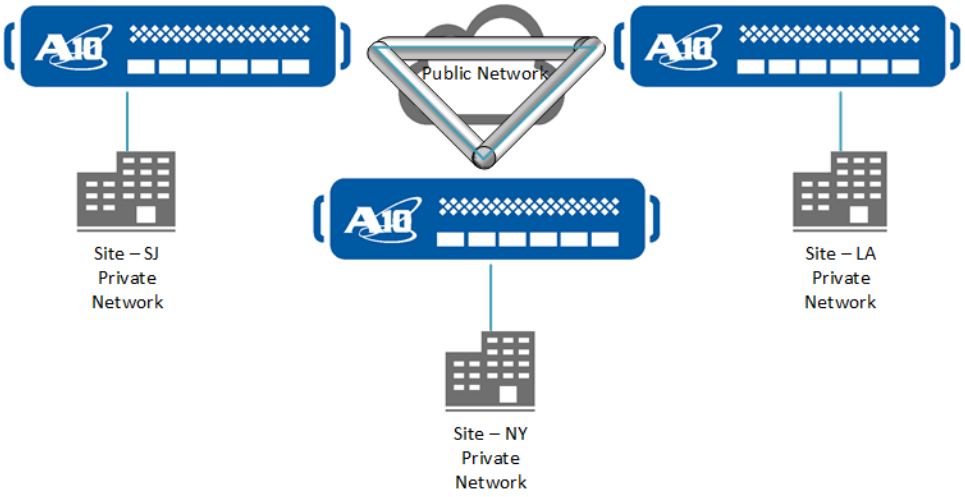
- Network access policies
- Cryptographic algorithms
- Modes of operation
- Security services such as confidentiality and authentication

When a client from Site-SJ sends traffic to a client at Site-LA, the ACOS device encapsulates and encrypts the packet. The encrypted packet is sent from Site-SJ to Site-LA, where it is decrypted and decapsulated.

Multi-Site

Multi-site IPsec VPN provides security for network traffic between multiple private networks.

Figure 9 : Multi-site IPsec VPN Topology



In this example, Site-SJ, Site-LA, and Site-NY act as VPN gateways for three private enterprise networks that are connected via the Internet.

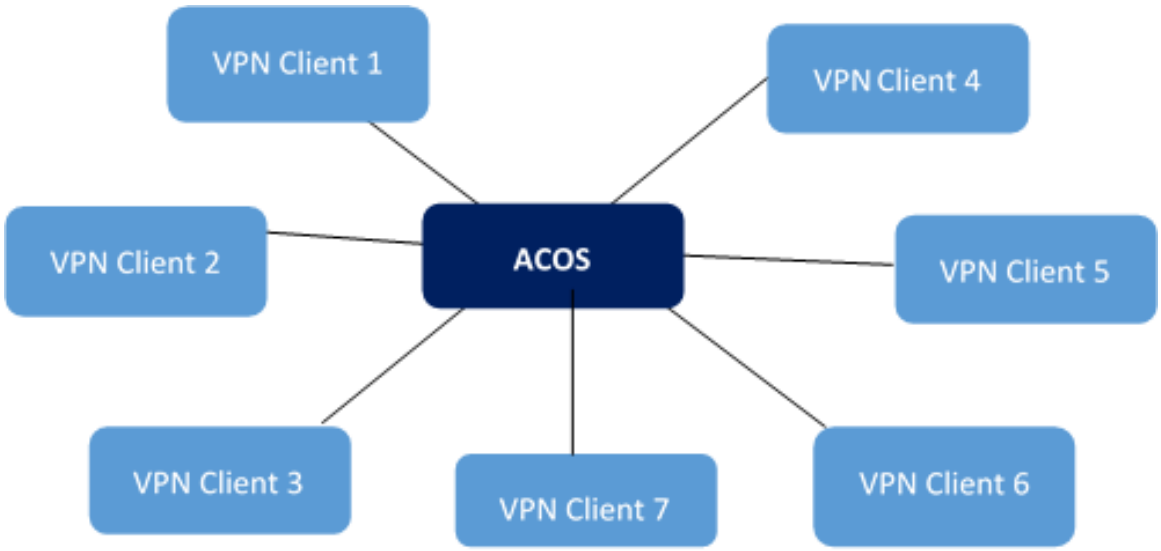
NOTE: Authentication Header (AH) security protocol is not supported.

Client-to-Site

Client-to-Site supports and allows the ACOS device to perform as a security gateway and acts as the IPsec remote access server (IRAS). The IPsec remote access client (IRAC) acts as the VPN device connecting to the ACOS device. Thus, the ACOS device acts as a VPN tunnel concentrator that allows multiple VPN clients to establish an IPsec tunnel.

The following figure describes the topology of the IPsec Client-to-Site.

Figure 10 : Example of a Potential IPsec Client-to-Site Topology



This feature describes the procedure for adding client-to-site VPN support. The client can be a host or a picocell, having a few hosts behind it. The ACOS works as a security gateway and it can accept multiple clients for the IPsec VPN connection.

The IKE CP, known as the **configuration payload** is an Internet Key Exchange (IKE) version 2 feature used to proliferate the facilitating data from an IKE responder to the IKE originator. It supports route-based VPNs only. The following is a list for salient features of IKE CP.

- It is a single IKE gateway that accepts the IKE connections from a group of peers.
- The IKE gateway uses CP to assign the IP address, netmask, gateway, and other parameters.

It supports the LTE mode of operation.

Components

The main components of IPsec VPN are as the following.

The following topics are covered:

Gateway Components

The VPN gateway component is used to configure basic gateway information for both the local and remote side of the VPN connection. The VPN gateway components include the following.

The following topics are covered:

Defined Mechanisms

IPsec defines mechanisms for:

- Security Association (SA) - SA is a representation of a one way connection that uses encryption and authentication methods to protect data communications.
- Internet Key Exchange (IKE) - IKE is a key management protocol that negotiates SAs for IPsec sessions. An IKE configuration defines the algorithms and keys used to establish a secure connection with the peer gateway. The keys are generated as part of the SAs and do not need to be configured.

IKE SAs authenticate the VPN gateways and IPsec SAs secure the VPN communication.

NOTE: Both IKEv1 and IKEv2 versions are supported. ACOS devices that are VPN peers are called IKE gateways.

Key Exchange

Diffie-Hellman (DH) key exchange is a public key exchange method that provides a way for two IPsec gateways to establish a shared secret key to set up IKE and IPsec SAs. The shared secret key cannot be reverse generated even if it is somehow intercepted.

The DH group controls the strength of the keying material exchanged. The devices at each end of the VPN tunnel use the keying materials to generate the shared secret key, and their own private keys.

NOTE: Higher-numbered DH groups provide stronger keying material than lower-numbered groups. Accordingly, higher-numbered DH groups also require more processing power than lower-numbered groups.

The following DH groups are supported:

- 1
- 2
- 5
- 14
- 15
- 16
- 18
- 19
- 20

NOTE: For IPsec SAs, a new DH key using a different DH group can be generated by enabling perfect forward secrecy (PFS).

IKE Gateway Components

The VPN IKE gateway component is used to establish an IKE SA.

The following topics are covered:

IKEv1

This section has the following sub-sections:

- Main Mode
- Aggressive Mode

Main and Aggressive are the two IKE encapsulation modes that are used to establish an IKE SA in IKE version 1. The IKE mode specifies the encapsulation used for IKEv1 during generation of the secret shared key and private keys.

NOTE: These modes apply only to IKE version 1.

Main Mode

Main mode performs 3 exchanges (6 packets total) to establish the IKE SA and to verify the identity of the peer gateway.

1. The first exchange negotiates the security policy.
2. The second exchange performs the DH exchange.
3. The third exchange authenticates the identity of the gateways.

The 3rd exchange is protected by the security policies of the first two exchanges. Thus, the main mode protects the identity of the gateways.

NOTE: Main mode is enabled by default.

Aggressive Mode

Aggressive mode establishes the IKE SA using fewer packets than main mode (3 packets total); however, it does not protect the identity of the gateways. Because of the lack of identity protection, aggressive mode is less secure than main mode.

IKEv2

IKEv2 is a more streamlined version of IKEv1 with NAT-T, DPD, and Anti-replay window built in by default. IKEv2 has less overhead and hence better performance than IKEv1. For IKEv1 and v2, the preshared key configured must be the same on both ends.

NOTE: For IKE version 2, only one exchange procedure is defined and it has no concept of main or aggressive mode.

Internet Key Exchange (IKE) Message Fragmentation

Internet Key Exchange (IKE) messages are exchanged using the User Datagram Protocol (UDP). The majority of IKE messages are small (below several hundred bytes). If IPsec uses certificates for authentication, IKE messages exceed the interface MTU size (1500 bytes) and fragment at the IP level. These fragments cannot pass through some intermediate network devices, preventing IKE communication and the establishment of an IPsec tunnel. Also, the jumbo frame MTU cannot be set if the IPsec peers are connected through public networks.

ACOS supports the IKE message fragmentation solution (RFC 7383) to split the large IKE messages into a series of smaller messages. This solution allows IKE messages to pass through environments that might block IP fragments. You can use the `fragment-size` command to configure the fragment size between 576 and 1280 bytes.

NOTE:

IKE message fragmentation is:

- specific to VPN in ACOS
- supports IKEv1 and IKEv2
- based on the IKE gateway

CLI Configuration

The following configuration example sets the fragment size to 578:

```
ACOS(config)# vpn ike-gateway name
ACOS(config-ike-gateway:name)# fragment-size 578
```

Show Command

To view the newly added counters, use the following show command:

```
show vpn ike-sa
```

For more information, see *Command Line Reference*.

Limitation

IKEv1 peers on Windows 7 and Windows 8 only fragment messages if they expect certificates.

Tunnel Components

The IPsec tunnel component is used to configure the tunnel between the two VPN gateways. The tunnel carries the IPsec traffic. The maximum number of IPsec tunnels that the ACOS platform supports depends on the total memory of the ACOS platform.

IPsec tunnels require dedicated gateway hardware or software equipment at both ends of the tunnel to encrypt and decrypt traffic flowing through the VPN.

The IPsec tunnel components include the following.

The following topics are covered:

Modes

IPsec mode specifies the encapsulation used for the IPsec traffic.

NOTE: Only the tunnel mode is supported. In tunnel mode, the client packet is encrypted and encapsulated in an IP packet. Transparent mode is not supported.

Re-Keying

IKE and IPsec each use SAs, and each type of SA ages out. When an IKE SA or IPsec SA ages out, a new IKE SA or IPsec SA is negotiated and the old IKE SA or IPsec SA is terminated. For IKE SAs, the age is measured in lifetime (seconds). For IPsec SAs, the age is measured in lifetime (seconds) or lifebytes (megabytes). By default, lifetime is used for IKE and IPsec SAs.

Lifebytes

Maximum number of bytes of data that are transferred using a given SA. You can

specify between 0-8,000,000 MB. The default is set to 0 MB (disabled).

Lifetime

Maximum number of seconds an SA is active. Both IKE and IPsec SAs have separate configurable SA lifetimes.

This section has the following sub-sections:

- IKE SA Lifetime
- IPsec SA Lifetime

IKE SA Lifetime

You can specify between 300-86400 seconds. The default is 84600 seconds (24 hours).

IPsec SA Lifetime

You can specify between 300-28800 seconds. The default is 28800 seconds (8 hours).

When an SA approaches a threshold near the age limit (measured in MB or seconds), ACOS re-keys to re-establish the SA. SA aging and re-key are used for security purposes to protect the traffic. The more traffic passed using the same key, the easier it is for the connection to be compromised.

Methods

The following topics are covered:

The ACOS implements the following IPsec VPN methods.

Authentication

Authentication methods are used to verify the peer's identity. During IKE phase 1, both peers authenticate each other using a negotiated authentication method.

One option is to manually configure the same pre-shared keys on both ends of the connection.

ACOS also supports IPsec and IKE authentication of the peer gateway using digital certificates for encryption and decryption called Rivest-Shamir-Adleman (RSA) signatures. With digital certificates, each peer is manually or dynamically enrolled with a Public Key Infrastructure (PKI) server. When a tunnel is established, the public keys are dynamically obtained through IKE and validated against the certificate from the PKI server. This method avoids manual configuration of keys on both devices.

Signature Authentication in IKEv2

Internet Key Exchange version 2 (IKEv2) supports Rivest-Shamir-Adleman (RSA) and Elliptic Curve Digital Signature Algorithm (ECDSA) authentication methods. The RSA authentication method always uses SHA1 for generating signature payloads, while the ECDSA authentication method supports three elliptic groups, with a fixed hash algorithm attached to each group.

ACOS supports signature authentication in IKEv2. It facilitates the use of stronger hash algorithms for generating authentication payloads during an IKEv2 negotiation. You can use different hash algorithms – SHA256, SHA384, and SHA512 for generating signature (authentication) payloads.

The `vpn signature-authentication` command must be configured to enable signature authentication. By default, it is disabled.

With the auth-method configuration, you can specify the hash algorithm used. If the signature authentication is enabled on both sides, then this algorithm is used for signing the authentication payload. If the signature authentication is enabled and if none of the algorithms is configured, the default algorithm is based on the key size.

The following table summarizes the algorithm selection for RSA and ECDSA certificates:

Table 1 : Hash Algorithms Based On Key Size

Key Type	Certificate Key Size in bits	Hash Algorithm Chosen
KEY_RSA	<= 3072	sha256
KEY_RSA	<= 7680	sha384
KEY_RSA	> 7680	sha512
KEY_ECDSA	<= 256	sha256
KEY_ECDSA	<= 384	sha384
KEY_ECDSA	> 384	sha512

If the same hash algorithm is configured on both sides, the configured hash algorithm is selected for signature authentication.

If only one side is configured with a hash algorithm, then the negotiation succeeds only if the other side is configured with the default algorithm corresponding to the key size of the certificate.

The `show vpn ike-sa <gateway-name>` command displays the algorithm selected for signature authentication.

The following table summarizes the selection algorithm for EC521 certificates:

Peer 1	Peer 2	Hash Selected	Negotiation Result
sha256	sha256	sha256	PASS
sha384	sha384	sha384	PASS
sha512	sha512	sha512	PASS
None	None	sha512	PASS
sha256	None	-	FAIL
sha512	None	sha512	PASS
sha384	None	-	FAIL

CLI Configuration

The following is the configuration:

```
!
vpn signature-authentication
```

```
!
vpn ike-gateway a1
  auth-method ecdsa-signature hash sha512
  key node1ec521key
  local-cert node1ec521crt
  local-id " C=US, ST=CA, L=SJ, O=A10, OU=Engineering, CN=node1_521"
  remote-id " C=US, ST=CA, L=SJ, O=A10, OU=Engineering, CN=node2_521"
  local-address ip 19.19.19.3
  remote-address ip 19.19.19.2
!
```

Show Commands

- To view the signature authentication field, use the following command:

  ```
  show vpn
  ```

  The `show vpn` command output includes the Signature Authentication field, which displays the configuration status of signature authentication. The signature authentication parameter is configured only in the shared partition. However, it is displayed in all the partitions.

- To view the hash algorithm used to sign authentication payloads, use the following command:

  ```
  show vpn ike-sa <gateway-name>
  ```

Limitations

The following are the limitations:

- Signature Authentication in IKEv2 works for site-to-site configuration only.
- The signature schemes corresponding to RSA (Section A.1 of RFC 7427) and ECDSA (Section A.3 of RFC 7427) are supported.
- When signature authentication is enabled, the IKE acceleration support on the QuickAssist Technology (QAT) devices has some limitations. The sign and verify functions for the ECDSA certificates are supported via the software paths.

Data Integrity

Data Integrity methods ensure that data is not modified in transit. Hashing algorithms are used during encryption to generate a hash value (also called digest) of the data. The hash value is used by the peer gateway that decrypts the data to verify that it has not been modified.

During SA negotiation, ACOS and the peer VPN gateway exchange a list of hashing algorithms that the devices can support.

NOTE: If the hash values at both ends do not match, the packet is dropped.

The following hashing algorithms are supported:

- Message Digest (MD) v5
- Secure Hashing Algorithm (SHA) v1
- SHA-256, SHA-384, SHA-512

Confidentiality

Encryption algorithms are used to maintain confidentiality of traffic. During SA negotiation, ACOS and the peer VPN gateway exchange a list of encryption algorithms the devices can support.

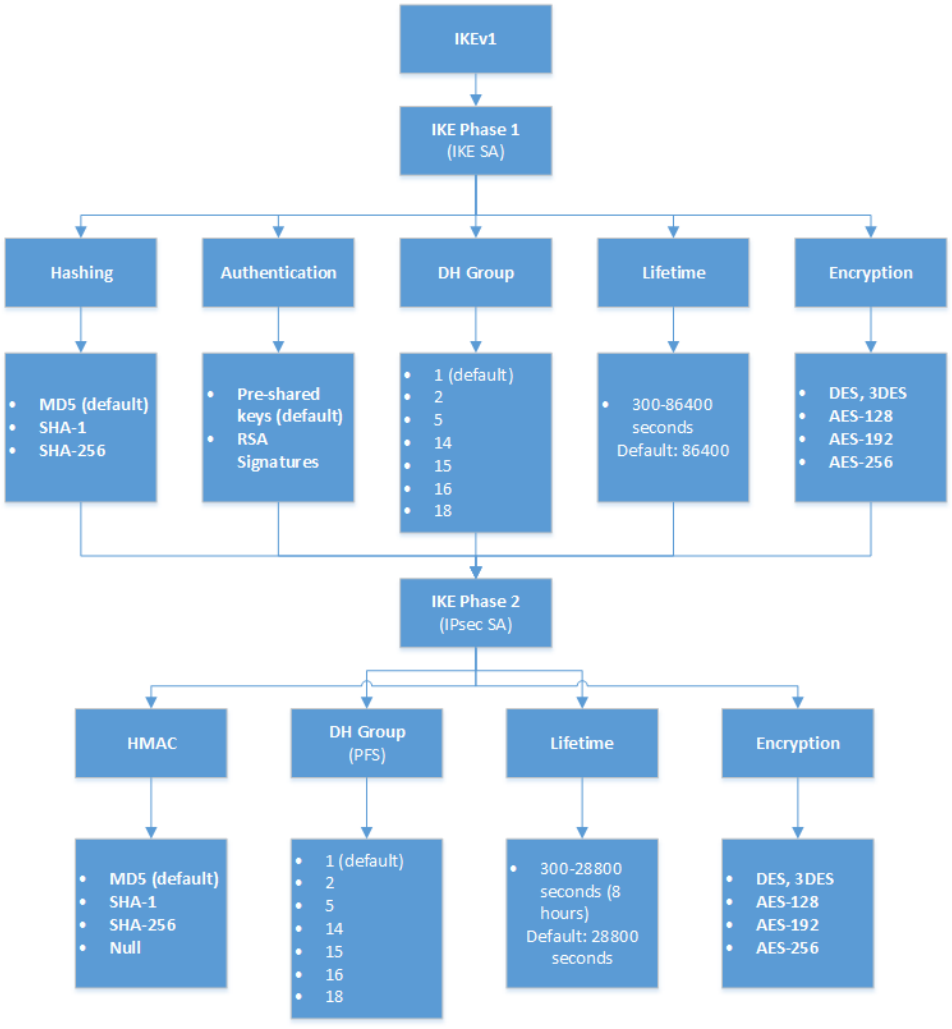
The following encryption algorithms are supported:

- Data Encryption Standard (DES)
- Triple DES (3-DES)
- Advanced Encryption Standard (AES) with 128-, 192-, or 256-bit keys, AES-GCM ciphers
- Null (no encryption)

NOTE: In cases where more than one set of encryption and hashing algorithms is supported by both VPN gateways, *priority value* is used as a tie-breaker. In this case, ACOS selects the group with the highest priority. The priority value can be 1-10. The highest priority value is 1. The lowest priority value is 10.

The following is a flowchart of the supported IPsec VPN standards and methods for IKEv1:

Figure 11 : IKEv1 Flowchart



Interface Management

ACOS supports management traffic between client and ACOS management interface over IPsec. The traffic will include Telnet, HTTP, DNS, SSH etc. whose destination is IP address on ACOS management port. And others send out through management port with source IP address on management port.

This can be configured on IKE gateway using both CLI and GUI. When IKE gateway is configured with management interface, the local IP of IKE gateway should be the IP address on management interface. The management traffic is sent to IKE gateway peer IP address over IPsec tunnel. The maximum IPsec gateway is 8, and maximum IPsec tunnel is 8.

Currently, interface management is supported only for shared partition. However, for management traffic that is supported by IPsec management plane, it includes traffic from both shared partition and L3V partition.

Virtual Router Redundancy (VRRP-A)

ACOS supports IPsec SA session synchronization for stateful failover. If a failover occurs, active IKE and IPsec SAs that have been synchronized continue uninterrupted. Only active-standby mode is supported.

NOTE: IPsec SA session synchronization requires the configurations on the ACOS devices in the VRRP-A pair to be the same.

L3V Partition Aware

IPsec is L3V partition aware. This means that you can specify the private partitions on which IPsec can be configured.

NOTE: Partitioning allows the ACOS device to be logically segmented to support separate configurations. Administrators assigned to a partition can manage only the resources inside that partition. Optionally, Layer 2/3 virtualization can be enabled on individual partitions, allowing partitions to own their own network resources. The Layer 2/3 virtualization functionality that you can enable on individual partitions is sometimes called “L3V".

L3V Partition Limitations

For L3V partitions, ACOS is unable to perform the DNS lookups. This implies that, no URLs for OCSP or CRL works for the L3V Partition.

Anti-Replay Window

Anti-replay window protects against replay attacks, in which a malicious party sends IPsec packets containing sequence numbers captured from the session’s legitimate traffic.

Anti-replay uses a moving window based on the sequence number. A sequence number received that is greater than the last previous sequence number is set as the new sequence number. The window is shifted based on the new sequence number. If a packet is received whose sequence number is not in the window, the packet is dropped because it is treated as an already seen packet.

If a packet is received whose sequence number is within the window, one of the following occurs:

- If the sequence number has already been seen, the packet is dropped.
- If the sequence number has not already been seen, the packet is passed to the rest of the system for processing.

The anti-replay window can be set to different sizes, which then determines how many sequence numbers can be marked in the window. By default, the anti-replay window is set to 0 (disabled). The anti-replay window can be set to a size of 0, 32, 64, 128, 256, 512, 1024, 2048, 3072, 4096, 8192.

Fragmentation and Jumbo Frames

The ACOS device supports jumbo frame fragmentation for IPsec VPN. The TCP Maximum Segment Size (MSS) adjustment feature enables you to specify the MSS.

NOTE: Jumbo frames have a size much larger than the typical 1500 byte Ethernet MTU size.

Management Traffic

The ACOS device supports management using the CLI over VPN tunnels on tunnel interfaces, loopback interfaces, and VE interfaces.

NOTE: IPsec VPN can be used to secure management traffic on data ports.

Management Parameters

The ACOS device supports management using the CLI over VPN tunnels on tunnel interfaces, loop-back interfaces, and VE interfaces.

NOTE: IPsec VPN can be used to secure management traffic on data ports.

The following topics are covered:

Gateway Parameters

The following is a list of the VPN Gateway Parameters you can configure:

Table 2 : VPN Gateway Parameters

Parameter	Description	CLI syntax	Supported values
Gateway name	VPN gateway name is the name of the VPN gateway configuration on the ACOS device.	`[no] vpn ike-gateway gateway-name`	Valid string Default: None

IKE version and mode (IKE v1 only)	IKE version is used to negotiate the SAs for IPsec sessions. Mode specifies the encapsulation used for IKE v1 during generation of the secret shared key and private keys. The version can be one of the following:
	• IKE v1
	◦ main - This option is more secure but uses more packets.
	◦ aggressive - This option uses fewer packets but is less secure.
	• IKE v2
		`ACOS(config-ike-gateway:jumbo)# ike-version ?`
		`v1 IKEv1 key exchange`
		`v2 IKEv2 key exchange`
		`ACOS(config-ike-gateway:jumbo)# ike-version v2 ?`
		`<cr>`
		`ACOS(config-ike-gateway:jumbo)# ike-version v1 mode ?`
		`main Negotiate Main mode (Default)`
		`aggressive Negotiate Aggressive mode`
			IKE v1 or v2
			Default: IKE v2
			Main or aggressive
			Default: Main
			This option is ignored if IKE v2 is used.

Interface Management	Interface Management enables support to handle traffic on management interface.	`[no] interface-management`	Enabled or disabled Default: Disabled

Table 2 : VPN Gateway Parameters

Parameter	Description	CLI syntax	Supported values
	The management traffic will use kernel to perform encrypt/decrypt. When the IPsec SA is generated, there will be a policy route added in kernel.		
Authentication	Authentication method is used for IKE phase 1. Pre-shared key, rsa-signature, and ecdsa-signature are supported. The same pre-shared key string must be configured on the peer VPN gateway.	`[no] auth-method preshare-key | rsa-signature | ecdsa-signature string`	Preshared key string: 1-127 characters
Fragmentation	Enables IKE message fragment and sets fragment size.	`[no] fragment-sizefragment-size`	Size: 576-1280
Diffie-Hellman (DH) group	The DH group controls the strength of the keying material exchanged during initiation of the IKE SA. During this phase, the devices at each end of the VPN tunnel use the keying materials to	`[no] dh-group group-num`	• 1 • 2 • 5 • 14 • 15 • 16 • 18 • 19 • 20

Table 2 : VPN Gateway Parameters

Parameter	Description	CLI syntax	Supported values
	generate the public (shared) key, and their own private keys. Higher-numbered DH groups provide stronger keying material than lower-numbered groups. Accordingly, higher-numbered DH groups also require more processing power than lower-numbered groups.		Default: 1
Algorithm Group (1-4)	Algorithm groups specify the encryption settings for IKE. A gateway can have up to 4 algorithm groups. During SA negotiation, one of the configured algorithm groups is selected. If one of algorithm groups are the same on both sides for the encryption configuration, the tunnel sets up. if all differ, tunnel sets down.	`[no] encryption encryption alg hash hash alg priority num`	Encryption algorithms: • DES, 3DES • AES-128 • AES-192 • AES-256 • AES-GCM-128 • AES-GCM-192 • AES-GCM-256 • Null Hashing algorithm: • MDv5 • SHA-1 • SHA-256,

Table 2 : VPN Gateway Parameters

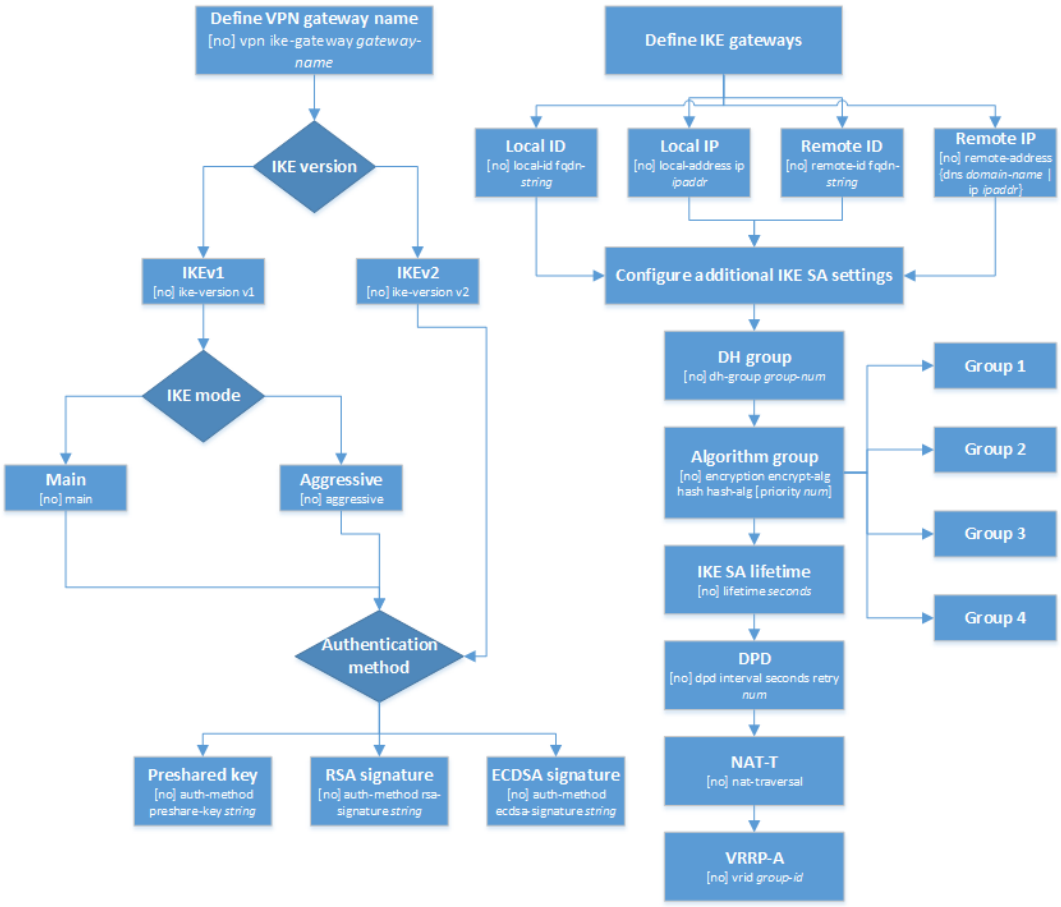
Parameter	Description	CLI syntax	Supported values
			SHA-384, SHA-512 Priority: 1-10 Default: DH group 1, aes128, sha1 and priority 5 Default priority for new groups is 5
Local ID	Local ID is the ID value of the gateway for IKE phase 1. It is used by the local peer to identify itself with the remote peer.	`[no] local-id`	Length: 1-256 Default: Not set
Remote ID	Remote ID is the ID value of the peer gateway for IKE phase 1. It is used by the remote peer to identify itself with the local peer.	`[no] remote-id`	Length: 1-256 Default: Not set
Local IP	Local IP is the IP address of the VPN gateway.	`[no] local-address {ip | ipv6 ipaddr}`	Valid IP address. Default: Not set
Remote IP	Remote IP is the DNS name or IP address of the peer VPN	`[no] remote-address {dns domain-name | ip |`	Valid DNS name or IP address

Table 2 : VPN Gateway Parameters

Parameter	Description	CLI syntax	Supported values
	gateway.	`ipv6 ipaddr}`	Default: Not set
Lifetime	Lifetime is the maximum number of seconds an IKE SA can remain in effect. After the SA ages out, a new IKE SA is negotiated. Then, the old SA for ESP based on the old IKE SA is deleted.	`[no] lifetime seconds`	300-86400 seconds Default: 86400
DPD	DPD tests idle VPN sessions to ensure the session peer is still responsive.	`[no] dpd interval seconds retry num`	Interval: 1-3600 seconds Retries: 1-10 Default: 0 (disabled)
NAT traversal	NAT traversal enables support for NAT. When this option is enabled, the ACOS device encapsulates ESP traffic inside UDP packets before sending them to the peer VPN gateway.	`[no] nat-traversal`	Enabled or disabled Default: Disabled

The following is a flowchart for the VPN gateway configuration:

Figure 12 : VPN Gateway Configuration



Tunnel Parameters

The following is the list of the VPN IPsec tunnel parameters you can configure:

Table 3 : VPN IPsec Tunnel Parameters

Parameter	Description	CLI syntax	Supported values
IPsec tunnel name	IPsec tunnel name is the name of the IPsec tunnel configuration on the ACOS device.	**`[no] vpn ipsec`** *`tunnel-name`*	Valid string Default: None
IKE	IKE gateway is the	**`[no] ike-gateway`**	Name of a VPN

Table 3 : VPN IPsec Tunnel Parameters

Parameter	Description	CLI syntax	Supported values
gateway	name of the VPN gateway that is used for tunnel connections.	*gateway-name*	gateway configuration Default: Not set
Tunnel interface and next hop	ACOS interface is used as the local end of the tunnel. The tunnel interface must be configured, before you can use it, in an IPsec tunnel configuration. The next-hop IP address typically is the remote tunnel IP address. This next hop IP address must match the static route entry's next hop. Connectivity can be verified using icmp ping success.	**[no] bind tunnel** *num next-hop ipaddr*	Tunnel ID and IP address Default: Not set
Mode	Mode specifies the encapsulation used for the IPsec traffic. Only the tunnel mode is supported. In tunnel mode, the client packet is encrypted and encapsulated in an IP packet.	**[no] mode tunnel**	Tunnel Default: Tunnel
Protocol	Protocol is used to encrypt traffic on the tunnel. The current release supports ESP.	**[no] proto esp**	ESP Default: ESP
Diffie-	DH group to use for	**[no] dh-group**	• 1

Table 3 : VPN IPsec Tunnel Parameters

Parameter	Description	CLI syntax	Supported values
Hellman Group	PFS.	*group-num*	• 2 • 5 • 14 • 15 • 16 • 18 • 19 • 20 Default: 0 (PFS disabled)
Algorithm Group (1-4)	Algorithm Group specifies the encryption settings for ESP/IPsec. The IPsec tunnel can have up to 4 algorithm groups. During SA negotiation, one of the configured algorithm groups is selected. If one of algorithm groups are the same on both sides for the encryption configuration, the tunnel sets up. if all differ, tunnel sets down.	**[no] encryption encrypt-alg hash hash-alg [priority** *num***]**	Encryption algorithms: • DES, 3DES • AES-128 • AES-192 • AES-256 • aes-gcm-128 • aes-gcm-192 • aes-gcm-256 • Null (no encryption) Hashing algorithm: • MDv5 • SHA-1 • SHA-256, SHA-384, SHA-512

Table 3 : VPN IPsec Tunnel Parameters

Parameter	Description	CLI syntax	Supported values
			Default: aes-128 sha1 Default priority for new groups is 5.
Lifetime	Lifetime is the maximum number of seconds an SA can remain in effect.	**[no] lifetime** *seconds*	300-28800 seconds (8 hours) Default: 28800 seconds
Lifebytes	Lifebyte is the maximum number of megabytes (MB) of data that can be transferred using a given SA.	**[no] lifebytes** *MB*	0-8000000 MB Setting this option to 0 disables aging based on bytes. Default: 0 (unlimited)
Anti-replay window	Anti-replay window specifies the number of IPsec packets for which the ACOS device remembers the IPsec packet sequence numbers. The anti-replay window protects against replay attacks, in which a malicious party sends IPsec packets containing sequence numbers captured from the session's legitimate traffic.	**[no] anti-replay-window win-size**	0, 32, 64, 128, 256, 512, 1024, 2048, 3072, 4096, 8192. Setting the window size to 0 disables anti-replay checking. Default: 0 (disabled)
Traffic	You can configure the	**[no] traffic-**	Auto, or valid IP

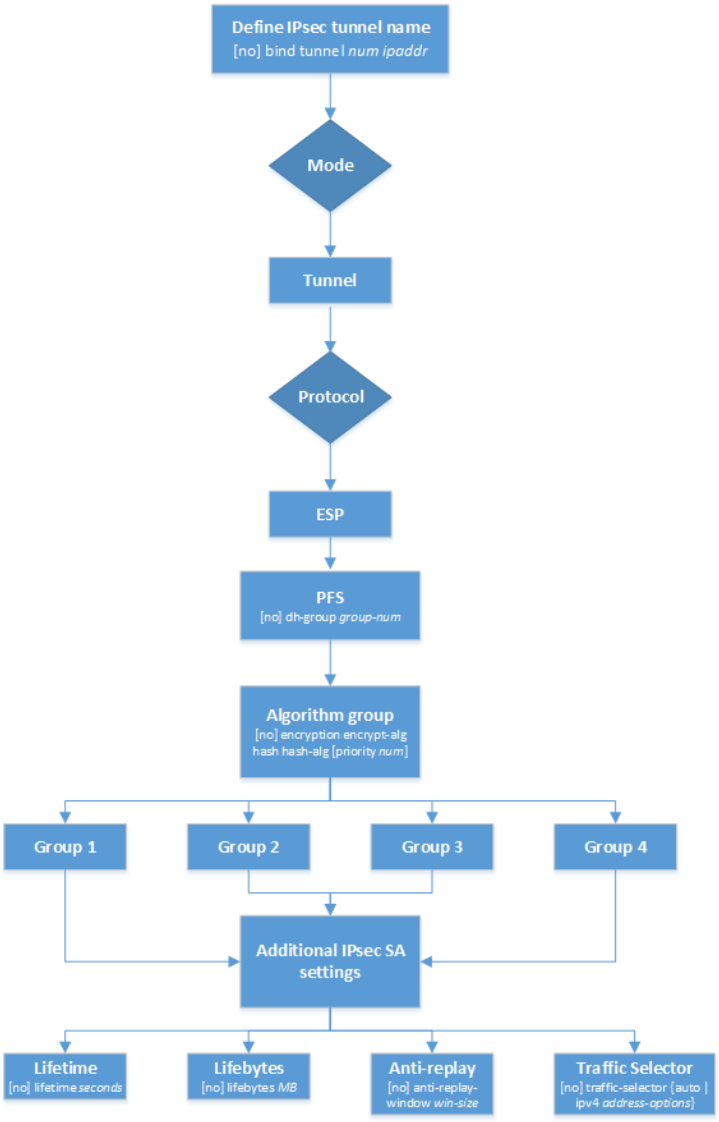
Table 3 : VPN IPsec Tunnel Parameters

Parameter	Description	CLI syntax	Supported values
selector	ACOS device to select the interface automatically, or you can explicitly specify the interface. If you plan to specify the interface, you must specify the following parameters: • Local IP address, protocol port, and IP protocol number • Remote (peer) IP address, Layer 4 transport protocol port, and IP protocol number	**selector** {auto | ipv4 | ipv6 *address-options* | ipv4 | ipv6 *address-options*}	addresses and port numbers. Default: Local network: 0.0.0.0/0 • Local protocol port: 0 (any port) • Local protocol: 0 (any protocol) • Remote network: 0.0.0.0/0 • Remote protocol port: 0 (any port) • Remote protocol: 0 (any protocol)

The default values for IKEv1 and IKEv2 settings are as the following:

- Phase 1 – AES-128, SHA1, and DH1
- Phase 2 – AES-128, SHA-1, and DH0

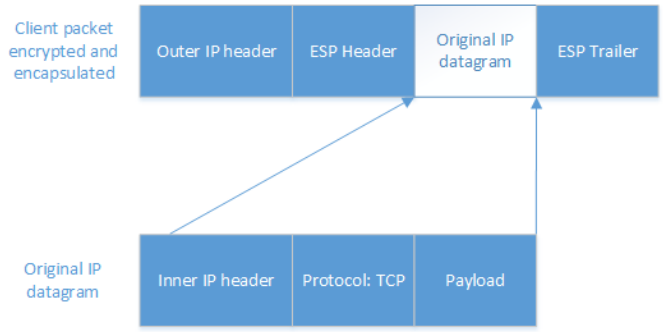
Figure 13 : IPsec Tunnel Configuration



Packet Format

ESP is the IPsec security protocol used to provide security in an IP datagram. ESP encapsulates the inner IP packet of the private networks over the public IP address space. The following figure shows an ESP-encapsulated packet:

Figure 14 : ESP-Encapsulated IP Packet in Tunnel Mode

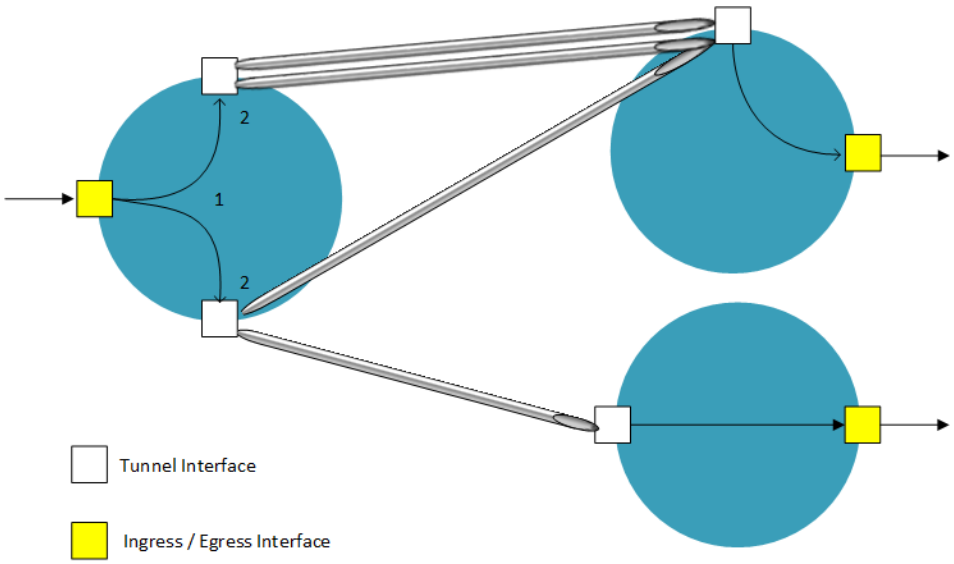


The inner IP header is encapsulated in another IP datagram and an ESP header is inserted between the outer and inner IP headers.

Packet Flow

The conceptual packet flow is shown as in the following Figure 15:

Figure 15 : IPsec Conceptual Packet Flow



The round circles reflect the routing domain of each ACOS device. The white boxes represent the tunnel interface. The yellow box represents the Ingress/egress interface, which could be a physical or virtual interface.

In Figure 15, the packet is coming in through an ingress interface. At point 1, a route lookup is done. Based on that route that is destination IP based, the ACOS device decides if the packet should go through a tunnel interface. The packet could be sent to another interface without a tunnel then the packet would not be tunneled, but if it does get routed to a tunnel interface as represented by the arrows, then the IPsec is applied on the packet. Multiple tunnels can be configured on the same tunnel interface. At point 2, two different IPsec tunnels are configured on the same tunnel interface. Point 2 can be considered as a demultiplex point. The next hop gateway IP address used in the configuration is used internally as a demultiplexer as it specifies the sub interface. The next hop IP address can be changed to specify which tunnel configured on that interface should be used. So there are two levels of demultiplexing, one at the route level and the other within the route at the next hop level. This is for the packets to get demultiplexed into the right tunnel.

It is recommended to use the following guidelines to keep the configuration simple, as the complexity increases when more and more tunnels are added:

- Use one tunnel interface per gateway pair.
- Use default traffic selectors unless it is absolutely needed by the peer device.

Policy-Based IPsec VPN

For supporting 5G network and making the ACOS more suitable for the multiple use cases, a policy-based IPsec VPN is required.

In some IPsec network environments, the peer IPsec device needs the destination IP of the inner packet to it, is the same as the destination IP of the IPsec outer packet to it. The current route-based IPsec VPN cannot work in this scenario and there is a need to add policy-based IPsec VPN support.

The following is a list of consideration points for opting policy-based IPsec VPN:

- Currently, the Firewall and the VPN can work together in IPsec VPN stateful mode.
- It is enhanced to the current Firewall function to implement the policy-based IPsec VPN, through the Firewall + VPN method.
- An additional new option IPsec is used to Firewall rule action, permitting to pass the traffic that matches this rule to the specified IPsec VPN tunnel.

The following topics are covered:

Feature Description

The following topics are covered:

Configuration

To utilize this feature and to make the policy-based IPsec VPN perform seamlessly, the user must configure the following components on the ACOS.

- Tunnel Interface
- VPN ike-gateway xxx and IPsec xxx
- Firewall rule that performs L2/L3/L4 match and the action “permit IPsec xxx”
- Firewall rule active status

Example

The user can refer to the following configuration example:

```
interface tunnel 3
  ip address 73.1.1.33 255.255.255.0
  ipv6 address 73::33/64
!
vpn stateful-mode
!
vpn ike-gateway to_34
  auth-method preshare-key 123456
  encryption aes-192 hash sha1
  dh-group 5
  local-address ip 6.1.1.33
  remote-address ip 6.1.1.34
!
vpn ike-gateway to_34_66
```

```
  auth-method preshare-key 123456
  encryption aes-192 hash sha1
  dh-group 5
  local-address ipv6 6006::1:33
  remote-address ipv6 6006::1:34
!
vpn ipsec to_34
  dh-group 5
  encryption aes-192 hash sha1
  bind tunnel 3 73.1.1.34
  ike-gateway to_34
!
vpn ipsec to_34_66
  dh-group 2
  encryption aes-192 hash sha1
  traffic-selector ipv6 localv6 ::/0 remotev6 ::/0
  bind tunnel 3 73::34
  ike-gateway to_34_66
!
rule-set rs1
  rule r1
    action permit ipsec to_34
    source ipv4-address 11.1.1.0/24
    source zone any
    dest ipv4-address any
    dest zone any
    service any
    application any
  rule 2
    action permit
    source ipv4-address 6.1.1.0/24
    source zone any
    dest ipv4-address any
    dest zone any
    service any
    application any
  rule 3
    action permit ipsec to_34_66
    ip-version v6
    source ipv6-address 6011::/64
```

```
    source zone any
    dest ipv6-address any
    dest zone any
    service any
    application any
  rule 4
    action permit
    ip-version v6
    source ipv6-address 6006::/64
    source zone any
    dest ipv6-address any
    dest zone any
    service any
    application any
!
fw active-rule-set rs1
!
```

Interfaces

The following is a list of the important interface points, which can be considered for this feature:

- The Policy-based IPsec VPN uses Firewall rule as the policy.
- The current ACOS Firewall rule match is used, when the packet comes in.
- If the packet can match a Firewall rule and the rule action is “permit IPsec”, then the user can pass the packet to the specified IPsec to do IPsec processing.

CLI Configuration

The following topics are covered:

Configuration Commands

The user can configure this feature by adding the new option IPsec to the Firewall rule action permit. The following is the configuration command set for this feature.

```
AX3030-105.33(config-rule set:rs1-rule:r1)#action permit ?
  listen-on-port        Listen on port
  log                   Enable logging
  cgnv6                 Apply CGNv6 policy
  forward               Forward packet
  ipsec                 Apply IPsec encapsulation
  <cr>
SoftAX105.42(config-rule set:rs1-rule:r1)#action permit ipsec ?
  NAME<length:1-31>  IPsec name
SoftAX105.42(config-rule set:rs1-rule:r1)#action permit ipsec ipsec_name?
  <cr>
```

Show Commands

The following is the show command set for this feature. The new action option IPsec can be shown out in the show rule-set.

```
AX3030-105.33#show rule-set rs1
Rule-Set-Name: rs1
Rule-Set-Status: active
Unmatched-Drops: 5    Action-Permit: 227    Action-Deny: 0    Action-
Reset: 0
Total-Rule-Count: 5
Rule-Name:                      r1
Hit-Count:                       2
Action:                             permit ipsec to_34
Status:                             enable
Permit-bytes:                   164
Deny-bytes:                      0
Reset-bytes:                     0
Total-bytes:                     164
Permit-packets:                  2
Deny-packets:                    0
Reset-packets:                   0
```

```
Total-packets:                  2
TCP-active-session:             0
UDP-active-session:             0
ICMP-active-session:            0
SCTP-active-session:            0
OTHER-protocol-active-session:  0
Total-active-session:           0
TCP-session:                    0
UDP-session:                    0
ICMP-session:                   0
SCTP-session:                   2
OTHER-protocol-session:         0
Total-session:                  2
```

Licensing and Platforms

The following topics are covered:

Upgrading or Downgrading Results

The following are the important points, referring to either upgrading or downgrading the system, as per the impact of this feature:

- The command is expected to be ported to all future releases.
- Upgrading is an expected and applicable mode, which is not an issue.
- Downgrading will wipe out the command and malfunctioning.

Dependencies

The following is a list of inter-related dependencies for this feature:

- The Firewall rule match is used to decide, whether the existing traffic needs IPsec VPN processing.

- The current “Enforce Traffic Selector” feature can be used to control the traffic.

Risks or Assumptions

The following is a list of risks or assumptions for this feature:

- The policy-based VPN has a high priority than the current route-based VPN.
- If the traffic is suited for the policy-based VPN and the route-based VPN at the same time, then the policy-based VPN gets priority and it overrules the other option.
- The need is to only perform the Firewall rule match for the first packet in the policy-based IPsec implementation.
- For the following FWD and REV packets, the performance of the IPsec processing and forward is based on the connection session.

Limitations

The following is a list of limitations for this feature:

- The policy-based VPN only works in the stateful mode and the VPN stateless mode does not support the policy-based VPN.
- The policy-based VPN implementation is only for the site-to-site VPN.
- The Firewall rule for policy-based VPN is L2/L3/L4 match rule, which is not supported using the Firewall application match rule.
- The Firewall rule for policy-based VPN is not supporting the logging option.
- In the Firewall + SLB + IPsec environment, the IPsec tunnel selected through the SLB server selection has a high priority than the IPsec tunnel selected by the Firewall rule, which is specific to the L4 SLB virtual port.
- For the local send out packet, the Firewall rule match is not done as of now and the policy-based VPN does not support the local traffic.

IPsec Active - Backup Tunnel Support

ACOS allows configuring the IPsec group to identify and group a set of tunnels to provide redundancy at the granularity of IPsec tunnels. This functionality is supported only for policy-based IPsec tunnels.

For policy-based IPsec, the tunnels are selected by the Firewall rule. The tunnels are configurable using the `action permit` or `action-group` options of the Firewall rule. The ipsec-group option is associated with the firewall rule to enable the ipsec-group functionality.

The `ipsec-group` command is used to configure the ipsec-group. An ipsec-group configuration consists of a group of tunnels with a priority number for each tunnel. The tunnel with the highest priority value is the active tunnel, and the other tunnel (s) serve as backup tunnels for that group.

During the tunnel establishment of a group, all the tunnels are brought up at the same time. After the tunnel establishment, traffic starts to flow through the active tunnel.

If the active tunnel is down, the traffic switches over to the backup tunnel with the next highest priority. The switchover between an active tunnel and a backup tunnel is done in the following ways:

- Automatic switchover through Dead Peer Detection (DPD) configuration.
- Manual switchover through priority configuration in the ipsec-group.
- When a tunnel interface is disabled or re-enabled.

For more information about the DPD configuration, see the Dead Peer Detection section.

CLI Configuration

The following is the IPsec configuration:

```
vpn ipsec a1
  lifetime 300
  bind tunnel 1 21.21.21.2
  ike-gateway a1
```

```
!
vpn ipsec a2
  lifetime 300
  bind tunnel 2 22.22.22.2
  ike-gateway a2
!
vpn ipsec-group group1
  ipsec a1 priority 7
  ipsec a2 priority 8
!
!
```

The following is the firewall configuration:

```
rule-set rs1
  rule 1
    source ipv4-address any
    source zone any
    dest ipv4-address any
    dest zone any
    service any
    application any
    action-group
      permit ipsec  a1
rule 2
  source ipv4-address any
  source zone any
  dest ipv4-address any
  dest zone any
  service any
  application any
  action permit ipsec-group abc
rule  3
  source ipv4-address any
  source zone any
  dest ipv4-address any
  dest zone any
  service any
  application any
  action-group
    permit ipsec-group abc
```

```
!
fw active-rule-set rs1
!
```

The following example creates an IPsec tunnel named a1, adds to the group named group1, and sets the priority to 9.

```
ACOS(config)#vpn ipsec a1
ACOS(config-ipsec:a1)#lifetime 300
ACOS(config-ipsec:a1)#bind tunnel 1 12.12.12.1
ACOS(config-ipsec:a1)#ike-gateway a1
ACOS(config-ipsec:a1)#exit
ACOS(config)#vpn ipsec-group group1
ACOS(config-ipsec-group:group1)# ipsec a1 priority 9
```

Show Commands

- The following is the show command to view the group and group member details:
  `show vpn ipsec-group <group-name>`
- The following is the show command to view all the groups in that partition:
  `show vpn ipsec-group`

Clear Commands

- The following is the clear command to clear all the SAs associated with the group:
  `clear vpn ipsec-group <group-name>`
- The following is the clear command to clear all the SAs of all the groups in all the partitions:
  `clear vpn ipsec-group all-partitions`

Limitations

- IPsec Active or Backup Tunnel supports only the site-to-site configuration.
- An IPsec tunnel can be associated with some data sessions per the configuration. With groups, a group member with the highest priority will be designated to pass the traffic. If the group member is disassociated from the group (using the no

command), while the traffic is being passed, the traffic will continue to pass through the existing member on these data sessions. However, when new ones are created, they will select a new group member to pass the traffic.

Digital Certificates Management

ACOS supports IKE authentication of the peer gateway using digital certificates in addition to preshared keys.

The digital certificates are centrally controlled on a Public Key Infrastructure (PKI) server. PKI uses asymmetric cryptography that includes a public and private key pair. The private key pair is known only to you or your local gateway and the public key pair is embedded in a certificate that is signed by a trusted certificate authority (CA). The CA establishes a trust point that is used to validate the certificates. CA certificates must be imported onto the ACOS device. The ACOS is not configured at the factory to contain a certificate store.

The certificate revocation list (CRL) is a list of client certificates that have been revoked by the CAs that signed them. The CRL has to be signed by the same issuer as the CA certificate. Otherwise, the IPsec endpoints are unable to establish a connection.

NOTE: Authentication using digital certificates is more secure than using preshared keys. Preshared keys also do not scale well because you have to manually enter the preshared key on each client.

PKI constitutes the following:

- Certificates
- Keys
- CA
- Validation

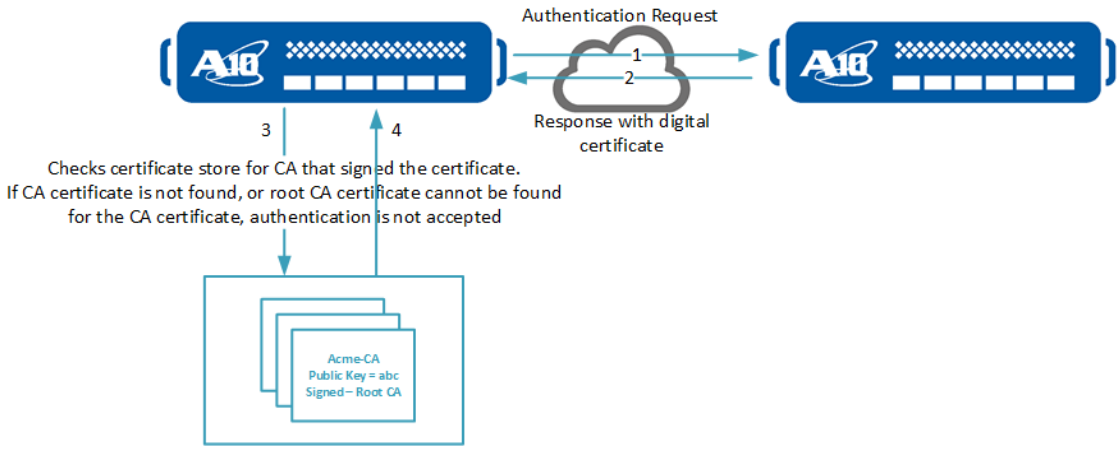
ACOS supports Privacy Enhanced Mail (PEM) format for certificate files and CRLs. This is the same as in SSL.

The following topics are covered:

Digital Certificates for IKE Authentication

Figure 16 shows a simplified example of an IKE digital certificate authentication handshake. The ACOS device sends an authentication request with all of its digital certificates to the peer ACOS device. The peer device checks its certificate store (also called certificate list) for a copy of the peer certificate. If it does not have a copy of the peer certificate, it then checks for a certificate from the CA that signed the peer certificate.

Figure 16 : Digital Certificates Authentication Topology



The following topics are covered:

Certificate Chain

Ultimately, a certificate must be validated by a root CA. Certificates from root CAs are the most trusted. They do not need to be signed by a higher (more trusted) CA.

If the CA that signed the certificate is a root CA, the peer device needs a copy of the root CA's certificate. If the CA that signed the peer certificate is not a root CA, the

peer device should have another certificate or a certificate chain that includes the CA that signed the CA’s certificate.

A certificate chain contains the “chain” of signed certificates that leads from the CA to the signature authority that signed the certificate for the peer. Typically, the CA that signs the peer certificate also provides the certificate chain. An example of a certificate chain containing three certificates is shown below:

```
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPIxrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpxDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPIxrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpxDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
ZS9naWYwITAfMAcGBSsOAwIaBBRLa7kolgYMu9BSOJsprEsHiyEFGDAmFiRodHRw
Oi8vbG9nby52ZXJpc2lnbi5jb20vdnNsb2dvMS5naWYwDQYJKoZIhvcNAQEFBQAD
gYEAheIVEe8vArUOZxKkUIGjaYymzJAh8Ty0uUPrikLpQ0IGezByVdbDUJ+HQLGp
2eruTPZpBNADaEfymstIPIxrsuCRhyr3Ymsa2rgzwy9kSXeG83H7E7HxRnpxDNZ8
l+uzpU/rk4j3bO/JVxPZMnwzMWriPSYgL1EKYcOSKyReACOSQ=
-----END CERTIFICATE-----
```

NOTE: The certificate chain file and the peer certificate files are text files. Each certificate must begin with the “-----BEGIN CERTIFICATE-----” line and end with the “-----END CERTIFICATE-----” line.

The certificate at the top of the certificate chain file is the root CA’s certificate. The next certificate is an intermediary certificate signed by the root CA. The next certificate is signed by the intermediate signature authority that was signed the root CA.

NOTE: If the CA that signs the peer certificate does not provide the certificate chain in a single file, you can use a text editor to chain the certificates together in a single file as shown above.

Certificates

The ACOS device has a certificate store that includes certificates signed by the various root CAs. The certificate store may also have some non-CA certificates that can be validated by a root CA certificate, either directly or through a chain of certificates that end with a root certificate.

The following topics are covered:

Each certificate is digitally "signed" to validate its authenticity. Certificates can be CA-signed or self-signed:

CA-signed

A CA-signed certificate is a certificate that is created and signed by a recognized CA. To obtain a CA-signed certificate, you must create a key and a Certificate Signing Request (CSR), and send the CSR to the CA.The CSR includes the key. The CA then creates and signs a certificate. Then, install the certificate on the ACOS device. When a peer device sends an authentication request, the ACOS device sends a copy of the certificate to the peer device, to verify its identity. To ensure that the peer device receives the required chain of certificates, you also can send peer devices a certificate chain in addition to the digital certificate. The example in Digital Certificates Authentication Topology uses a CA-signed certificate.

Self-signed

A self-signed certificate is a certificate that is created and signed by the ACOS device. A CA is not used to create or sign the certificate. CA-signed certificates are considered to be more secure than self-signed certificates. Likewise, peer devices are more likely to be able to validate a CA-signed certificate than a self-signed certificate.

Configuring Digital Certificates for IKE Authentication

Each IKE gateway needs a certificate and key pair. Local and remote IDs need to be configured on each gateway. These IDs are the subject distinguished name (DN) of your certificate. The remote ID of your peer needs to be configured with the subject DN of the remote gateway certificate. These IDs are used to determine which gateway configuration to select, as you can have multiple gateway configurations.

The following topics are covered:

Generating Certificate and Key

To configure digital certificates for IKE gateways, the first step is to get a certificate and key pair for the gateway. There are three ways to install a certificate and key pair on ACOS.

The following topics are covered:

NOTE: For EC private keys, ACOS supports prime256v1, secp384r1, and secp521r1.

Manual Method

To request and install a CA-signed certificate, use the following process.

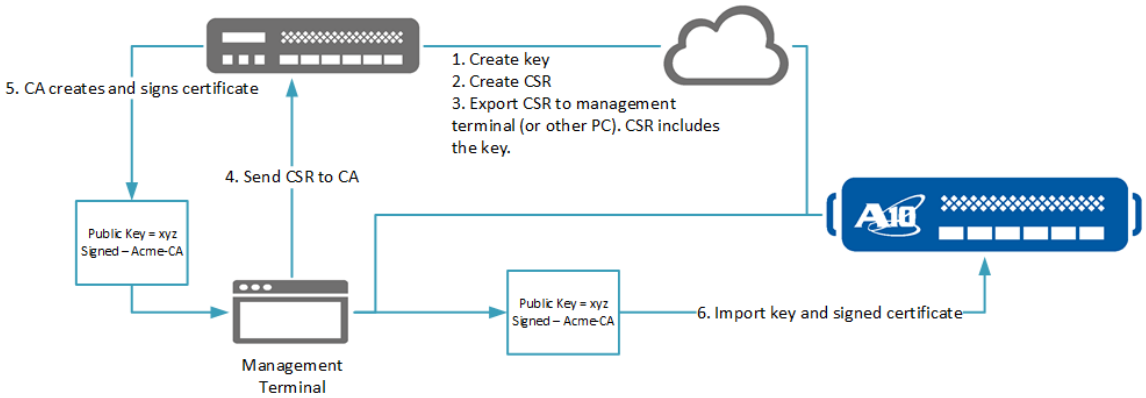
1. Create an encryption key.
2. Create a Certificate Signing Request (CSR). The CSR includes the public portion of the key, as well as information that you enter when you create the CSR.
3. Submit the CSR to the CA. If the CSR was created on the ACOS device, do one of the following:

- Copy and paste the CSR from the ACOS CLI onto the CSR submission page of the CA server.
- Export the CSR to another device, such as the PC from which you access the ACOS CLI. Email the CSR to the CA, or copy-and-paste it onto the CSR submission page of the CA server.

If the CSR was created on another device, email the CSR to the CA, or copy-and-paste it onto the CSR submission page of the CA server.

4. After receiving the signed certificate and the CA's public key from the CA, import them onto the ACOS device.
   - If the key and certificate are provided by the CA in separate files (PKCS #7 format), import the certificate. You do not need to import the key if the CSR was created on the ACOS device. In this case, the key is already on the ACOS device. If the certificate is not in PEM format, specify the certificate format (type) when you import it.
   - If the CSR was not created on the ACOS device, you do need to import the key also.
   - If the key and certificate are provided by the CA in a single file (PKCS #12 format), specify the certificate format (type) when you import it. If the CSR was not created on the ACOS device, you need to import the key also.
5. If applicable, import the certificate chain onto the ACOS device. The certificate chain must be a single text file, beginning with a root CA's certificate at the top, followed in order by each intermediate signing authority's certificate.

Figure 17 : Obtaining and Installing Signed Certificate from CA



To generate a key and a CSR, use the following command at the global configuration level of the CLI:

```
pki create csr csr-name  url
```

The *csr-name* can be 1-31 characters. The *url* specifies the file transfer protocol, username (if required), directory path, and filename. You can enter the entire URL on the command line or press Enter to display a prompt for each part of the URL. If you enter the entire URL and a password is required, you are still be prompted for the password.

- `tftp://host/file`
- `ftp://[user@]host[:port]/file`
- `scp://[user@]host/file`
- `rcp://[user@]host/file`
- `http://[user@]host/file`
- `https://[user@]host/file`
- `sftp://[user@]host/file`

This command displays a series of prompts, for the following information:

- IP address of the peer device to which to export the CSR
- Username for write access to the peer device
- Password for write access to the peer device
- Path and filename
- Key length, which can be 1024, 2048 or (on some 64-bit ACOS models) 4096 bits
- Common name, 1-64 characters
- Division, 0-31 characters
- Organization, 0-63 characters
- Locality, 0-31 characters
- State or Province, 0-31 characters
- Country, 2 characters

- Email address, 0-64 characters
- Passphrase to use for the key, 0-31 characters

NOTE: If you need to create a request for a wildcard certificate, use an asterisk as the first part of the common name. For example, to request a wildcard certificate for domain example.com and its sub-domains,enter the following common name: *.example.com.

After the CSR is generated, send the CSR to the CA. After you receive the signed certificate from the CA, use the *import* command to import the CA onto the ACOS device. The key does not need to be imported. The key is generated along with the CSR.

The following command generates and exports a CSR, then import the signed certificate as the following:

```
certificate:ACOS(config)#pki create csr training3 scp://user@172.16.101.228/home/user
Password[]?
input key bits(1024, 2048, 4096) default 1024:2048
input Common Name, 1~64:training3
input Division, 0~31:
input Organization, 0~63:A10
input Locality, 0~31:
input State or Province, 0~31:
input Country, 2 characters:US
input email address, 0~64: training3@a10networks.com
```

Importing a Certificate and Key

To import certificate and key files, place them on the PC that is running the CLI session, or onto a PC or file server that can be locally reached over the network.

NOTE: If you are importing a CA-signed certificate for which you used the ACOS device to generate the CSR, you do not need to import the key. The key is automatically generated on the ACOS device when you generate the CSR.

To import a certificate and its key, or a certificate chain, use the following command at the global configuration level of the CLI:

```
ACOS(config)#import cert training3
scp://user@172.16.101.228/home/user/training3.cer
```

You can enter the entire URL on the command line or press Enter to display a prompt for each part of the URL. If you enter the entire URL and a password is required, you are still be prompted, for the password.

To see the certificate, use the following commands:

```
ACOS(config)#show pki certificate
Name: ocsp-responder Type: certificate Expiration: Jul 15 23:47:42 2015
GMT [Unexpired, Unbound]
ACOS(config)#show pki certificate training3
Certificate:
        Date:
                Version: 3 (0x2)
                Serial Number:
                        34:e9:e8:61:00:00:00:00:fc:26
        Signature Algorithm: sha1WithRSAEncryption
                Issuer: DC=com, DC=a10lab, CN=AD03-CA
                Validity
                        Not Before: Jul 10 03:38:58 2015 GMT
                        Not After: Jul 10 07:38:58 2015 GMT
                Subject: DC=a10networks, DC=com, CN=training3
```

The “Issuer” filed in the above example is the CA. The subject represents the gateway, which in this example is called *training3*. The subject in the DN format is used as the local ID.

The process for exporting certificates, keys, and CRLs is the same as for SSL.

NOTE:

- In the manual method, you would have to manually upload and replace certificates for renewal and deletion respectively.
- For more information, see *SSL Certificate Management* chapter in the *SSL Configuration Guide*.

Configuring Simple Certificate Enrollment Protocol (SCEP) Certificates

Automatic Method

ACOS supports Simple Client Enrollment Protocol (SCEP) protocol for automatic certificate enrollment. SCEP is a part of the PKI infrastructure, which provides a simplified means of handling certificates for devices. ACOS SCEP automates the following processes:

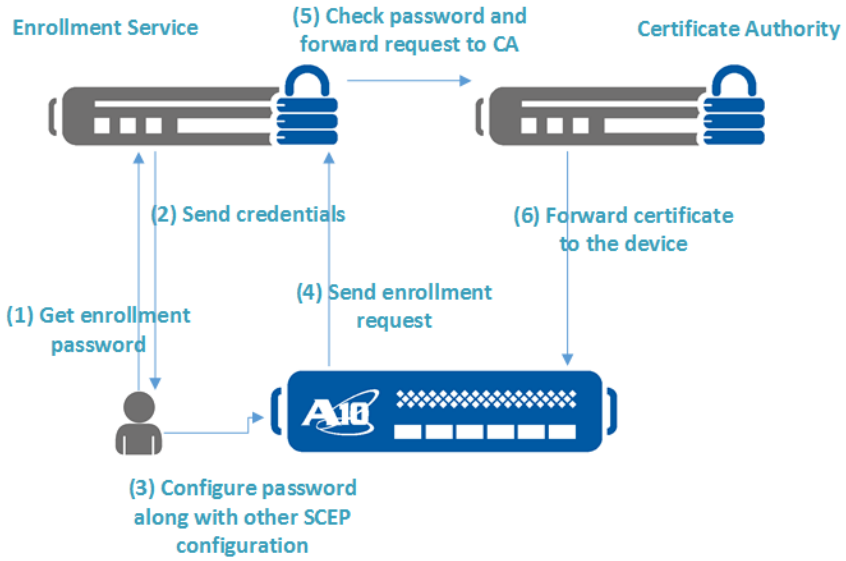
- Installing the certificates on each device
- Configuring each certificate
- Keeping a note of the certificate's expiry
- Getting and reinstalling renewed certificates

You only need to configure the CA credentials to enroll for the certificates the first time and after that the certificates are self-maintained by ACOS using the SCEP protocol.

Key Considerations:

- SCEP supports shared and L3V partitions. SCEP traffic follows the routing table rule and DNS setting on the shared partition. Hence, the network interfaces, routing rules, and DNS must be configured correctly in the shared partition, as the SCEP certificate is created in the L3V partition.

Figure 18 : SCEP Sample Topology



Configuring SCEP for IKE

The SCEP protocol client inside the ACOS queries an Enrollment Service (ES) for a certificate by passing appropriate credentials. The ES in turn passes the certificate request to the CA. The CA validates the request and passes the certificate to the ACOS.

ACOS also automatically handles renewal of the certificate when it is about to expire. This certificate can later be used for IPsec.

NOTE: ACOS supports SCEP for IKE using TCP, SSL, and HTTP. The IPsec data packets do not need this certificate.

1. Configure all required SCEP certificates.
2. Check configuration with the `show running configuration` command.
   a. show pki scep-cert status – Displays the status of the enrolled certificates.
   b. show pki scep-cert log <cert_name> from start – Displays the details of the certificate from the initial phases.
   c. show slb ssl cert – Displays the certificate notified as “SCEP Enrolled.”
3. Enroll to apply certificate.

The SCEP configuration is as follows:

```
pki scep-cert training
        url http://192.168.230.101/certsrv/mscep/mscep.dll
        dn cn=training, dc=a10networks, dc=com
        subject-alternate-name email training@a10networks.com
        password 5CF70971F182EB5945661123B856F424
        renew-before day 2
```

In the above configuration, an SCEP object called “training” is defined for PKI. The first line is the URL for which the device communicates to the peer device. The DN field describes the DN of the certificate you want to use for your IKE gateway, and a certificate is requested for this DN. The “subject-alternate-name” is the same as what is seen in the certificate request. The “password” field is required and can be requested via another URL from the SCEP server. This password needs to be used to communicate with the peer. Using the `renew-before` field you can actively specify when you want to renew the
certificate. You can renew before expiration by days, hours, or minutes.

```
#enroll !! Use to initiate communication and roll with the SCEP server
```

The `enroll` command is an operational command that forces the first communication of enrollment with the SCEP server. Without this command, the SCEP configuration would not work.

```
#show pki scep status
Certificate name: training status: SUCCESS
        Renew every 2 hours
        rotated files:1
```

The `show pki scep status` command shows you the status of the SCEP enrollment. If the server is down, the status indicates it as a FAILURE. In the event of a problem with the current certificate, you can revert to an older certificate as long as the certificate is valid. A maximum of five SCEP files can be rotated.

```
#pki copy-cert training rotation 1 training1R1
```

This command allows you to return back to a previous certificate. You would have to rename the certificate to a new name.

Verifying SCEP Configuration

The following is an example of a successful SCEP configuration log:

```
ACOS#show pki scep log training
loaded plugins: curl random x509 pkcs1 pkcs7 pkcs8 pem openssl gmp
transaction ID: D1C3FD30B7654E180C50D0178E7811A8
fingerprint: 9710fd08afa97616e24b79da2a75d1d6
no issuer certificate found for "C=US, ST=California, L=San Jose, O=A10
NETWORKS, OU=Software Engineering, CN=AD03-MSCEP-RA"
using trusted certificate "C=US, ST=California, L=San Jose, O=A10
NETWORKS, OU=Software Engineering, CN=AD03-MSCEP-RA"
written requested cert file '/a10data/cert/training' (1398 bytes)
return value 0
Client enrollment completed successfully
```

NOTE:

- SCEP is supported on L3V partitions. You can configure the same steps on L3V partition. Make sure the data interface is reachable to your certification server.
- Certificate renewal can be handled only if the administrator password is constant. In the event of changes to the administrator password, the configuration for that certificate must be changed accordingly.

Sample IKE Gateway Configuration

The following is a sample IKE gateway configuration:

```
vpn ike-gateway 38
        ike-version v2
auth-method rsa-signature
       key training
       local-cert training
       local-id “DC=a10networks, DC=com, CN=training”
       remote-ca-cert AD03-CA
       remote-id “DC=a10networks, DC=com, CN=training2”
        encryption aes-256 hash sha256
        local-address ip 192.168.38.1
        remote-address ip 192.168.382
```

NOTE: In the sample IKE gateway configuration, the user can only configure either the command '`auth-method rsa-signature`' or the command '`auth-method ecdsa-signature`', but not both of these commands.

By convention, the key name and cert name when using SCEP is the same. The subject in the DN format is used as the local ID. Use the `show pki cert` command for the local ID as shown below:

NOTE: You may have to ask the operator of the peer device for the remote ID.

```
#show pki certificate training
Certificate
        Data:
                Version: 3 (0x2)
```

```
            Serial Number:
                    34:e9:e8:61:00:00:00:00:fc:26
    Signature Algorithm: sha1WithRSAEncryption
            Issuer: DC=com, DC=a10lab, CN=AD03-CA
            Validity
                    Not Before: jul 10 03:38:58 2015 GMT
                    Not After: Jul 10 07:38:58 2015 GMT
            Subject: DC=a10networks, DC=com, CN=training
```

Certificate Management Protocol (CMPv2) Support for Security Gateway (SeG)

The Certificate Management Protocol (CMPv2) Support for Security Gateway (SeG) is required for implementing and supporting the CMPv2 protocol.

The following are the important points of this feature:

- It specifies Internet standards track protocol for the Internet community, relating to the Internet X.509 Public Key Infrastructure (PKI) Certificate Management Protocol (CMP).
- It is used for the purpose of utilizing the CFW as a Security Gateway as defined in the 3GPP TS 33.310 standards.
- With this protocol implementation and support, thunder CFW IPsec is complying with most of the SeG requirement opportunity, as SeG users are already utilizing the CMPv2 to distribute and manage certs for IPsec tunnels between E or G node B toward the IP Core.
- CMP supports shared and L3V partitions. CMP traffic follows the routing table rule and DNS setting on the shared partition. Hence, the network interfaces, routing rules, and DNS must be configured correctly in the shared partition, as the CMP certificate is created in the L3V partition.

Currently, ACOS supports SCEP protocol to perform cert enrollment and renewal for modules like SSL and IPsec. The CMPv2 is a protocol like SCEP, which is a part of the PKI infrastructure and provides a simplified means of handling certificates.

The following topics are included:

- Dependencies or Risks or Assumptions
- CLI Configuration
- GUI Configuration
- API Configuration
- Licensing and Platforms

Feature Description

The following topics are included:

- Functionalities
- Processes
- Log Files
- System Logs
- Status Command

Functionalities

The following is a list of important points, referring to the functionalities of this feature:

- **Enrolling the Cert Initial:** The ACOS generates the private key and sends an initialization request to CMP server(CA), by using a pre-shared secret key for mutual authentication.
- **Updating the Cert/Key:** The ACOS generates a new private key and sends a key update request to CMP server(CA), with an existing cert/key pair.
- **Renewing the Certificate:** It must be able to handle the renewal of certificates, when either the ACOS's certificate or the issuing CA's certificate is about to expire.
- Cert/key/log rotation after a successful update (in a10 directory).
- Allowing other a10lb modules to utilize CMP cert., such as VPN and SSL module. Notify a10lb for events like cert enrollment, delete, and so on.

- Supporting the following:
  - HTTP/HTTPs to connect with CMP server
  - EJBCA RA mode and client mode
  - IPv4 and IPv6

Processes

The ACOS goes through the following steps after a certificate is configured for enrollment:

- **Generating a private key:** In this step, RSA or EC key with specified key-length is generated for the certificate.
- **Getting the certificate:** The ACOS sends an initialization request (IR) to CMP server (CA), by using a pre-shared secret key for mutual authentication.
- **Storing the certificate:** After successful verification of the response from CA, the ACOS accepts the certificate and stores it in the appropriate location along with the private key is as well in its respective folder.
- **Notifying the other applications:** There are possibilities of applications such as VPN IPsec or client-SSL template waiting for the CMP enrolled the certificate. To address this, after successful enrollment, the CMP-client shell script notifies these applications, indicating that the certificate is ready for use.
- **Scheduling the renewal:** The AX handles the automatic renewal of the certificate when it is about to expire. The expiry date of both the enrolled certificate and the issuing CA's certificate is parsed to find the latest expiry date/time. Based on the configuration, it is either schedules a periodic renewal or a specific time renewal, before the certificate expires.
- **Rotating and storing the files:** After the renewal of the certificate, the old certificate and the key files are still stored for any future reference. The old files are rotated, and the new file replaces the existing file.

Log Files

The following is a list of important points for this topic:

- Each certificate is associated with a log file placed in a specific folder.
- The user can utilize the following: “show pki cmp-cert log <cert-name>”.

The following is an example:

```
AX1030(LOADING)#show pki cmp-cert log xx ?
  follow      Display current running cmp-cert log
  from-start  Display cmp-cert log from the beginning
  num-lines   Number of lines displayed of most recent entries in cmp-cert
log
  |           Output modifiers
```

System Logs

After each enrollment or renewal event, the system logs can be seen through the `show log` command.

Status Command

The current status of the certificate can be assessed through the “show pki cmp-cert status” command.

CLI Configuration

It covers the following topics:

- Configuration Commands
- aXAPI Commands
- Show Commands

Configuration Commands

The following is the CLI configuration command set for this feature.

```
CLI
AX1030(config)#pki cmp-cert ?
  NAME<length:1-63>  Specify Certificate name to be enrolled by using CMP
protocol
AX1030(config-cmp cert:test-https
AX1030(config-cmp cert:test-https)#?
```

```
  allow-unprotected-errors  Accept missing or invalid protection of
negative responses(CA likes EJCBA
                            tends to not protect negative responses)
  cert-type                 Specify the type of certificate
  clear                     Clear or Reset Functions
  cmp-ca                          The specific CA to trust while verifying the
signature of CMP response message
  cmp-cert                  The specific CMP server certificate to use and
directly trust when
                            verifying the signature of CMP response
message
  do                        To run exec commands in config mode
  end                       Exit from configure mode
  enroll                    Initiates enrollment of device with the CA
  exit                      Exit from configure mode or sub-mode
  log-level                 Level for logging output of CMP commands
(default 1 and detailed 2)
  max-poll time              Maximum time in seconds a(n) enrollment/key
update may take (default
120)
  no                        Negate a command or set its defaults
  recipient-dn              Distinguished Name of the CMP message
recipient, i.e., the CMP server
                            (usually a CA or RA entity))
  renew-before              Specify interval before certificate expiry to
renew the certificate
  renew-every               Specify the periodic interval in which to
renew the certificate
  secret                    Specify the pre-shared secret used to enroll
the device's certificate
  show                      Show Running System Information
  subject-alternate-name    Specify the Subject Alternate Name to use
while enrolling the
certificate
  subject-dn                Distinguished Name to use while enrolling the
certificate(For EJBCA
CA, this is the subject DN of an End Entity)
url                       CMP server's absolute URL(http(s)://host:
[port]/path), path is the
location to use for the CMP server(aka CMP alias)
```

```
  user-tag                    Customized tag
  write                       Write Configuration
AX1030(config-cmp cert:test-https)#cert-type ?
  ecdsa  ECDSA certificate
  rsa    RSA certificate (default)
AX1030(config-cmp cert:test-https)#cert-type ecdsa ?
  ec-key-length  Specify the size of the private key for the device
certificate in bits (default 384)
  <cr>
AX1030(config-cmp cert:test-https)#cert-type rsa ?
  rsa-key-length  Specify the size of the private key for the device
certificate in bits (default 2048)
  <cr>
AX1030(config-cmp cert:test-https)#log-level ?
  <1-2>  level for logging output of CMP commands(default 1 and detailed
2)
AX1030(config-cmp cert:test-https)#renew-before ?
  hour   Number of hours before cert expiry
  day    Number of days before cert expiry
  week   Number of weeks before cert expiry
  month  Number of months before cert expiry(1 month=30 days)
AX1030(config-cmp cert:test-https)#renew-every ?
  minute  Periodic interval in minutes
  hour    Periodic interval in hours
  day     Periodic interval in days
  week    Periodic interval in weeks
  month   Periodic interval in months(1 month=30 days)
AX1030(config-cmp cert:test-https)#subject-alternate-name ?
  email  Enter the e-mail address of the subject
  dns    Enter hostname of the subject
  ip     Enter the IP address of the subject
```

aXAPI Commands

The following is a list of aXAPI commands and examples for the CMP cert feature:

- CMP Cert Log
- CMP Cert Status

CMP Cert Log

```
Print the last 10 lines (default)
GET /axapi/v3/slb/ssl-cmp-cert-log/oper?name=<cmp-cert-name>
Print logs from start of the file
Get /axapi/v3/slb/ssl-cmp-cert-log/oper?name=<cmp-cert-name>&from-start=1
Print # of lines
Get /axapi/v3/slb/ssl-cmp-cert-log/oper?name=<cmp-cert-name>&num-lines=<#
of lines>
Follow the log file, print any new logs
Get /axapi/v3/slb/ssl-cmp-cert-log/oper?name=<cmp-cert-name>&follow=1
```

CMP Cert Status

```
Print all CMP cert enrollment/renew status
GET /axapi/v3/slb/ssl-cmp-cert-status/oper
Print the total # of records of cmp status
Get /axapi/v3/slb/ssl-cmp-cert-status/oper?total=true
```

Show Commands

The following is the show command set for this feature.

```
AX1030#show pki cmp-cert ?
log         Show cmp-cert enrollment log and debug information
  status     Show CMP enrollment status
AX1030(config)#show pki cmp-cert log xxx ?
  follow      Display current running cmp-cert log
  from-start  Display cmp-cert log from the beginning
  num-lines   Number of lines displayed of most recent entries in the cmp-
cert log
  |           Output modifiers
```

The following is a list of show commands and examples for `show pki cmp-cert log`

- Show Last 10 Lines by Default
- Show 30 Lines
- Show from the Start of the Log File
- Follow the Log File Print any New Logs

Show Last 10 Lines by Default

```
AX1030(config-cmp cert:test-https)#show pki cmp-cert log test-https
CMP INFO: received 1 enrolled certificate, saving to file
'/a10data/cert/test-https'
return value 0
Client enrollment completed successfully
notify other modules for enrollment/update status
return value 0
successfully send enrollment/update status(success)
rotating file /a10data/cert/test-https
file /a10data/cert/test-https rotation completed
rotating file /a10data/key/test-https
file /a10data/key/test-https rotation completed
```

Show 30 Lines

```
AX1030(config-cmp cert:test-https)#show pki cmp-cert log test-https  num-
lines 30
return value 0
successfully send enrollment/update status(success)
rotating file /a10data/cert/test-https
file /a10data/cert/test-https rotation completed
rotating file /a10data/key/test-https
file /a10data/key/test-https rotation completed
Begin enrollment of certificate test-https on 05/09/2019 16:42:10
Generating private key
/a10/bin/openssl-cmp genrsa -out /a10data/key/test-https 8192
Generating RSA private key, 8192 bit long modulus (2 primes)
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+
.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.+.
```

```
e is 65537 (0x010001)
return value 0
Sending enrollment request
CMP INFO: sending ir
CMP INFO: got response
CMP INFO: sending certConf
CMP INFO: got response
CMP INFO: received 1 CA certificate, saving to file '/a10data/ca/tmp-
cmp/test-https'
CMP INFO: received 1 enrolled certificate, saving to file
'/a10data/cert/test-https'
return value 0
Client enrollment completed successfully
notify other modules for enrollment/update status
return value 0
successfully send enrollment/update status(success)
rotating file /a10data/cert/test-https
file /a10data/cert/test-https rotation completed
rotating file /a10data/key/test-https
file /a10data/key/test-https rotation completed
```

Show from the Start of the Log File

```
AX1030(config-cmp cert:test-https)#show pki cmp-cert log test-https from-
start
Begin enrollment of certificate test-https on 04/09/2019 22:15:28
received arguments /a10data/cert/test-https /a10data/key/test-https
https://192.168.90.141:8442/ejbca/publicweb/cmp/CMP_RA a10 CN=hku_
ra,O=MyOrganization,C=SE CN=Root dns=walter.a10tplab.com 1 384 cmp_ca.pem
0 120 2 /a10data shared 0 0 1
Generating private key
/a10/bin/openssl-cmp ecparam -name secp384r1 -genkey -noout -out
/a10data/key/test-https
return value 0
convert subject-dn CN=hku_ra,O=MyOrganization,C=SE to cmp-subject-dn
"/CN=hku_ra/O=MyOrganization/C=SE"
convert recipient-dn CN=Root to cmp-recipient-dn "/CN=Root"
Sending enrollment request
```

```
/a10/bin/openssl-cmp cmp -cmd ir -server 192.168.90.141:8442 -path
ejbca/publicweb/cmp/CMP_RA -secret pass:a10 -certout /a10data/cert/test-
https -cacertsout /a10data/ca/tmp-cmp/test-https -newkey
/a10data/key/test-https -subject "/CN=hku_ra/O=MyOrganization/C=SE" -
recipient "/CN=Root" -trusted /a10data/ca/cmp_ca.pem -tls_used
CMP INFO: sending ir
CMP INFO: got response
CMP INFO: sending certConf
CMP INFO: got response
CMP INFO: received 1 CA certificate, saving to file '/a10data/ca/tmp-
cmp/test-https'
CMP INFO: received 1 enrolled certificate, saving to file
'/a10data/cert/test-https'
return value 0
Client enrollment completed successfully
notify other modules for enrollment/update status
/a10/bin/comm_cmp -a -s -c test-https -p shared -r success
return value 0
successfully send enrollment/update status(success)
rotating file /a10data/cert/test-https
file /a10data/cert/test-https rotation completed
rotating file /a10data/key/test-https
file /a10data/key/test-https rotation completed
Begin enrollment of certificate test-https on 05/09/2019 15:39:34
copying certificate /a10data/cert/test-https to temporary location
/a10data/tmp/test-https.cert
copying key /a10data/key/test-https to temporary location
/a10data/tmp/test-https.key
received arguments /a10data/cert/test-https /a10data/key/test-https
https://192.168.90.141:8442/ejbca/publicweb/cmp/CMP_RA a10 CN=hku_
ra,O=MyOrganization,C=SE CN=Root email=abc@tplab.com 1 384 cmp_ca.pem 0
120 2 /a10data shared 0 0 1
Generating private key
/a10/bin/openssl-cmp ecparam -name secp384r1 -genkey -noout -out
/a10data/key/test-https
return value 0
convert subject-dn CN=hku_ra,O=MyOrganization,C=SE to cmp-subject-dn
"/CN=hku_ra/O=MyOrganization/C=SE"
convert recipient-dn CN=Root to cmp-recipient-dn "/CN=Root"
Sending enrollment request
```

```
/a10/bin/openssl-cmp cmp -cmd ir -server 192.168.90.141:8442 -path
ejbca/publicweb/cmp/CMP_RA -secret pass:a10 -certout /a10data/cert/test-
https -cacertsout /a10data/ca/tmp-cmp/test-https -newkey
/a10data/key/test-https -subject "/CN=hku_ra/O=MyOrganization/C=SE" -
recipient "/CN=Root" -trusted /a10data/ca/cmp_ca.pem -tls_used -sans
email=abc@tplab.com
CMP INFO: sending ir
CMP INFO: got response
CMP INFO: sending certConf
CMP INFO: got response
CMP INFO: received 1 CA certificate, saving to file '/a10data/ca/tmp-
cmp/test-https'
CMP INFO: received 1 enrolled certificate, saving to file
'/a10data/cert/test-https'
return value 0
Client enrollment completed successfully
notify other modules for enrollment/update status
/a10/bin/comm_cmp -a -s -c test-https -p shared -r success
return value 0
--MORE—
```

Follow the Log File Print any New Logs

```
AX1030(config-cmp cert:test-https)#show pki cmp-cert log test-https follow
….
```

GUI Configuration

This section describes how to configure CMP Certificate by using the ACOS GUI.

To create CMP Certificate:

1. Navigate to **ADC > SSL Management > CMP Certificates**.
2. Click **Create**.
   The **Create CMP Certificate** page is displayed.
3. Enter the following details to create CMP Certificate.

- Certificate Name
- Cert Type
- RSA Key Length
- CMP CA
- CMP Cert
- Subject Distinguished Name
- Subject Alternate Name
- Recipient Distinguished Name
- URL
- Log-Level
- Max Polltime
- Secret
- Allow Unprotected Errors
- Renew Before/Every

4. Click **Create**.
   CMP Certificate is created.

Licensing and Platforms

This section has the following sub-section:

- Upgrading or Downgrading Results

Upgrading or Downgrading Results

The following is a list of important points, referring to either upgrading or downgrading the system, as per the impact of this feature:

- The command is expected to be ported to all future releases.
- Upgrading is an expected and applicable mode, which is not an issue.
- Downgrading will wipe out the command and malfunctioning.

Limitations

The following is a list of limitations for this feature:

- Support for CMP over HTTP/HTTPs (CMP directly on top of the TCP layer is not supported).
- As of now, for the cert initial enrollment, support pre-shared is a secret.
- It does not support the authentication request by using an external entity cert.
- One cert can be either CMP or SCEP enrolled, but not both.
- Max # of CMP certs per partition are as the following:

	HW Box						vThunder
Memory	256	128	64	32	16	08	N/A
Max # of CMP Cert per Partition	1024	1024	512	512	256	128	128

Dependencies or Risks or Assumptions

The following is a list of dependencies or risks or assumptions for this feature.

- The requirement for CMPv2 client is to enroll the gateway and renew when the certs are nearing expiry.
- The workflow and the design of this feature are more like SCEP.

Configuring Automatic Certificate Management Environment (ACME) Certificates

ACME is used to obtain the certificates for websites (HTTPS). The purpose is to validate domain names for issuing certificates in the web PKI. This protocol is based on passing JSON-formatted messages over HTTPS and was designed by the Internet Security Research Group (ISRG) in RFC 8555 for their Let's Encrypt service. It also enables automating a few aspects of certificate management.

Using both Let’s Encrypt and the ACME protocol, you can set up an HTTPS server and automatically obtain a browser-trusted certificate. Generally, ACME client runs as an agent on a web server and supports ACOS to obtain the certificate and renew it.

The domain verification is done using challenge HTTP-01, for provisioning an HTTP resource under a well-known URI. The domain must be certificate's Common Name and the IP address must be mapped with the ACOS's virtual IP address. Also, in aVCS and VRRP-A deployment, aVCS master accepts the configuration, and the master syncs the configuration to the slave device(s). Data traffic is served by HA primary device(s), so HTTP-01 type challenge from the CA server is also served by HA primary device(s). If you do not want to use the default vrid(0), then you can configure it using VRID option.

To configure an ACME certificate, you need to first enable reply ACME HTTP-01 challenge for the CA server. Since the CA server verifies whether the ACME client controls the domain, on the ACOS side, you must manually configure reverse proxy.

NOTE: Currently, A10 supports only HTTP-01 challenge type. Additionally, the ACME protocol traffic follows the routing table rule and DNS setting on the shared partition.

Key Considerations:

- ACME certificate from the CA server goes into the data interface.
- ACME supports shared and L3V partitions. ACME traffic follows the routing table rule and DNS setting on the shared partition. Hence, the network interfaces, routing rules, and DNS must be configured correctly in the shared partition, as the ACME certificate is created in the L3V partition.

NOTE:

- Lets Encrypt or ACME feature is not supported in a VRRP-A standalone environment mode.
- The VIP must be publicly accessible for certificate validation. To configure the VIP IP as the same as the data interface on vThunder, use the **`use-if-ip`** option.

The following topics are included:

- Configuring ACME Certificate
- Viewing ACME Certificate
- Configuration Examples

Enrollment and Renewal Process

After you configure an ACME certificate for enrollment, ACOS performs the following steps:

- Generates an account key and registers this account with the CA server.
- Generates a domain private key and RSA or EC key with the specified key-length for the certificate.
- ACOS proves the CA server that the user domain is in control. If the domain is already verified, ACOS skips this step. If the domain is not verified yet, ACOS deploys the challenge and triggers the CA server to start the verification process.
- Certificate Issuance - Once the CA verification succeeds, the ACOS account key pair is authorized, requested, renewed, to create/send CSR and sign them with the authorized key pair. Then, ACOS downloads certificates for the domain from the CA server.
- Store the certificate. After successful verification of the response from the CA, ACOS accepts the certificate and stores it.
- Notify application layer - The client-SSL template would accept the ACME certificate. ACOS notifies the application that the certificate is ready for use.
- Schedule renewal - ACOS handles the automatic renewal of the certificate when it is about to expire. ACOS checks the expiration date, depending on the periodic renewal or a specific time before the certificate expires configurations.
- Rotate and store files - After certificate renewal, the old certificate and key files are still stored for any future reference. Old files are rotated and the new file replaces the existing files.

NOTE:

- After enrollment, the account-email and domain are already registered with CA. Hence, account-email and domain cannot be changed after certificate is enrolled. To change these, you must remove acme-cert and re-enroll.
- When the ACME enrollment or renewal fails, the scheduled renewal time continues as planned, and the next renewal time will be scheduled.

You can configure different ACME clients (agent-v1 or agent-v2) to perform the ACME enrollment or renewal. The default agent is agent-v1.

To configure agent-v1 as the ACME client to perform ACME enrollment, use the following command:

```
ACOS(config-acme cert:cert1)# acme-client agent-v1
```

You can also configure the number of concurrent ACME processes. The default and the maximum number of ACME concurrent processes depend on the number of control CPUs. To set the number of ACME concurrent processes, use the following command:

```
ACOS(config)# pki acme-concurrent-process-count <num>
```

If the number of control CPUs is greater than 12, the default and the maximum numbers of ACME concurrent processes are limited to the number of 12 control CPUs. For more information, see *Command Line Interface Reference*.

The formula: Default: (<# of ctrl CPU> * 5) - (<# of ctrl CPU> - 1)

Max: <# of default> + 2

Table 4 : Default and Maximum Numbers of ACME Concurrent Process

No. of control CPUs	Default # of concurrent process	Maximum # of concurrent process
1	5	7
2	9	11
3	13	15
4	17	19

No. of control CPUs	Default # of concurrent process	Maximum # of concurrent process
5	21	23
6	25	27
7	29	31
8	33	35
9	37	39
10	41	43
11	45	47
12	49	51
# > 12	49	51

ACME Directory URL

By default, Let's encrypt is used as CA server. Let's Encrypt have rate limits to ensure fair usage by as many people as possible.

- Staging URL - https://acme-staging-v02.api.letsencrypt.org/directory
- Production URL - https://acme-v02.api.letsencrypt.org/directory

NOTE: A10 strongly recommends user run with staging environment to test your configurations, then switch to production environment.

Configuring ACME Certificate

To configure ACME using the CLI:

1. Use the **`reply-acme-challenge`** option under HTTP virtual port.
2. Use the **`pki acme-cert`** command to create the certificate and change the CLI to edit it.
3. Use the **`account-email`**, **`cert-type`**, **`domain`**, and **`url`** commands.
4. (Optional) Configure additional parameters.

- Renew Intervals
- Log level
- Staging URL
- SAN domain
- VRID

5. Use the **enroll** command to begin the enrollment process for the certificate.

NOTE: Make sure that the HTTP port 80 is not blocked by the firewall. The virtual ports 80 replying ACME challenges must be up and running.

Viewing ACME Certificate

To view ACME information, use the **show pki acme-cert** command.

For more details about the ACME CLI commands, see *Command Line Reference Guide*.

Configuration Examples

- The following commands enroll the certificate with the Let's encrypt CA. You need to enroll each certificate only once. After a certificate is enrolled, ACOS uses ACME to administer the certificate. This includes renewing the certificate before it expires. You do not need to manually administer the certificates after you enroll them.

```
ACOS(config)# pki acme-cert test
ACOS(config-acme cert:test)# account-email test@url.com
ACOS(config-acme cert:test)# cert-type rsa
ACOS(config-acme cert:test)# domain test.com
ACOS(config-acme cert:test)# enroll
ACOS(config-acme cert:test)# run-with-staging-server
ACOS(config-acme cert:test)# exit
```

- The following commands show information about the certificate. You can view both the log and status of the certificate. Also, `show pki acme-cert log <cert-name>` can be used to display the detailed log of the ACME protocol happened during the ACME cert registration or update process.

```
ACOS(config)# show pki cert acme-test-cert status
```

```
Certificate name: test status: SUCCESS
     Renew every 2 minutes
     rotated files: 4
```

```
ACOS(config)# show pki cert acme-test-cert detail
Certificate:
    Data:
        Version: 3 (0x2)
        Serial Number: 1770931951 (0x698e46ef)
    Signature Algorithm: sha256WithRSAEncryption
        Issuer: CN=aws2020.ddns.net
        Validity
            Not Before: Oct 15 02:30:26 2019 GMT  ➔issued time
            Not After : Oct 14 02:30:26 2021 GMT ➔cert validity time
        Subject: CN=hku_ra, O=MyOrganization, C=SE
```

Certificate Validation

The CA establishes a trust point used to validate the certificates. The CA itself can be one or more certificate files in the chain, all the way to the root. The following command is to import the certificate to the CA:

```
import ca-cert AD03-CA certificate-type p7b use-mgmt-port
scp://user@172.16.101.227/home/user/AD03-CA-root.p7b
```

The certificate revocation list (CRL) lists the certificates that have been revoked by the CA based on the serial number.

The following topics are covered:

CRL Distribution Point

The distribution point can be obtained from either within the certificate - in this case no configuration is needed, or you could manually configure the distributed point. An example of a distribution point is as follows:

```
X509v3 CRL Distribution Points:
        Full Name:
                URI:ldap:///CN=AD03-
        CA, CN=AD03, CN=CDP, CN=Public%20Key%20Services, CN=Services,
CN=Configuration, DC=a10lab,
DC=com?certificationRevocationList?base?objectClass=cRLDistributionPoint
```

During the IKE negotiation when the certificate is being validated, the IKE daemon uses this URL to retrieve the certificate. This is done via LDAP or HTTP.

NOTE: The CRL can also be local.

Configuring CRL Distribution Point

If the distribution point is in the certificate, it can be obtained from the `show pki certificate` command. If not, the CRL can be configured using the *crl* command from the VPN revocation mode. The configuration allows for using HTTP.

crl {**pri** | **sec**} *filename*

where

pri *filename* – Primary CRL File Name or URL (http://www.example.com/ocsp) (only .der filetypes).

sec *filename* – Secondary CRL File Name or URL (http://www.example.com/ocsp) (only .der filetypes).

NOTE: The ACOS does not try to get the CRL file if it already has a valid cached CRL file.

Online Certificate Status Protocol (OCSP)

Online Certificate Status Protocol (OCSP) is a network component that provides certificate verification services.

You can use OCSP to verify client certificates for access to an HTTPS virtual port. ACOS uses OCSP to authenticate the client, instead of a CRL that is imported to the ACOS device. When this feature is configured, the ACOS device acts as an authentication client in relation to the OCSP server.

The following topics are covered:

Certificate Verification Process

1. ACOS sends the certificate that is sent by the client to an OCSP responder for validation. If the responder is a member of a service group, ACOS uses the configured load-balancing method to select a responder and sends the request to that responder.
2. The OCSP responder checks its CRL database to determine whether the certificate is still valid or has been revoked.
3. The responder sends the verification result to ACOS.
4. ACOS caches the response in one of the following ways:
   a. If the certificate is valid, ACOS completes the IKE session setup with the client.
   b. If the certificate is not valid, ACOS does not complete the IKE session setup with the client.

ACOS Verification of Replies from OCSP Responder

As part of the session setup with the OCSP responder, ACOS receives a copy of the responder's certificate. To ensure that the response from the OCSP responder is not spoofed, ACOS verifies the identity of the responder by checking the responder's certificate.

CA-signed Certificate

OCSP responder's certificate that is signed by a root CA.

Intermediate Certificate

OCSP responder's certificate that is signed by an intermediate CA.

You must import the certificate(s) to the ACOS device as part of the configuration for OCSP support.

Configuring OCSP

OCSP can be configured as below. The CA certificate is needed to validate the OCSP responder.

```
aam authentication server ocsp ocsp_serv
        url http://192.168.230.101:80/ocsp <=necessary
        responder-ca AD03-CA <=optional
        responder-cert 33 <=necessary
        exit-module
```

The “AD03-CA” and “33” are the names of the certificate in the `show pki ca` and `show pki cert` commands respectively.

```
vpn revocation training
        ca AD03-CA
!This CA file is the "remote-ca-cert" in the VPN configuration.
        ocsp pri ocsp_serv
#show vpn ocsp training
        subject: “DC=a10networks, DC=com, CN=training”
        issuer: “DC=com, DC=a10lab, CN=AD03-CA”
        validity: produced at Apr 17 00:23:50 UTC 2015
        usable till Apr 17 15:22:23 UTC 2015, ok (expires in 14 hours)
certificate status: GOOD
```

NOTE: If the certificate status is REVOKED, the IKE connection would not be established. If the OCSP response is previously cached, the server is not queried again.

Configure the CRL

1. Navigate to **Security** >> **IPsec VPN**.
2. Select the Settings tab, then click the “Create” button at the top right corner of this page.

3. Enter the Revocation Name.
4. Select the CA filename from the drop-down list.
5. Enter the primary and secondary CRL URL.
6. Select the primary and secondary OCSP authorization server from the drop-down lists.

Digital Certificate Fragmentation

Digital certificates used for authentication may be large, which may result in fragmentation.

You can configure jumbo fragmentation using the following command:

```
ACOS(config)# vpn jumbo-fragment
```

NOTE:

- This command is only required when the MTU is larger than 1500 bytes on the IKE gateway outgoing interface.
- If fragmentation is not used, IKE packets may be truncated that can be seen in the output of the `debug vpn level 4` command.

Configuring HW Crypto Engines to Accelerate IKE Performance

The HW crypto engines are used for the performance acceleration of IKE. Various activities such as the DH generate/compute, the RSA sign/verify, and so on are performed during the IPsec VPN IKE intercession. The OpenSSL APIs used for these processes in the VPN daemon results in poor performance.

Configuring Cavium Crypto Engines to Accelerate IKE Performance

As it needs the CPU to perform the computation, the software implementation needs more CPU time, and the performance is sluggish. To mitigate this risk and to enhance the IKE negotiation performance and to accelerate the DH generate/compute, the RSA sign/verify operations, it is proposed to use the Cavium N3/N5 card in Thunder.

The following topics are covered:

Limitations

The following is a list of limitations for this feature:

- The support is only for the Cavium N3 and the N5 cards.
- This feature only works with the IPsec hardware mode. In IPsec software mode the IKE hardware acceleration is not used.
- All the Cavium cores cannot be assigned to IPsec, as there is a minimum need of a few cores running on SSL microcode to perform the DH/RSA computation for IKE negotiation.
- The support condition applies only for the hardware acceleration for the DH generate/compute and the RSA sign/verify computation in this implementation.
- The Cavium Nitrox III MOD_EXP supports 17 to the 512-byte range (136 to 4096 bit), and OpenSSL is used for DH and RSA computation, when it is out of the range.
- For N5 card, the maximum length of MOD_EXP is 8192 bits.
- When the Cavium card is busy and the request queue is full, then the new IKE DH/RSA computation request to Cavium fails and it cannot be sent, resulting into an error return note and the negotiation failure status for the IKE.

CLI Configuration

The following topics are included:

Configuration Commands

To enable IKE HW acceleration, a new command is added. The IKE HW acceleration is disabled by default.

```
TH3040S-N5(config)#vpn ?
  asymmetric-flow-support      Support asymmetric flows pass through IPsec
SA
  fragment-after-encap         Fragment after adding IPsec headers
  ike-acc-enable               Enable IKE Acceleration by Cavium Nitrox card
  ike-gateway                  IKE-gateway settings
  ike-sa-timeout               Timeout IKE-SA in connecting state in
seconds (default 600s)
  ike-stats-global             IKE-stats-global statistic
  ipsec                        IPsec settings
  ipsec-error-dump             Support record the error ipsec cavium
information in dump file
  jumbo-fragment               Support IKE jumbo fragment packet
  nat-traversal-flow-affinity  Choose IPsec UDP source port based on port
of inner flow (only for A10 to A10)
  revocation                   IPsec VPN revocation settings
  sampling-enable              Enable baselining
  stateful-mode                VPN module will work in stateful mode and
create sessions
  tcp-mss-adjust-disable       Disable TCP MSS adjustment in SYN packet
```

NOTE: The acceleration for IKE HW can work, only when this command is configured, and there is available SSL Cavium core.

Show Commands

Adding a line in the `show vpn` output to show the current IKE HW acceleration state.

```
TH1040-30.20#show vpn
Partition shared
IKE Gateway total:   2
IPsec total:         3

IKE SA total:        1
```

```
IPsec SA total:      2

IPsec stateful mode
IPsec encryption mode: Hardware (1 devices)
IKE hardware acceleration: enabled
Crypto cores total:       32
Crypto cores assigned to IPsec: 20
Crypto memory percentage assigned to IPsec: 50
Crypto cores request error: 6523
Bad Context Pointer:      6523

IPsec passthrough traffic

HA standby drop:          0
```

Functionalities

The CLI command is added to enable the HW accelerate function, which is disabled by default. When the HW accelerate function is enabled and the availability of the SSL Cavium core, then the HW acceleration is performed for all the IKE negotiations in the system-wide.

For the DH/RSA computation, the SSL microcode of Cavium is used. As IPsec data plane traffic encap/decap need IPsec microcode, the need is to assign part of the cores for SSL and another part of the cores for IPsec.

By default, the Cavium cores load SSL microcode and work for SSL. By using the command "**`system ipsec crypto-core xxx`**" the user can assign the required number of cores for the IPsec.

During the system start-up, the IPsec microcode loads for a required number of cores and the SSL microcode for other cores in a single Cavium device.

To successfully execute this feature, the required number must be less than the total number of cores of a Cavium device.

User Stories

Based on various user stories, the following are the points to consider on this feature.

This feature is used for the enhancement of the IKE negotiation performance.

It reduces the time needed to bring up the IPsec SA.

It is valuable for all the IPsec user.

VRRP-A

For this feature, there is no compatibility issues. This feature must work under VRRP-A.

Licensing/Supported Platforms

For this feature, there is no impact/changes to the existing licensing support.

Deployment Modes

These topics provide examples of IPsec VPN deployment.

The following topics are covered:

IPsec VPN Deployment using CLI

Basic IPsec deployment requires only the following steps on each ACOS device:

1. Configure a tunnel interface.

```
[no] interface tunnel num
```

This command changes the CLI to the configuration level for the interface, where you can use the following command to add IP interfaces to the tunnel interface:

```
[no] ip address ipaddr {subnet-mask | /mask-length}
```

2. Create a VPN gateway configuration.

```
[no] vpn ike-gateway gateway-name
```

This command changes the CLI to the configuration level for the gateway.

Use the following command to configure preshared key for authentication:

```
[no] auth-method preshare-key string
```

Use the following command to specify the interface on this ACOS device to use as the local endpoint of the tunnel:

```
[no] local-address
        {ip | ipv6 ipaddr}
```

Use the following command to specify the interface on the peer VPN gateway to use as the remote endpoint of the tunnel:

```
[no] remote-address {dns domain-name | ip ipaddr}
```

Use the following command to configure the management interface to handle the traffic on VPN gateway for shared partition only:

```
[no] interface-management
```

3. Create an IPsec tunnel configuration.

```
[no] vpn ipsec tunnel-name
```

This command changes the CLI to the configuration level for the IPsec tunnel. At this level, use the following command to specify the IPsec gateway configuration to use for the local endpoint of the VPN tunnel:

```
[no] ike-gateway gateway-name
```

Use the following command to bind the tunnel interface to the VPN tunnel:

```
[no] bind tunnel num ipaddr
```

This tunnel processes traffic to the specified tunnel interface.The *num* option specifies the tunnel ID. This is the ID you assign to the tunnel when you configure it. The *ipaddr* specifies the next-hop IP address of the traffic to be processed by this tunnel.

NOTE: These steps do not include setting up routing. Dynamic or static routing is required to enable the ACOS device to send traffic to the VPN gateway at the remote end of the tunnel. To configure routing, see the *ACOS Series Administration and Configuration Guide*. These steps need to be mirrored on the peer ACOS device.

IPsec VPN Deployment using GUI

The following topic is covered:

Deploying Method

Basic IPsec deployment requires only the following steps on each ACOS device:

1. Configure a tunnel interface.
   a. Navigate to **Security** > **IPsec VPN**.
   b. Select the Interfaces tab, then click the “Create” button at the top right corner of this page.
   c. Enter the number of the tunnel interface, 1 - 128.
   d. Enter the name of the tunnel interface.
   e. Enter the load interval value, 5 - 300.

2. Create a VPN gateway configuration.
   a. Navigate to **Security** > **IPsec VPN**.
   b. Select the IKE gateways tab, then click the “Create” button at the top right corner of this page.
   c. Enter the name of the IKE Gateway.
   d. Select the Authorization Method, the supported values are preshare-key, rsa-signature, and ecdsa-signature.
   e. Select the local gateway address protocol and IP address need to be configured.
3. Create an IPsec tunnel configuration.
   a. Navigate to **Security** > **IPsec VPN**.
   b. Select the VPN Tunnels tab, then click Create.
   c. Enter the name of the VPN tunnel.
   d. Select the IKE Gateway from the drop-down list.
   e. Specify the Traffic Selector information of the networks, protocols, and ports, which are passing through the IPsec tunnel.
   f. Select the Interface Management to handle traffic on management interface.
   g. Select the Encryption algorithm, Hash value, Priority and click Apply.

NOTE: Bind the tunnel interface to the VPN tunnel.These steps do not include setting up routing. Dynamic or static routing is required to enable the ACOS device to send traffic to the VPN gateway at the remote end of the tunnel. To configure routing, see the *ACOS Series Administration and Configuration Guide*. These steps need to be mirrored on the peer ACOS device.

Bringing the Tunnel UP

The following topics are covered:

Manual

When there is no traffic, the IPsec tunnel can be brought up manually using the following command:

```
vpn ipsec <tunnel name>up
```

For more information about the `up` command, see *Command Line Interface Reference Guide*.

Automatic

When traffic matches the IPsec tunnel conditions, the tunnel is brought up automatically.

Bringing the Tunnel DOWN

The following commands are used to bring the IPsec tunnel DOWN:

- clear vpn ike-sa
- clear vpn ipsec-sa

NOTE: Every time you edit the configuration, the tunnel comes down automatically.

Configuring Maximum Transmission Unit (MTU) on the Tunnel Interface

The MTU on the tunnel interface should be configured to be the maximum size of the packet before the IPsec overhead is applied.

- Tunnel MTU = Maximum size of the inner packet
- Egress Interface MTU = Maximum size of outer packet (with all encapsulations and header overhead)

- Tunnel interface MTU <= Egress Interface MTU – Encapsulation overhead (approximately 150 bytes)

For example, if the egress is ethernet2 and MTU is 1500 bytes, the tunnel interface mtu should be1500-150 = 1350 bytes.

IPsec IPv6 Tunnel Deployment

For the deployment of the basic IPsec IPv6 Tunnel using the CLI mode, only the following steps are required on each ACOS device:

1. Configure a tunnel interface.

   ```
   [no] ACOS:# interface tunnel num
   ```

   This command changes the CLI to the configuration level for the interface, where you can use the following command to add IPv6 interfaces to the tunnel interface:

   ```
   [no] ACOS:# ipv6 address ipv6addr {subnet-mask | /mask-length}
   ```

   This command changes the CLI to the configuration level for the IPsec IPv6 address and enables an IPv6 address.

2. Create a VPN gateway configuration.

   ```
   [no] ACOS:# vpn ike-gateway gateway-name
   ```

   This command changes the CLI to the configuration level for the gateway.

   Use the following command to configure the pre-shared key for authentication:

   ```
   [no] ACOS:# auth-method preshare-key string
   ```

   Use the following command to specify the interface on this ACOS device to use as the local endpoint of the tunnel:

   ```
   [no] ACOS:# local-address {ipv6 ipaddr}
   ```

   Use the following command to specify the interface on the peer VPN gateway to use as the remote endpoint of the tunnel:

   ```
   [no] ACOS:# remote-address {dns domain-name | ipv6 ipaddr}
   ```

3. Create an IPsec IPv6 tunnel configuration.

   ```
   [no] ACOS:# vpn ipsec {ipv6 tunnel-name}
   ```

This command changes the CLI to the configuration level for the IPsec IPv6 tunnel. At this level, use the following command to specify the IPsec IPv6 tunnel configuration to use for the local endpoint of the VPN tunnel:

```
[no] ACOS:# ike-gateway gateway-name
```

Use the following command to bind the IPsec IPv6 tunnel interface to the VPN tunnel:

```
[no] ACOS:# bind tunnel num ipaddr
```

Use the following command to encrypt the IPsec IPv6 tunnel interface to the VPN tunnel:

```
(config:1-ipsec:1)#encryption ?
  des           Data Encryption Standard algorithm
  3des          Triple Data Encryption Standard algorithm
  aes-128       Advanced Encryption Standard algorithm CBC Mode(key size:
128 bits)
  aes-192       Advanced Encryption Standard algorithm CBC Mode(key size:
192 bits)
  aes-256       Advanced Encryption Standard algorithm CBC Mode(key size:
256 bits)
  aes-gcm-128   Advanced Encryption Standard algorithm Galois/Counter
Mode(key size: 128 bits, ICV size: 16 bytes)
  aes-gcm-192   Advanced Encryption Standard algorithm Galois/Counter
Mode(key size: 192 bits, ICV size: 16 bytes)
  aes-gcm-256   Advanced Encryption Standard algorithm Galois/Counter
Mode(key size: 256 bits, ICV size: 16 bytes)
  null          No encryption algorithm
```

This tunnel processes traffic to the specified tunnel interface.The *num* option specifies the tunnel ID.

This is the ID you assign to the tunnel when you configure it. The *ipaddr* specifies the next-hop IP address of the traffic to be processed by this tunnel.

Use the following command to assign and configure the IPsec IPv6 tunnel interface traffic selector to the VPN tunnel:

```
[no] ACOS:# vpn ipsec {traffic-selector ipv6}
```

NOTE: These steps do not include setting up routing.

Dynamic or static routing is required to enable the ACOS device to send traffic to the VPN gateway at the remote end of the tunnel.

These steps need to be mirrored on the peer ACOS device.

NOTE: To configure routing, see the *ACOS System Administration and Configuration Guide*.

NAT Before IPsec Tunnel

Network Address Translation (NAT) is the process where a network device assigns a public address to a device (or set of devices) inside a private network. The NAT is a cost-effective and secured solution to limit the number of public IP addresses in an organization.

The following topics are covered:

A10 Thunder® Convergent Firewall (CFW)

A10 Thunder® Convergent Firewall (CFW) is the first converged security solution for service providers, cloud providers, and large enterprises that includes integrated application delivery and security solutions in a single and standalone product. Thunder CFW includes all the features of Thunder ADC, CGN, and SSLi.

This section provides configuration procedures for A10 Thunder® CFW series NAT before IPsec Tunnel.

The following topics are covered:

Overview

Thunder CFW enables organizations to deploy IPsec site-to-site VPN solution and protect sensitive data transfer between remote sites. It demonstrates how to utilize NAT before IPsec tunnel to achieve same network from multiple sites to access HQ or Data Center internal network.

This situation may happen when “Company A” acquires “Company B”, and they both have same internal IP subnet. With NAT before IPsec Tunnel, HQ, or Data Center is able to distinguish the traffic from “Company A” or “Company B”.

Deployment Prerequisites

In this section, the following components are used to deploy the IPsec VPN solution between A10 Thunder CFW devices:

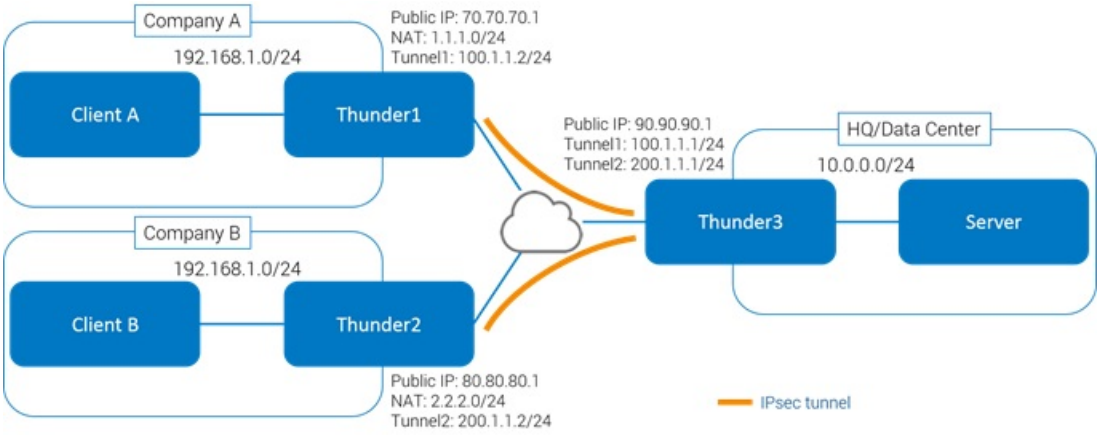
- A10 Thunder Convergent Firewall (CFW) hardware appliance
- A10 Networks Advanced Core Operating System (ACOS®)
- Internet access through a (gateway) router
- The side with source NAT requires SLB and VPN stateful mode

Network Topology

For this deployment procedure, the Thunder CFW device is deployed as a route-based L3 mode.

Interfaces on Thunder CFW are connected as follows:

Figure 19 : IPsec VPN Tunnel - A10 Thunder CFW Between Multi-sites



For simpler topology with only two sites, the illustration and configuration of “Company A” and “HQ/Data Center” can be applied.

Accessing A10 Thunder CFW

Thunder CFW can be accessed from a Command Line Interface (CLI):

Text-based interface in which you type commands on a command line.

You can access the CLI directly through the serial console or the network using either of the following protocols:

- Secure protocol - Secure Shell (SSH) version 2
- Unsecure protocol - Telnet (if enabled; not recommended)

Thunder CFW IPsec Configuration using CLI

The IPsec VPN can also be configured directly on the Thunder CFW through the Command Line Interface (CLI) of the device.

The following topics are covered:

Interface Configuration

Configure the following Internal and External interfaces on the Thunder CFW.

Thunder1

Table 5 : Configuring the Interface - Thunder1

```
Thunder1
ethernet1 IP: 70.70.70.1/24
ethernet2 IP: 192.168.1.1/24
tunnel1 IP: 100.1.1.2/24
```

Thunder1

```
interface ethernet 1
  enable
  ip address 70.70.70.1 /24
!
interface ethernet 2
  enable
  ip address 192.168.1.1 /24
  ip allow-promiscuous-vip
!
interface tunnel 1
  ip address 100.1.1.2 /24
!
```

Thunder2

Table 6 : Configuring the Interface - Thunder2

```
Thunder2
ethernet1 IP: 80.80.80.1/24
ethernet2 IP: 192.168.1.1/24
tunnel2 IP: 200.1.1.2/24
```

Thunder2

```
interface ethernet 1
  enable
```

```
  ip address 80.80.80.1 /24
!
interface ethernet 2
  enable
  ip address 192.168.1.1 /24
  ip allow-promiscuous-vip
!
interface tunnel 2
  ip address 200.1.1.2 /24
!
```

Thunder3

Table 7 : Configuring the Interface - Thunder3

```
Thunder3
ethernet1 IP: 90.90.90.1/24
ethernet2 IP: 10.0.0.1/24
tunnel1 IP: 100.1.1.1/24
tunnel2 IP: 200.1.1.1/24
```

Thunder3

```
interface ethernet 1
  enable
  ip address 90.90.90.1 /24
!
interface ethernet 2
  enable
  ip address 10.0.0.1 /24
!
interface tunnel 1
  ip address 100.1.1.1 /24
!
interface tunnel 2
  ip address 200.1.1.1 /24
!
```

IKE and IPsec Configuration

IPsec involves many component technologies and encryption methods. This step involves configuring matching IKE (Internet Key Exchange) and IPsec tunnel parameters, such as IKE Local ID, IKE Remote ID, and IPsec traffic selectors. IKE works in two steps, generally called IKE phases 1 and 2 as described in the following:

IKE Phase 1

IKE authenticates IPsec peers and negotiates IKE Security Association (IKE SAs) during this phase, setting up a secure channel for negotiating IPsec SAs in phase 2.

IKE Phase 2

IKE negotiates IPsec SA parameters and sets up matching IPsec SAs in the peers.

The following are the commands to configure IKE and IPsec:

Thunder1

```
vpn stateful-mode
!
vpn ike-gateway TH1_to_TH3
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 70.70.70.1
  remote-address ip 90.90.90.1
!
vpn ipsec Tunnel_TH1_to_TH3
  ike-gateway TH1_to_TH3
  encryption aes-128 hash sha1
  bind tunnel 1 100.1.1.1
!
```

Thunder2

```
vpn stateful-mode
!
vpn ike-gateway TH2_to_TH3
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
```

```
  local-address ip 80.80.80.1
  remote-address ip 90.90.90.1
!
vpn ipsec Tunnel_TH2_to_TH3
  ike-gateway TH2_to_TH3
  encryption aes-128 hash sha1
  bind tunnel 2 200.1.1.1
!
```

Thunder3

```
vpn ike-gateway TH3_to_TH1
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 90.90.90.1
  remote-address ip 70.70.70.1
!
vpn ike-gateway TH3_to_TH2
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 90.90.90.1
  remote-address ip 80.80.80.1
!
vpn ipsec Tunnel_TH3_to_TH1
  ike-gateway TH3_to_TH1
  encryption aes-128 hash sha1
  bind tunnel 1 100.1.1.2
!
vpn ipsec Tunnel_TH3_to_TH2
  ike-gateway TH3_to_TH2
  encryption aes-128 hash sha1
  bind tunnel 2 200.1.1.2
!
```

Routing Configuration

The step is to configure the static route parameters, as follows

Thunder1

```
!
```

```
ip nat pool natpool 1.1.1.1 1.1.1.255 netmask /24
!
ip route 10.0.0.0 /24 tunnel 1 100.1.1.1
!
```

Thunder2

```
!
ip nat pool natpool 2.2.2.1 2.2.2.255 netmask /24
!
ip route 10.0.0.0 /24 tunnel 2 200.1.1.1
!
```

Thunder3

```
!
ip route 1.1.1.0 /24 tunnel 1 100.1.1.2
!
ip route 2.2.2.0 /24 tunnel 2 200.1.1.2
!
```

SLB Configuration

The final step is to configure the slb parameters and only apply to Thunder1 and Thunder2, vip is the peer tunnel IP address, as follows:

Thunder1

```
!
slb server vip 100.1.1.1
  port 0 tcp
    health-check-disable
!
slb service-group sg1 tcp
  member vip 0
!
slb virtual-server wildcard 0.0.0.0
  port 0 tcp
    source-nat pool natpool
    service-group sg1
    no-dest-nat
```

```
!
```

Thunder2

```
!
slb server vip 200.1.1.1
  port 0 tcp
    health-check-disable
!
slb service-group sg1 tcp
  member vip 0
!
slb virtual-server wildcard 0.0.0.0
  port 0 tcp
    source-nat pool natpool
    service-group sg1
    no-dest-nat
!
```

Viewing SLB Status on A10

Once the configurations are done, use the following commands to check the SLB status on Thunder CFW:

The following topics are covered:

Show SLB Server Status

```
Thunder#show slb server
Total Number of Servers configured: 1
Total Number of Services configured: 1
                  Current = Current Connections, Total = Total
Connections
                  Fwd-pkt = Forward packets, Rev-pkt = Reverse packets
Service                  Current    Total      Fwd-pkt    Rev-pkt
Peak-conn  State
```

```
-------------------------------------------------------------------------------
-------------
vip:0/tcp                  0          2          13         8          0
      Up
vip: Total                 0          2          13         8          0
      Up
Thunder#
```

Show SLB Virtual-Server Status

```
Thunder#show slb virtual-server
Total Number of Virtual Services configured: 1
Virtual Server Name      IP              Current    Total      Request
Response Peak
Service-Group            Service         connection connection packets
packets  connection
-------------------------------------------------------------------------------
--------------
*wildcard 0.0.0.0         All Up
    port 0  tcp                          0          2          13       8
      0
sg1                      0/tcp           0          2          13       8
      0
Total received conn attempts on this port: 2
Thunder#
```

Limitations

The following is the limitation for this feature:

The side with source NAT requires SLB and VPN stateful mode.

Configuring Thunder CFW CLI

Thunder1

```
interface ethernet 1
  enable
  ip address 70.70.70.1 255.255.255.0
```

```
!
interface ethernet 2
  enable
  ip address 192.168.1.1 /24
  ip allow-promiscuous-vip
!
interface tunnel 1
  ip address 100.1.1.2 /24
!
ip nat pool natpool 1.1.1.1 1.1.1.255 netmask /24
!
ip route 10.0.0.0 /24 tunnel 1 100.1.1.1
!
slb server vip 100.1.1.1
  port 0 tcp
    health-check-disable
!
slb service-group sg1 tcp
  member vip 0
!
slb virtual-server wildcard 0.0.0.0
  port 0 tcp
    source-nat pool natpool
    service-group sg1
    no-dest-nat
!
vpn stateful-mode
!
vpn ike-gateway TH1_to_TH3
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 70.70.70.1
  remote-address ip 90.90.90.1
!
vpn ipsec Tunnel_TH1_to_TH3
  ike-gateway TH1_to_TH3
  encryption aes-128 hash sha1
  bind tunnel 1 100.1.1.1
!
```

Thunder2

```
interface ethernet 1
  enable
  ip address 80.80.80.1 /24
!
interface ethernet 2
  enable
  ip address 192.168.1.1 /24
  ip allow-promiscuous-vip
!
interface tunnel 2
  ip address 200.1.1.2 /24
!
ip nat pool natpool 2.2.2.1 2.2.2.255 netmask /24
!
ip route 10.0.0.0 /24 tunnel 2 200.1.1.1
!
slb server vip 200.1.1.1
  port 0 tcp
    health-check-disable
!
slb service-group sg1 tcp
  member vip 0
!
slb virtual-server wildcard 0.0.0.0
  port 0 tcp
    source-nat pool natpool
    service-group sg1
    no-dest-nat
!
vpn stateful-mode
!
vpn ike-gateway TH2_to_TH3
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 80.80.80.1
  remote-address ip 90.90.90.1
!
vpn ipsec Tunnel_TH2_to_TH3
  ike-gateway TH2_to_TH3
  encryption aes-128 hash sha1
```

```
  bind tunnel 2 200.1.1.1
!
```

Thunder3

```
interface ethernet 1
  enable
  ip address 90.90.90.1 /24
!
interface ethernet 2
  enable
  ip address 10.0.0.1 /24
!
interface tunnel 1
  ip address 100.1.1.1 /24
!
interface tunnel 2
  ip address 200.1.1.1 /24
!
ip route 1.1.1.0 /24 tunnel 1 100.1.1.2
!
ip route 2.2.2.0 /24 tunnel 2 200.1.1.2
!
vpn ike-gateway TH3_to_TH1
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 90.90.90.1
  remote-address ip 70.70.70.1
!
vpn ike-gateway TH3_to_TH2
  auth-method preshare-key A10networks
  encryption aes-128 hash sha1
  local-address ip 90.90.90.1
  remote-address ip 80.80.80.1
!
vpn ipsec Tunnel_TH3_to_TH1
  ike-gateway TH3_to_TH1
  encryption aes-128 hash sha1
  bind tunnel 1 100.1.1.2
!
vpn ipsec Tunnel_TH3_to_TH2
```

```
ike-gateway TH3_to_TH2
encryption aes-128 hash sha1
bind tunnel 2 200.1.1.2
```

IPsec VPN Configuration Examples

The following topics are covered:

Overview

The first topology example shows a a single tunnel between two sites. The second topology example builds on the first example, by adding a second VPN tunnel between the two sites and a third site.

Single Tunnel Deployment

Before deploying IPsec VPN, decide on the protocols you want to use for IKE and IPsec SAs and identify the IP addresses of the traffic that needs to be encrypted. The configurations to deploy a single IPsec tunnel over two sites.

The following topics are covered:

Traffic Selectors

- Local 192.168.20.0 /24
- Remote 192.168.30.0 /24

Route

- Static

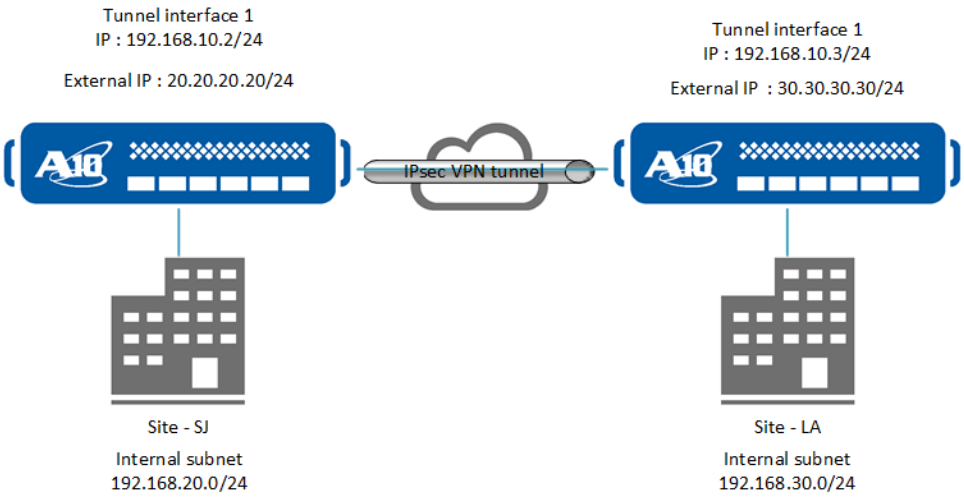
IKE Phase 1

- Hashing algorithm: MD5
- Authentication method: Preshared key
- DH group: 1
- Encryption algorithm: DES
- Lifetime: Default

IKE Phase 2

- HMAC: SHA-1
- DH group: 1
- Encryption algorithm: AES-128
- Lifetime: Default

Figure 20 : Single Tunnel IPsec Deployment



Configuring A Tunnel Interface

The following topics are covered:

Site-SJ Configuration

Enter the following commands to configure the tunnel interface:

```
ACOS(config)#interface tunnel 1
ACOS(config-if:tunnel1)#ip address 192.168.10.2 /24
ACOS(config-if:tunnel1)#exit
```

The following command configures a static route that is used in the tunnel interface:

```
ACOS(config)#ip route 192.168.30.0 /24 tunnel 1 192.168.10.3
```

The destination network of the route is 192.168.30.0/24, an internal network at Site-LA. The next hop of the route is the IP address of the remote end of the tunnel, on the peer VPN device.

Configuring VPN Gateway Settings

The following commands configure VPN gateway settings:

```
ACOS(config)#vpn ike-gateway SJ_to_LA_GW
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#dh-group 1
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#auth-method preshare-key 123456
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#encryption des hash md5 priority
5
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#local-id Site-SJ
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#remote-id Site-LA
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#local-address ip 20.20.20.20
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#remote-address ip 30.30.30.30
ACOS(config-vpn ike-gateway:SJ_to_LA_GW)#exit
```

The *preshare-key* option of the `auth-method` command specifies the preshared key to use during IKE phase 1. The same preshared key string must also be configured on the peer. The encryption command configures an encryption group for gateway traffic.

The `local-id` command specifies the ID value of this gateway for IKE phase 1. Likewise, the `remote-id` command specifies the name of the peer.

The `local-address` and `remote-address` commands specify the local IP and peer IP addresses for IKE and IPsec. These are globally unique, externally routable IP addresses.

NOTE: Dynamic `remote-address` and `remote-id` for the IPsec VPN peers are supported only with IKEv2. If the `remote-address` or `remote-id` is dynamic, define only the local-address or local-id in the configuration. The remote device must initiate the tunnel as the local device will not know the remote device's address.

Configuring VPN Tunnel Settings

The following topics are covered:

Configuration Method

Enter the following commands to configure VPN tunnel settings:

```
ACOS(config)#vpn ipsec SJ_to_LA_TUN1
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#ike-gateway SJ_to_LA_GW
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#dh-group 1
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#encryption aes-128 hash sha1 priority
5
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#traffic-selector ipv4 local
192.168.20.0 /24 remote 192.168.30.0 /24
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#traffic-selector ipv6 localv6
192.168.20.0 /24 remote 192.168.30.0 /24
ACOS(config-vpn ipsec:SJ_to_LA_TUN1)#bind tunnel 1 192.168.10.3
```

The `ike-gateway` command specifies the gateway configuration to use (the one configured above).

The `dh-group` command enables Perfect Forward Secrecy (PFS). The encryption command configures an encryption group for tunnel traffic.

The `traffic-selector` command specifies the local and remote internal subnets to be joined by the tunnel. For route-based VPN traffic selectors need not be specified.

If not configured by default, it is going to be wild-card addresses. If the peer device is not an ACOS device and it specifies subnets then traffic selectors have to be configured otherwise there would be a policy mismatch and the tunnel would not come up.

The `bind` command binds the VPN tunnel configuration to the tunnel interface on the peer.

The configuration without the CLI prompts are shown below:

```
!This is the logical interface and the IP address represents the network
for the tunnel.
interface tunnel 1
ip address 192.168.10.2 /24
!Represents the next hop to the tunnel.
ip route 192.168.30.0 /24 tunnel 1 192.168.10.3
!IKE gateway configuration includes the authentication method, encryption
algorithms, local and remote ids and IP addresses.
vpn ike-gateway SJ_to_LA_GW
        auth-method preshare-key 123456
        encryption des hash md5 priority 5
        local-id Site-SJ
        remote-id Site-LA
        local-address ip 20.20.20.20
        remote-address ip 30.30.30.30
!IPsec tunnel configuration specifies the IKE gateway tunnel interface and
encryption algorithms.
vpn ipsec SJ_to_LA_TUN1
        ike-gateway SJ_to_LA_GW
        dh-group 1
        encryption aes-128 hash sha1 priority 5
        traffic-selector ipv4 local 192.168.20.0 /24 remote 192.168.30.0 /24
!Bind the IKE gateway and tunnel interface. The IP address has to match
the configured nexthop in the route.
        bind tunnel 1 192.168.10.3
```

Site-LA Configuration

Enter the following mirrored configuration for the Site-LA VPN device:

```
interface tunnel 1
ip address 192.168.10.3 /24
```

```
!
ip route 192.168.20.0 /24 tunnel 1 192.168.10.2
!
vpn ike-gateway LA_to_SJ_GW
        auth-method preshare-key 123456
        encryption des hash md5 priority 5
        local-id Site-LA
        remote-id Site-SJ
        local-address ip 30.30.30.30
        remote-address ip 20.20.20.20
!
vpn ipsec LA_to_SJ_TUN1
        ike-gateway LA_to_SJ_GW
        dh-group 1
        encryption aes-128 hash sha1 priority 5
        traffic-selector ipv4 local 192.168.30.0 /24 remote 192.168.20.0 /24
        bind tunnel 1 192.168.10.2
```

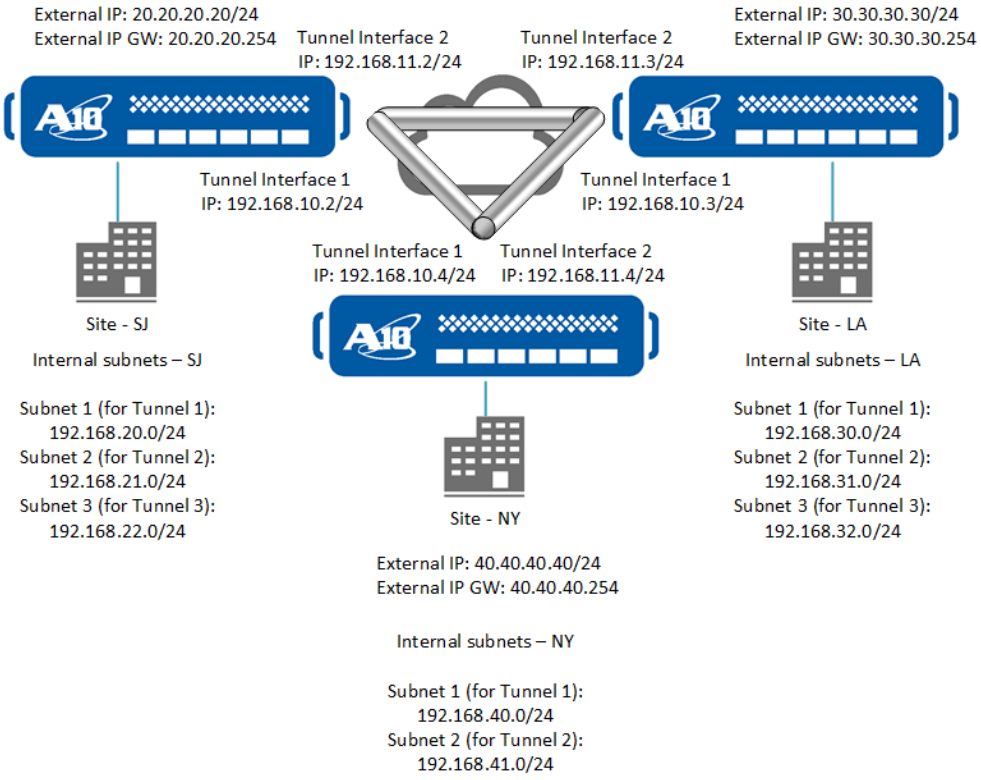
Multiple Tunnel Deployment

The following topics are covered:

Overview

The following commands deploy multiple IPsec tunnels over three sites:

Figure 21 : Multi-tunnel IPsec Deployment



SJ-LA Second Tunnel

The following topics are covered:

Site-SJ Configuration

Enter the following configuration for the Site-SJ VPN device:

```
interface tunnel 2
ip address 192.168.11.2 /24
!
ip route 192.168.31.0 /24 tunnel 2 192.168.11.3
!
vpn ipsec SJ_to_LA_TUN2
```

```
ike-gateway SJ_to_LA_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.21.0 /24 remote 192.168.31.0 /24
bind tunnel 2 192.168.11.3
```

Site-LA Configuration

Enter the following configuration for the Site-LA VPN device:

```
interface tunnel 2
ip address 192.168.11.3 /24
!
ip route 192.168.21.0 /24 tunnel 2 192.168.11.2
!
vpn ipsec LA_to_SJ_TUN2
ike-gateway LA_to_SJ_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.31.0 /24 remote 192.168.21.0 /24
bind tunnel 2 192.168.11.2
```

SJ-LA-NY Tunnel

The following topics are covered:

Enter the following commands to configure the tunnels between Site-NY, and sites Site-SJ and Site-LA:

Site-SJ Configuration

```
ip route 192.168.40.0 /24 tunnel 1 192.168.10.4
!
vpn ike-gateway SJ_to_NY_GW
auth-method preshare-key 123456
encryption des hash md5 priority 5
```

```
local-id Site-SJ
remote-id Site-NY
local-address ip 20.20.20.20
remote-address ip 40.40.40.40
!
vpn ipsec SJ_to_NY_TUN1
ike-gateway SJ_to_NY_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.22.0 /24 remote 192.168.40.0 /24
bind tunnel 1 192.168.10.4
```

Site-LA Configuration

```
ip route 192.168.41.0 /24 tunnel 1 192.168.10.4
!
vpn ike-gateway LA_to_NY_GW
auth-method preshare-key 123456
encryption des hash md5 priority 5
local-id Site-LA
remote-id Site-NY
local-address ip 30.30.30.30
remote-address ip 40.40.40.40
!
vpn ipsec LA_to_NY_TUN1
ike-gateway LA_to_NY_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.32.0 /24 remote 192.168.41.0 /24
bind tunnel 1 192.168.10.4
```

Site-NY Configuration

```
interface tunnel 1
ip address 192.168.10.4 /24
!
ip route 192.168.22.0 /24 tunnel 1 192.168.10.2
ip route 192.168.32.0 /24 tunnel 1 192.168.10.3
!
vpn ike-gateway NY_to_SJ_GW
auth-method preshare-key 123456
```

```
encryption des hash md5 priority 5
local-id Site-NY
remote-id Site-SJ
local-address ip 40.40.40.40
remote-address ip 20.20.20.20
!
vpn ipsec NY_to_SJ_TUN1
ike-gateway NY_to_SJ_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.40.0 /24 remote 192.168.22.0 /24
bind tunnel 1 192.168.10.2
!
vpn ike-gateway NY_to_LA_GW
auth-method preshare-key 123456
encryption des hash md5 priority 5
local-id Site-NY
remote-id Site-LA
local-address ip 40.40.40.40
remote-address ip 30.30.30.30
!
vpn ipsec NY_to_LA_TUN1
ike-gateway NY_to_LA_GW
dh-group 1
encryption aes-128 hash sha1 priority 5
traffic-selector ipv4 local 192.168.41.0 /24 remote 192.168.32.0 /24
bind tunnel 1 192.168.10.3
!
```

IPv6 in IPv4 IPsec Tunnel

IPv6 packets over IPv4 IPsec tunnel is supported.

NOTE: Configure the `frag-after-encap` command. The default setting of `fragment-before-encap` is not supported for IPv6 in IPv4 IPsec Tunnel mode.

The following topics are covered:

Sample Configuration

This release provides support for configuring a tunnel interface as a nexthop for IPv6 routes.

```
ACOS#show run interface tunnel 2
!Section configuration: 134 bytes
!
interface tunnel 2
  mtu 9000
  enable
  speed 5
  ipv6 address 2401:f000:82:1501::1/64
!
ACOS#show run | sec vpn01
vpn ike-gateway vpn01
  auth-method preshare-key a10networks
  encryption aes-256 hash sha256 priority 1
  local-address ip 12.12.12.1
  remote-address ip 11.11.11.1
vpn ipsec vpn01-ipsec64
  ike-gateway vpn01
  encryption aes-256 hash sha256 priority 1
  traffic-selector ipv6 localv6 11:11::/64 remotev6 22:22::/64
  bind tunnel 1 2401:f000:81:1501::2
```

IPv6 in IPv6 IPsec Tunnel Configuration

The following is a set of example for the IPv6 in IPv6 IPsec Tunnel Configuration:

ACOS-1

```
vlan 7
  tagged ethernet 1
  router-interface ve 7
!
```

```
interface ve 7
  ipv6 address 7:7:7::1/64
!
interface tunnel 1
  ipv6 address 64:64:64::1/64
  ipv6 enable
!
vpn ike-gateway v6
  auth-method preshare-key A10Networks
  local-address ipv6 7:7:7::1
  remote-address ipv6 7:7:7::2
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 52:52:52::/64 remotev6 55:55:55::/64
  bind tunnel 1 64:64:64::2
```

ACOS-2

```
vlan 7
  tagged ethernet 1
  router-interface ve 7
!
interface ve 7
  ipv6 address 7:7:7::2/64
!
interface tunnel 1
  ipv6 address 64:64:64::2/64
  ipv6 enable
!
vpn ike-gateway v6
  auth-method preshare-key A10Networks
  local-address ipv6 7:7:7::2
  remote-address ipv6 7:7:7::1
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 55:55:55::/64 remotev6 52:52:52::/64
  bind tunnel 1 64:64:64::1
!
```

IPsec Management over VPN

Based on the two site IPsec topologies, an internal network in LA can manage the ACOS device in SJ, and an internal network in SJ can manage the ACOS device in LA. For example, a client IP address in SJ's internal network (IP: 192.168.21.21) can manage the LA ACOS device on the tunnel interface (IP: 192.168.11.3), or a client IP address in LA's internal network (IP: 192.168.31.31) can manage the SJ ACOS device on the tunnel interface (IP: 192.168.11.2).

Client-to-Site VPN Support with IKE Configuration Payload

The following topics are covered:

Feature Description

The Client-to-Site VPN feature simplifies the configuration.

It can be further explained as the following.

- Using this Client-to-Site, a single **`ike-gateway`** and **`ipsec`** configuration object pair are configured, and each VPN client can connect to the same configuration object pair.
- The ACOS device no longer needs peer information such as IP address or ID to establish the tunnel.
- Additionally, coupled with the configuration payload feature, the ACOS device can exchange configuration information between itself and the VPN clients.

- Some common configuration information exchanged as the internal IP address, internal netmask, and so on.
- To use the configuration payload, the ACOS device connects to a DHCP or a RADIUS server to retrieve the configuration information.

Assumptions

The maximum support number of IPsec tunnels includes both Client-to-Site and Site-to-Site IPsec tunnels.

CLI Configuration Commands

The following topics are covered:

Requirements

The Client-to-Site VPN allows for any connection destined to the local-address IP to establish a VPN tunnel. The following three configuration requirements for the Client-to-Site VPN are needed:

1. The **`remote-address ip`** in the **`ike-gateway object`** is not configured.
2. The remote traffic selector in the **`ipsec object`** is set to `0.0.0.0 /0`.
3. The **`bind tunnel`** in **`ipsec object`** refers to a tunnel interface number without a next hop.
4. To combine the Client-to-Site with IP assignment with the configuration payload, the following five configuration requirements are needed:
5. The **`remote-address ip`** in the **`ike-gateway object`** is not configured.
6. The **`configuration-payload flag`** must be set on **`ike-gateway`**.
7. The DHCP server or RADIUS AAM server must be configured on **`ike-gateway`**.

8. The remote traffic selector in the **ipsec object** is set to **ip-assigned**.
9. The **bind tunnel** in **ipsec object** refers to a tunnel interface number without a next hop.

The following are a few examples of the Client-to-Site and Client-to-Site with Configuration Payload (DHCP and RADIUS).

Site Config(no configuration payload)

```
vpn ike-gateway site1
  auth-method rsa-signature
  key ax3030_rsa_1.key
  local-cert ax3030_rsa_1.crt
  local-id "C=US, ST=CA, L=SJ, O=A10, OU=Security, CN=ax3030_rsa_1"
  encryption aes-256 hash sha256
  local-address ip 70.70.70.186

vpn ipsec site1_ipsec
  ike-gateway site1
  encryption aes-256 hash sha256
  traffic-selector ipv4 local 1.1.1.0 255.255.255.0 remote 0.0.0.0 0.0.0.0
  bind tunnel 1
```

Site Config(dhcp)

```
vpn ike-gateway site1
  auth-method rsa-signature
  key ax3030_rsa_1.key
  local-cert ax3030_rsa_1.crt
  local-id "C=US, ST=CA, L=SJ, O=A10, OU=Security, CN=ax3030_rsa_1"
  encryption aes-256 hash sha256
  local-address ip 70.70.70.186
  configuration-payload dhcp
  dhcp-server pri <DHCP_SERVER PRI>
  dhcp-server sec <DHCP_SERVER SEC>

vpn ipsec site1_ipsec
  ike-gateway site1
  encryption aes-256 hash sha256
  traffic-selector ipv4 local 1.1.1.0 255.255.255.0 remote ip-assigned
  bind tunnel 1
```

Site Config(radius)

```
vpn ike-gateway site1
   auth-method eap-radius
   key ax3030_rsa_1.key
   local-cert ax3030_rsa_1.crt
   local-id "C=US, ST=CA, L=SJ, O=A10, OU=Security, CN=ax3030_rsa_1"
   encryption aes-256 hash sha256
   local-address ip 70.70.70.186
   configuration-payload radius
   radius-server pri <RADIUS_SERVER PRI>
   radius-server sec <RADIUS_SERVER_SEC>

vpn ipsec site1_ipsec
  ike-gateway site1
  encryption aes-256 hash sha256
  traffic-selector ipv4 local 1.1.1.0 255.255.255.0 remote ip-assigned
        bind tunnel 1
```

Show Commands

The following are the show commands that are needed for operation support (new and existing) for this feature.

The following topics are covered:

show vpn ike-sa

Description This command shows all the IKE SA. It shows the Site-to-Site and all Client-to-Site SA.

Example `AX3200-12-186#show vpn ike-sa`

```
AX3200-12-186#show vpn ike-sa

Name             Local/Peer IP Enc/Hash  Lifetime  Status     Auth-method
-------------------------------------------------------------------------
----------------
s2s_1            9.12.12.186 aes-128     73317s  Established  preshare-key
```

```
                  9.12.12.183                          sha1

s2s_2             11.12.12.186 aes-128     85983s  Established  preshare-
key
                  11.12.12.183                          sha1

site              8.8.8.186 aes-128        384s    Established  preshare-
key
                  8.8.8.128                             sha256

site              8.8.8.186 aes-128        253s    Established  preshare-
key
                  8.8.8.124                             sha256

-------------------------------------------------------------------------
----------------
Total: 4
AX3200-12-186#
```

show vpn ike-sa <NAME>

Description This command shows IKE SAs associated with a config (name).

For Site-to-Site, it shows a single SA. For Client-to-Site it shows all SAs which are associated.

Example

```
AX3200-12-186#show vpn ike-sa site

Gateway Name:  site
Initiator SPI: 0xb9ce137bf3559519                    Responder SPI:
0x968119a4bc390331
Local IP:      8.8.8.186                             Remote IP:
8.8.8.128
Local ID:      tuy.site                              Remote ID:
tuy.client
Encryption:    aes-128                               Hash:
sha256
Lifetime:      358s                                  Auth-method:
preshare-key
```

```
DH group:       2                                       NAT-T:
disabled
Status:         Established

Gateway Name:   site
Initiator SPI:  0x02a6590d42215352                      Responder SPI:
0x953c38b91b1e0880
Local IP:       8.8.8.186                               Remote IP:
8.8.8.124
Local ID:       tuy.site                                Remote ID:
tuy.124.client
Encryption:     aes-128                                 Hash:
sha256
Lifetime:       202s                                    Auth-method:
preshare-key
DH group:       2                                       NAT-T:
disabled
Status:         Established
```

show vpn ike-sa <NAME> remote-ip <IP>

This is a new/highlighted command.

Description This command shows IKE SAs based on the name + remote IP filter.

Example

```
AX3200-12-186#show vpn ike-sa site remote-ip 8.8.8.124

Gateway Name:   site
Initiator SPI:  0xd251527bbecd3427  Responder SPI:  0x0204d047870db5a3
Local IP:       8.8.8.186           Remote IP:      8.8.8.124
Local ID:       tuy.site            Remote ID:      tuy.124.client
Encryption:     aes-128             Hash:           sha256
Lifetime:       371s                Auth-method:    preshare-key
DH group:       2                   NAT-T:          disabled
Status:         Established
```

show vpn ike-sa <NAME> remote-id <ID>

This is a new/highlighted command.

Description This command shows IKE SAs based on the name + remote ID filter.

Example

```
AX3200-12-186#show vpn ike-sa site remote-id tuy.client

Gateway Name:  site
Initiator SPI: 0x34cf35ff00edbf38    Responder SPI: 0xca6b2d7a84f1721e
Local IP:      8.8.8.186             Remote IP:     8.8.8.128
Local ID:      tuy.site              Remote ID:     tuy.client
Encryption:    aes-128               Hash:          sha256
Lifetime:      289s                  Auth-method:   preshare-key
DH group:      2                     NAT-T:         disabled
Status:        Established
```

show vpn ipsec-sa

Description This command shows all IPsec SAs (both Client-to-Site and Site-to-Site).

Example

```
\AX3200-12-186#show vpn ipsec-sa
Gateway:s2s_1    Local IP:9.12.12.186    Remote IP:9.12.12.183
Name             Selectors In/Out SPI  Mode/xform    Time/Bytes
-------------------------------------------------------------------------
----------------
s2s_1            10.13.13.186/32              0x668f6cac  esp-tunnel        28787s
                 10.13.13.183/32 0x577d1ee7  aes-128-sha1       9.76G

Gateway:s2s_2    Local IP:11.12.12.186 Remote IP:11.12.12.183
Name             Selectors In/Out SPI  Mode/xform    Time/Bytes
-------------------------------------------------------------------------
----------------
s2s_2            11.13.13.186/32 0x668f6cad  esp-tunnel        28791s
                 11.13.13.183/32 0x577d1ee8  aes-128-sha1       9.76G

Gateway:site    Local IP:8.8.8.186    Remote IP:0.0.0.0
Name             Selectors In/Out SPI  Mode/xform    Time/Bytes
-------------------------------------------------------------------------
----------------
site             70.70.70.0/24 0x668f6ca4  esp-tunnel         3692s
                 7.7.7.10/32 0xcabf72e2  aes-128-sha256      9.76G
site             70.70.70.0/24 0x668f6cab  esp-tunnel         5984s
                 7.7.7.11/32 0xc2fa2962  aes-128-sha256      9.76G
```

```
AX3200-12-186#
```

show vpn ipsec-sa <NAME>

Description This command shows IPsec SAs associated with a config (name).

For Site-to-Site, only 1 SA is shown.

For Client-to-Site, all SAs associated is shown.

Example

```
AX3200-12-186#show vpn ipsec-sa site

SA Index: 5  Tunnel Name: site
Local IP: 70.70.70.0/24  Peer IP: 7.7.7.11/32
Local SPI: 0x668f6cab    Remote SPI: 0xc2fa2962
Protocol: esp            Mode: tunnel
Encryption Algorithm: aes-128 Hash Algorithm: sha256
DH Group: 2   NAT-T: disabled
Anti-Replay: disabled
Encrypted Packets: 0    Encrypted Bytes: 0
Decrypted Packets: 0    Decrypted Bytes: 0
Anti-Replay Failure: 0  Rekey Times: 6
Pad Check Error: 0       ICV Check Error: 0
Next Header Check Error: 0
Sequence Number: 0 Sequence Number Rollover: 0
Lifetime: 5840        Lifebytes: 9.76G
Pre-frag Success: 0 Pre-frag Error: 0
Frag-after-encap Fragment Generated: 0 Fragment Received: 0


SA Index: 4  Tunnel Name: site
Local IP: 70.70.70.0/24    Peer IP: 7.7.7.10/32
Local SPI: 0x668f6ca4      Remote SPI: 0xcabf72e2
Protocol: esp              Mode: tunnel
Encryption Algorithm: aes-128   Hash Algorithm: sha256
DH Group: 2          NAT-T: disabled
Anti-Replay: disabled
Encrypted Packets: 0    Encrypted Bytes: 0
Decrypted Packets: 0    Decrypted Bytes: 0
Anti-Replay Failure: 0  Rekey Times: 6
```

```
Pad Check Error: 0 ICV Check Error: 0
Next Header Check Error: 0
Sequence Number: 0 Sequence Number Rollover: 0
Lifetime: 3548     Lifebytes: 9.76G
Pre-frag Success: 0 Pre-frag Error: 0
Frag-after-encap Fragment Generated: 0 Fragment Received: 0
```

show VPN IPsec-SA <NAME> in-spi/out-spi <SPI>

This is a new/highlighted command.

Description This command shows IPsec SAs based on filter Name + In-SPI/Out-SPI

Example

```
AX3200-12-186#show vpn ipsec-sa site in-spi 0x668f6ca4

SA Index: 4    Tunnel Name: site
Local IP: 70.70.70.0/24   Peer IP: 7.7.7.10/32
Local SPI: 0x668f6ca4     Remote SPI: 0xcabf72e2
Protocol: esp    Mode: tunnel
Encryption Algorithm: aes-128  Hash Algorithm: sha256
DH Group: 2         NAT-T: disabled
Anti-Replay: disabled
Encrypted Packets: 0   Encrypted Bytes: 0
Decrypted Packets: 0   Decrypted Bytes: 0
Anti-Replay Failure: 0  Rekey Times: 7
Pad Check Error: 0 ICV Check Error: 0
Next Header Check Error: 0
Sequence Number: 0 Sequence Number Rollover: 0
Lifetime: 2917     Lifebytes: 9.76G
Pre-frag Success: 0 Pre-frag Error: 0
Frag-after-encap Fragment Generated: 0 Fragment Received: 0

AX3200-12-186#show vpn ipsec-sa site out-spi 0xcabf72e2

SA Index: 4         Tunnel Name: site
Local IP: 70.70.70.0/24    Peer IP: 7.7.7.10/32
Local SPI: 0x668f6ca4      Remote SPI: 0xcabf72e2
Protocol: esp              Mode: tunnel
Encryption Algorithm: aes-128    Hash Algorithm: sha256
DH Group: 2           NAT-T: disabled
```

```
Anti-Replay: disabled
Encrypted Packets: 0  Encrypted Bytes: 0
Decrypted Packets: 0  Decrypted Bytes: 0
Anti-Replay Failure: 0    Rekey Times: 8
Pad Check Error: 0 ICV Check Error: 0
Next Header Check Error: 0
Sequence Number: 0 Sequence Number Rollover: 0
Lifetime: 1975      Lifebytes: 9.76G
Pre-frag Success: 0 Pre-frag Error: 0
Frag-after-encap Fragment Generated: 0 Fragment Received: 0
```

show VPN IPsec-SA <NAME> remote-ts <IP/MASK>

This is a new/highlighted command.

```
show vpn ipsec-sa <NAME> remote-ts-v6 <A:B:C:D:E:F:G:H/nn>
```

Description This command shows IPsec SAs based on filter Name + Remote Traffic Selector.

Example

```
AX3200-12-186#show vpn ipsec-sa site remote-ts 7.7.7.11 /32

SA Index: 5     Tunnel Name: site
Local IP: 70.70.70.0/24   Peer IP: 7.7.7.11/32
Local SPI: 0x668f6caf     Remote SPI: 0xc880e19b
Protocol: esp             Mode: tunnel
Encryption Algorithm: aes-128  Hash Algorithm: sha256
DH Group: 2      NAT-T: disabled
Anti-Replay: disabled
Encrypted Packets: 0    Encrypted Bytes: 0
Decrypted Packets: 0    Decrypted Bytes: 0
Anti-Replay Failure: 0  Rekey Times: 8
Pad Check Error: 0 ICV Check Error: 0
Next Header Check Error: 0
Sequence Number: 0 Sequence Number Rollover: 0
Lifetime: 5372     Lifebytes: 9.76G
Pre-frag Success: 0 Pre-frag Error: 0
Frag-after-encap Fragment Generated: 0 Fragment Received: 0
```

show VPN IKE-SA-Brief <NAME>

This is a new/highlighted command.

Description This command summarizes the IKE SAs for a given IKE Gateway Config.

Example

```
AX3200-12-186#show vpn ike-sa-brief site

Name: site
Local IP: 8.8.8.186
Remote IP     Remote ID          Lifetime   Status
-------------------------------------------------------------------------------
-----------------------------------
8.8.8.124     tuy.124.client     137s       Established
8.8.8.128     tuy.client         132s       Established
```

show VPN IKE-Clients <NAME>

This is a new/highlighted command.

Description This command shows IKE Clients based on filter Name.

For Site-to-Site, it shows nothing.

Example

```
AX3200-12-186#show vpn ike-clients site

Name: site
Local IP: 8.8.8.186
Remote IP   Remote ID      User ID        Idle Time  Session Time Bytes

-------------------------------------------------------------------------------
-
8.8.8.124   tuy.124.client tuy.124.client 5543s      5543s        0

8.8.8.128   tuy.client     tuy.client     5548s      5548s        0
```

show VPN IPsec-SA-Clients

This is a new/highlighted command.

Description This command summarizes the IPsec SAs from an IPsec Client perspective.

Example

```
AX3200-12-186#show vpn ipsec-sa-clients
Remote TS: 7.7.7.11/32
Name Local TS In SPI             Out SPI      Lifetime  Lifebytes
-------------------------------------------------------------------------
site 70.70.70.0/24 0x668f6cb5    0xc1892059   5514s     9.76G

Remote TS: 7.7.7.10/32
Name Local TS In SPI             Out SPI      Lifetime   Lifebytes
-------------------------------------------------------------------------
site 70.70.70.0/24 0x668f6cb0    0xc4b078e9   4826s      9.76G

AX3200-12-186#
```

show vpn ike-stats

Description This command shows all IKE SA Stats.

For Client-to-Site IKE Gateways, this displays a summary of all the Client SAs associated with the IKE Gateway.

Example

```
AX3200-12-186#show vpn ike-stats

Gateway: site-dhcp                        Remote ID: tuy.client134-dhcp
Initiate Rekey: 0                         Respond Rekey: 0
Child SA Rekey: 0
Child SA Invalid SPI: 0
Outgoing Init Request: 0                  Incoming Init Request: 1
Outgoing Init Response: 1                 Incoming Init Response: 0
Outgoing Auth Request: 0                  Incoming Auth Request: 1
Outgoing Auth Response: 1                 Incoming Auth Response: 0
Outgoing Create Child Request: 0          Incoming Create Child Request: 0

Outgoing Create Child Response: 0         Incoming Create Child Response: 0

Outgoing Info Request: 0                  Incoming Info Request: 0
Outgoing Info Response: 0                 Incoming Info Response: 0
```

```
Incoming Invalid: 0                        Incoming Invalid SPI: 0

Gateway: site                             Remote ID: tuy.client124
Initiate Rekey: 0                         Respond Rekey: 0
Child SA Rekey: 1
Child SA Invalid SPI: 0
Outgoing Init Request: 0                  Incoming Init Request: 1
Outgoing Init Response: 1                 Incoming Init Response: 0
Outgoing Auth Request: 0                  Incoming Auth Request: 1
Outgoing Auth Response: 1                 Incoming Auth Response: 0
Outgoing Create Child Request: 0          Incoming Create Child Request: 1

Outgoing Create Child Response: 1         Incoming Create Child Response: 0

Outgoing Info Request: 0                  Incoming Info Request: 1
Outgoing Info Response: 1                 Incoming Info Response: 0
Incoming Invalid: 0                       Incoming Invalid SPI: 0

Gateway: site-dhcp                         Remote ID: tuy.client138-dhcp
Initiate Rekey: 0                          Respond Rekey: 0
Child SA Rekey: 0
Child SA Invalid SPI: 0
Outgoing Init Request: 0                   Incoming Init Request: 1
Outgoing Init Response: 1                  Incoming Init Response: 0
Outgoing Auth Request: 0                   Incoming Auth Request: 1
Outgoing Auth Response: 1                  Incoming Auth Response: 0
Outgoing Create Child Request: 0           Incoming Create Child Request: 0

Outgoing Create Child Response: 0          Incoming Create Child Response: 0

Outgoing Info Request: 0                   Incoming Info Request: 0
Outgoing Info Response: 0                  Incoming Info Response: 0
Incoming Invalid: 0                        Incoming Invalid SPI: 0

Gateway: site                             Remote ID: tuy.client128
Initiate Rekey: 0                         Respond Rekey: 0
Child SA Rekey: 0
Child SA Invalid SPI: 0
Outgoing Init Request: 0                  Incoming Init Request: 1
Outgoing Init Response: 1                 Incoming Init Response: 0
```

```
Outgoing Auth Request: 0                Incoming Auth Request: 1
Outgoing Auth Response: 1               Incoming Auth Response: 0
Outgoing Create Child Request: 0        Incoming Create Child Request: 0

Outgoing Create Child Response: 0       Incoming Create Child Response: 0

Outgoing Info Request: 0                Incoming Info Request: 0
Outgoing Info Response: 0               Incoming Info Response: 0
Incoming Invalid: 0                     Incoming Invalid SPI: 0
```

show vpn ike-stats <NAME>

Description This command shows IKE SA stats associated with a config (name). The first is a summary of all the IKE Stats associated with the config. This is followed by individual client IKE Stats if the IKE Gateway is in Client-to-Site Mode.

Example

```
AX3200-12-186#show vpn ike-stats site

Gateway: site
Remote IP: Remote ID:
Initiate Rekey: 6   Respond Rekey: 0
Child SA Rekey: 7
Child SA Invalid SPI: 0
Outgoing Init Request: 0            Incoming Init Request: 2
Outgoing Init Response: 2           Incoming Init Response: 0
Outgoing Auth Request: 0            Incoming Auth Request: 2
Outgoing Auth Response: 2           Incoming Auth Response: 0
Outgoing Create Child Request: 6    Incoming Create Child Request: 7

Outgoing Create Child Response: 7   Incoming Create Child Response: 6

Outgoing Info Request: 12           Incoming Info Request: 7
Outgoing Info Response: 7           Incoming Info Response: 12
Incoming Invalid: 0                 Incoming Invalid SPI: 0

Gateway: site
Remote IP: 8.8.8.214                 Remote ID: tuy.client214
Initiate Rekey: 3                    Respond Rekey: 0
Child SA Rekey: 4
```

```
Child SA Invalid SPI: 0
Outgoing Init Request: 0              Incoming Init Request: 1
Outgoing Init Response: 1             Incoming Init Response: 0
Outgoing Auth Request: 0              Incoming Auth Request: 1
Outgoing Auth Response: 1             Incoming Auth Response: 0
Outgoing Create Child Request: 3      Incoming Create Child Request: 4

Outgoing Create Child Response: 4     Incoming Create Child Response: 3

Outgoing Info Request: 6              Incoming Info Request: 4
Outgoing Info Response: 4             Incoming Info Response: 6
Incoming Invalid: 0                   Incoming Invalid SPI: 0

Gateway: site
Remote IP: 8.8.8.244                  Remote ID: tuy.client244
Initiate Rekey: 3                     Respond Rekey: 0
Child SA Rekey: 3
Child SA Invalid SPI: 0
Outgoing Init Request: 0              Incoming Init Request: 1
Outgoing Init Response: 1             Incoming Init Response: 0
Outgoing Auth Request: 0              Incoming Auth Request: 1
Outgoing Auth Response: 1             Incoming Auth Response: 0
Outgoing Create Child Request: 3      Incoming Create Child Request: 3

Outgoing Create Child Response: 3     Incoming Create Child Response: 3

Outgoing Info Request: 6              Incoming Info Request: 3
Outgoing Info Response: 3             Incoming Info Response: 6
Incoming Invalid: 0                   Incoming Invalid SPI: 0

AX3200-12-186#
```

show vpn ike-stats <NAME> remote-ip <IP>

This is a new/highlighted command.

Description This command shows IKE SA stats that match the filter IKE Gateway Name + Remote IP.

Example

```
AX3200-12-186#show vpn ike-stats site remote-ip 8.8.8.124
```

```
Gateway: site
Remote IP: 8.8.8.124                    Remote ID: tuy.client124
Initiate Rekey: 0                       Respond Rekey: 0
Child SA Rekey: 1
Child SA Invalid SPI: 0
Outgoing Init Request: 0                Incoming Init Request: 1
Outgoing Init Response: 1               Incoming Init Response: 0
Outgoing Auth Request: 0                Incoming Auth Request: 1
Outgoing Auth Response: 1               Incoming Auth Response: 0
Outgoing Create Child Request: 0        Incoming Create Child Request: 1

Outgoing Create Child Response: 1       Incoming Create Child Response: 0

Outgoing Info Request: 0                Incoming Info Request: 1
Outgoing Info Response: 1               Incoming Info Response: 0
Incoming Invalid: 0                     Incoming Invalid SPI: 0
```

show vpn ike-stats <NAME> remote-id <ID>

This is a new/highlighted command.

Description This command shows IKE SA stats that match the filter IKE Gateway Name + Remote ID.

Example

```
AX3200-12-186#show vpn ike-stats site-dhcp remote-id tuy.client138-dhcp

Gateway: site-dhcp
Remote IP: 8.8.8.138                     Remote ID: tuy.client138-dhcp
Initiate Rekey: 0                        Respond Rekey: 0
Child SA Rekey: 0
Child SA Invalid SPI: 0
Outgoing Init Request: 0                 Incoming Init Request: 1
Outgoing Init Response: 1                Incoming Init Response: 0
Outgoing Auth Request: 0                 Incoming Auth Request: 1
Outgoing Auth Response: 1                Incoming Auth Response: 0
Outgoing Create Child Request: 0         Incoming Create Child Request: 0

Outgoing Create Child Response: 0        Incoming Create Child Response: 0
```

```
Outgoing Info Request: 0                Incoming Info Request: 0
Outgoing Info Response: 0               Incoming Info Response: 0
Incoming Invalid: 0                     Incoming Invalid SPI: 0
```

aXAPI

In addition to the CLI changes this feature also has changes to the OPER/STATS.

The aXAPI was changed significantly to support multiple SAs in a single T2 OPER.

- The **ike-gateway** and **ipsec T2** OPERs were modified to support a multi for the OPER data.
- It allows for multiple clients to be shown for a single **vpn/ike/oper** and **vpn/ipsec/oper**.
- Additionally, multiple T1 counters are added to retrieve OPER information for **ike-sa-brief**, **ike-clients**, and **ipsec-sa-clients**.

The following are the examples of the aXAPI related to Client-to-Site.

```
GET /axapi/v3/vpn/ike-gateway/site/oper
{
  "ike-gateway": {
    "oper" : {
      "SA-List": [
        {
          "Initiator-SPI":"0x87204cf53083f73b",
          "Responder-SPI":"0xe41ebd394f7c1c8e",
          "Local-IP":"8.8.8.186",
          "Remote-IP":"8.8.8.128",
          "Encryption":"aes-128",
          "Hash":"sha256",
          "Lifetime":374,
          "Status":"Established",
          "NAT-Traversal":0,
          "Remote-ID":"tuy.client128"
        },
        {
          "Initiator-SPI":"0x6fd0f0a32b5fd86e",
```

```
          "Responder-SPI":"0x1c3677cfe2cad769",
          "Local-IP":"8.8.8.186",
          "Remote-IP":"8.8.8.124",
          "Encryption":"aes-128",
          "Hash":"sha256",
          "Lifetime":371,
          "Status":"Established",
          "NAT-Traversal":0,
          "Remote-ID":"tuy.client124"
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-gateway/site/oper",
    "name":"site"
  }
}
```



```
GET /axapi/v3/vpn/ike-gateway/site/oper?remote-ip-filter=8.8.8.124
{
  "ike-gateway": {
    "oper" : {
      "SA-List": [
        {
          "Initiator-SPI":"0x6fd0f0a32b5fd86e",
          "Responder-SPI":"0x1c3677cfe2cad769",
          "Local-IP":"8.8.8.186",
          "Remote-IP":"8.8.8.124",
          "Encryption":"aes-128",
          "Hash":"sha256",
          "Lifetime":309,
          "Status":"Established",
          "NAT-Traversal":0,
          "Remote-ID":"tuy.client124"
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-gateway/site/oper",
    "name":"site"
```

```
  }
}
```

```
GET /axapi/v3/vpn/ike-gateway/site/oper?remote-id-filter=tuy.client128
{
  "ike-gateway": {
    "oper" : {
      "SA-List": [
        {
          "Initiator-SPI":"0x87204cf53083f73b",
          "Responder-SPI":"0xe41ebd394f7c1c8e",
          "Local-IP":"8.8.8.186",
          "Remote-IP":"8.8.8.128",
          "Encryption":"aes-128",
          "Hash":"sha256",
          "Lifetime":281,
          "Status":"Established",
          "NAT-Traversal":0,
          "Remote-ID":"tuy.client128"
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-gateway/site/oper",
    "name":"site"
  }
}
```

```
GET /axapi/v3/vpn/ike-stats-by-gw/oper?gateway-name-filter=site
{
  "ike-stats-by-gw": {
    "oper" : {
      "ike-stats-list": [
        {
          "name":"site",
          "remote-id":"",
          "remote-ip":"",
```

```
          "ike-version":"v2",
          "v2-init-rekey":0,
          "v2-rsp-rekey":0,
          "v2-child-sa-rekey":0,
          "v2-in-invalid":0,
          "v2-in-invalid-spi":0,
          "v2-in-init-req":2,
          "v2-in-init-rsp":0,
          "v2-out-init-req":0,
          "v2-out-init-rsp":2,
          "v2-in-auth-req":2,
          "v2-in-auth-rsp":0,
          "v2-out-auth-req":0,
          "v2-out-auth-rsp":2,
          "v2-in-create-child-req":0,
          "v2-in-create-child-rsp":0,
          "v2-out-create-child-req":0,
          "v2-out-create-child-rsp":0,
          "v2-in-info-req":0,
          "v2-in-info-rsp":0,
          "v2-out-info-req":0,
          "v2-out-info-rsp":0,
          "v2-child-sa-invalid-spi":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-stats-by-gw/oper"
  }
}
```

```
GET /axapi/v3/vpn/ike-stats-by-gw/oper?gateway-name-filter=site&remote-ip-
filter=8.8.8.128
{
  "ike-stats-by-gw": {
    "oper" : {
      "ike-stats-list": [
        {
          "name":"site",
```

```
          "remote-id":"tuy.client128",
          "remote-ip":"8.8.8.128",
          "ike-version":"v2",
          "v2-init-rekey":0,
          "v2-rsp-rekey":0,
          "v2-child-sa-rekey":0,
          "v2-in-invalid":0,
          "v2-in-invalid-spi":0,
          "v2-in-init-req":1,
          "v2-in-init-rsp":0,
          "v2-out-init-req":0,
          "v2-out-init-rsp":1,
          "v2-in-auth-req":1,
          "v2-in-auth-rsp":0,
          "v2-out-auth-req":0,
          "v2-out-auth-rsp":1,
          "v2-in-create-child-req":0,
          "v2-in-create-child-rsp":0,
          "v2-out-create-child-req":0,
          "v2-out-create-child-rsp":0,
          "v2-in-info-req":0,
          "v2-in-info-rsp":0,
          "v2-out-info-req":0,
          "v2-out-info-rsp":0,
          "v2-child-sa-invalid-spi":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-stats-by-gw/oper"
  }
}
```

```
GET /axapi/v3/vpn/ike-stats-by-gw/oper?gateway-name-filter=site&remote-id-
filter=tuy.client124
{
  "ike-stats-by-gw": {
    "oper" : {
      "ike-stats-list": [
```

```
        {
          "name":"site",
          "remote-id":"tuy.client124",
          "remote-ip":"8.8.8.124",
          "ike-version":"v2",
          "v2-init-rekey":0,
          "v2-rsp-rekey":0,
          "v2-child-sa-rekey":0,
          "v2-in-invalid":0,
          "v2-in-invalid-spi":0,
          "v2-in-init-req":1,
          "v2-in-init-rsp":0,
          "v2-out-init-req":0,
          "v2-out-init-rsp":1,
          "v2-in-auth-req":1,
          "v2-in-auth-rsp":0,
          "v2-out-auth-req":0,
          "v2-out-auth-rsp":1,
          "v2-in-create-child-req":0,
          "v2-in-create-child-rsp":0,
          "v2-out-create-child-req":0,
          "v2-out-create-child-rsp":0,
          "v2-in-info-req":0,
          "v2-in-info-rsp":0,
          "v2-out-info-req":0,
          "v2-out-info-rsp":0,
          "v2-child-sa-invalid-spi":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-stats-by-gw/oper"
  }
}
```

```
GET /axapi/v3/vpn/ipsec/site/oper
{
  "ipsec": {
    "oper" : {
      "SA-List": [
```

```
        {
          "Status":"UP",
          "SA-Index":21,
          "Local-IP":"7.70.70.0/24",
          "Peer-IP":"9.9.9.128/32",
          "Local-SPI":"0x38dfd159",
          "Remote-SPI":"0xc70bc51e",
          "Protocol":"esp",
          "Mode":"tunnel",
          "Encryption-Algorithm":"aes-128",
          "Hash-Algorithm":"sha256",
          "Lifetime":5526,
          "Lifebytes":"9.76G",
          "DH-Group":2,
          "NAT-Traversal":0,
          "Anti-Replay":"disabled",
          "packets-encrypted":0,
          "packets-decrypted":0,
          "anti-replay-num":0,
          "rekey-num":2,
          "packets-err-inactive":0,
          "packets-err-encryption":0,
          "packets-err-pad-check":0,
          "packets-err-pkt-sanity":0,
          "packets-err-icv-check":0,
          "packets-err-lifetime-lifebytes":0,
          "bytes-encrypted":0,
          "bytes-decrypted":0,
          "prefrag-success":0,
          "prefrag-error":0,
          "cavium-bytes-encrypted":0,
          "cavium-bytes-decrypted":0,
          "cavium-packets-encrypted":0,
          "cavium-packets-decrypted":0,
          "tunnel-intf-down":0,
          "pkt-fail-prep-to-send":0,
          "no-next-hop":0,
          "invalid-tunnel-id":0,
          "no-tunnel-found":0,
          "pkt-fail-to-send":0,
```

```
        "frag-after-encap-frag-packets":0,
        "frag-received":0,
        "sequence-num":0,
        "sequence-num-rollover":0,
        "packets-err-nh-check":0
      },
      {
        "Status":"UP",
        "SA-Index":20,
        "Local-IP":"7.70.70.0/24",
        "Peer-IP":"9.9.9.124/32",
        "Local-SPI":"0x38dfd15b",
        "Remote-SPI":"0xc01dbef7",
        "Protocol":"esp",
        "Mode":"tunnel",
        "Encryption-Algorithm":"aes-128",
        "Hash-Algorithm":"sha256",
        "Lifetime":5923,
        "Lifebytes":"9.76G",
        "DH-Group":2,
        "NAT-Traversal":0,
        "Anti-Replay":"disabled",
        "packets-encrypted":0,
        "packets-decrypted":0,
        "anti-replay-num":0,
        "rekey-num":2,
        "packets-err-inactive":0,
        "packets-err-encryption":0,
        "packets-err-pad-check":0,
        "packets-err-pkt-sanity":0,
        "packets-err-icv-check":0,
        "packets-err-lifetime-lifebytes":0,
        "bytes-encrypted":0,
        "bytes-decrypted":0,
        "prefrag-success":0,
        "prefrag-error":0,
        "cavium-bytes-encrypted":0,
        "cavium-bytes-decrypted":0,
        "cavium-packets-encrypted":0,
        "cavium-packets-decrypted":0,
```

```
          "tunnel-intf-down":0,
          "pkt-fail-prep-to-send":0,
          "no-next-hop":0,
          "invalid-tunnel-id":0,
          "no-tunnel-found":0,
          "pkt-fail-to-send":0,
          "frag-after-encap-frag-packets":0,
          "frag-received":0,
          "sequence-num":0,
          "sequence-num-rollover":0,
          "packets-err-nh-check":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ipsec/site/oper",
    "name":"site"
  }
}
```

```
GET /axapi/v3/vpn/ipsec/site/oper?in-spi-filter=0x38dfd15b
{
  "ipsec": {
    "oper" : {
      "SA-List": [
        {
          "Status":"UP",
          "SA-Index":20,
          "Local-IP":"7.70.70.0/24",
          "Peer-IP":"9.9.9.124/32",
          "Local-SPI":"0x38dfd15b",
          "Remote-SPI":"0xc01dbef7",
          "Protocol":"esp",
          "Mode":"tunnel",
          "Encryption-Algorithm":"aes-128",
          "Hash-Algorithm":"sha256",
          "Lifetime":5890,
          "Lifebytes":"9.76G",
          "DH-Group":2,
```

```
          "NAT-Traversal":0,
          "Anti-Replay":"disabled",
          "packets-encrypted":0,
          "packets-decrypted":0,
          "anti-replay-num":0,
          "rekey-num":2,
          "packets-err-inactive":0,
          "packets-err-encryption":0,
          "packets-err-pad-check":0,
          "packets-err-pkt-sanity":0,
          "packets-err-icv-check":0,
          "packets-err-lifetime-lifebytes":0,
          "bytes-encrypted":0,
          "bytes-decrypted":0,
          "prefrag-success":0,
          "prefrag-error":0,
          "cavium-bytes-encrypted":0,
          "cavium-bytes-decrypted":0,
          "cavium-packets-encrypted":0,
          "cavium-packets-decrypted":0,
          "tunnel-intf-down":0,
          "pkt-fail-prep-to-send":0,
          "no-next-hop":0,
          "invalid-tunnel-id":0,
          "no-tunnel-found":0,
          "pkt-fail-to-send":0,
          "frag-after-encap-frag-packets":0,
          "frag-received":0,
          "sequence-num":0,
          "sequence-num-rollover":0,
          "packets-err-nh-check":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ipsec/site/oper",
    "name":"site"
  }
}
```

```
GET /axapi/v3/vpn/ipsec/site/oper?out-spi-filter=0xc70bc51e
{
  "ipsec": {
    "oper" : {
      "SA-List": [
        {
          "Status":"UP",
          "SA-Index":21,
          "Local-IP":"7.70.70.0/24",
          "Peer-IP":"9.9.9.128/32",
          "Local-SPI":"0x38dfd159",
          "Remote-SPI":"0xc70bc51e",
          "Protocol":"esp",
          "Mode":"tunnel",
          "Encryption-Algorithm":"aes-128",
          "Hash-Algorithm":"sha256",
          "Lifetime":5466,
          "Lifebytes":"9.76G",
          "DH-Group":2,
          "NAT-Traversal":0,
          "Anti-Replay":"disabled",
          "packets-encrypted":0,
          "packets-decrypted":0,
          "anti-replay-num":0,
          "rekey-num":2,
          "packets-err-inactive":0,
          "packets-err-encryption":0,
          "packets-err-pad-check":0,
          "packets-err-pkt-sanity":0,
          "packets-err-icv-check":0,
          "packets-err-lifetime-lifebytes":0,
          "bytes-encrypted":0,
          "bytes-decrypted":0,
          "prefrag-success":0,
          "prefrag-error":0,
          "cavium-bytes-encrypted":0,
          "cavium-bytes-decrypted":0,
          "cavium-packets-encrypted":0,
          "cavium-packets-decrypted":0,
          "tunnel-intf-down":0,
```

```
          "pkt-fail-prep-to-send":0,
          "no-next-hop":0,
          "invalid-tunnel-id":0,
          "no-tunnel-found":0,
          "pkt-fail-to-send":0,
          "frag-after-encap-frag-packets":0,
          "frag-received":0,
          "sequence-num":0,
          "sequence-num-rollover":0,
          "packets-err-nh-check":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ipsec/site/oper",
    "name":"site"
  }
}
```


```
GET /axapi/v3/vpn/ipsec/site/oper?remote-ts-filter=9.9.9.124/32
{
  "ipsec": {
    "oper" : {
      "SA-List": [
        {
          "Status":"UP",
          "SA-Index":20,
          "Local-IP":"7.70.70.0/24",
          "Peer-IP":"9.9.9.124/32",
          "Local-SPI":"0x38dfd15b",
          "Remote-SPI":"0xc01dbef7",
          "Protocol":"esp",
          "Mode":"tunnel",
          "Encryption-Algorithm":"aes-128",
          "Hash-Algorithm":"sha256",
          "Lifetime":5739,
          "Lifebytes":"9.76G",
          "DH-Group":2,
          "NAT-Traversal":0,
```

```
          "Anti-Replay":"disabled",
          "packets-encrypted":0,
          "packets-decrypted":0,
          "anti-replay-num":0,
          "rekey-num":2,
          "packets-err-inactive":0,
          "packets-err-encryption":0,
          "packets-err-pad-check":0,
          "packets-err-pkt-sanity":0,
          "packets-err-icv-check":0,
          "packets-err-lifetime-lifebytes":0,
          "bytes-encrypted":0,
          "bytes-decrypted":0,
          "prefrag-success":0,
          "prefrag-error":0,
          "cavium-bytes-encrypted":0,
          "cavium-bytes-decrypted":0,
          "cavium-packets-encrypted":0,
          "cavium-packets-decrypted":0,
          "tunnel-intf-down":0,
          "pkt-fail-prep-to-send":0,
          "no-next-hop":0,
          "invalid-tunnel-id":0,
          "no-tunnel-found":0,
          "pkt-fail-to-send":0,
          "frag-after-encap-frag-packets":0,
          "frag-received":0,
          "sequence-num":0,
          "sequence-num-rollover":0,
          "packets-err-nh-check":0
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ipsec/site/oper",
    "name":"site"
  }
}
```

```
GET /axapi/v3/vpn/ike-sa-brief/oper?name=site
{
  "ike-sa-brief": {
    "oper" : {
      "name":"site",
      "local-ip":"8.8.8.186",
      "ike-sa-brief-remote-gw": [
        {
          "ike-sa-brief-remote-gw-ip":"8.8.8.128",
          "ike-sa-brief-remote-gw-id":"tuy.client128",
          "ike-sa-brief-remote-gw-lifetime":"305s",
          "ike-sa-brief-remote-gw-status":"Established"
        },
        {
          "ike-sa-brief-remote-gw-ip":"8.8.8.124",
          "ike-sa-brief-remote-gw-id":"tuy.client124",
          "ike-sa-brief-remote-gw-lifetime":"302s",
          "ike-sa-brief-remote-gw-status":"Established"
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-sa-brief/oper"
  }
}
```

```
GET /axapi/v3/vpn/ike-sa-clients/oper?name=site
{
  "ike-sa-clients": {
    "oper" : {
      "name":"site",
      "ike-sa-clients-local-ip":"8.8.8.186",
      "ike-sa-clients-remote-gw": [
        {
          "ike-sa-clients-remote-gw-ip":"8.8.8.128",
          "ike-sa-clients-remote-gw-remote-id":"tuy.client128",
          "ike-sa-clients-remote-gw-user-id":"tuy.client128",
          "ike-sa-clients-remote-gw-idle-time":"744s",
          "ike-sa-clients-remote-gw-session-time":"744s",
```

```
          "ike-sa-clients-remote-gw-bytes":"0"
        },
        {
          "ike-sa-clients-remote-gw-ip":"8.8.8.124",
          "ike-sa-clients-remote-gw-remote-id":"tuy.client124",
          "ike-sa-clients-remote-gw-user-id":"tuy.client124",
          "ike-sa-clients-remote-gw-idle-time":"747s",
          "ike-sa-clients-remote-gw-session-time":"747s",
          "ike-sa-clients-remote-gw-bytes":"0"
        }
      ]
    },
    "a10-url":"/axapi/v3/vpn/ike-sa-clients/oper"
  }
}
```

```
GET /axapi/v3/vpn/ipsec-sa-clients/oper
{
  "ipsec-sa-clients": {
    "oper" : {
      "ipsec-clients": [
        {
          "ipsec-clients-ip":"9.9.9.128/32",
          "sa-list": [
            {
              "name":"site",
              "local-ts":"7.70.70.0/24",
              "in-spi":"0x38dfd159",
              "out-spi":"0xc70bc51e",
              "lifetime":"5226s",
              "lifebytes":"9.76G"
            }
          ]
        },
        {
          "ipsec-clients-ip":"9.9.9.124/32",
          "sa-list": [
            {
```

```
                "name":"site",
                "local-ts":"7.70.70.0/24",
                "in-spi":"0x38dfd15b",
                "out-spi":"0xc01dbef7",
                "lifetime":"5623s",
                "lifebytes":"9.76G"
              }
            ]
          }
        ]
      },
      "a10-url":"/axapi/v3/vpn/ipsec-sa-clients/oper"
  }
}
```


Limitations

The following is a list of limitations for this feature.

- The configuration payload with RADIUS is dependent on the AAM module.

  An AAM authentication server of type RADIUS needs to be created and referenced in the IPsec object.

This feature supports the following:

- The IKE version 2
- Only the tunnel mode
- The implementation for PSK, PKI, EAP-MSCHAPv2, EAP-TLS, and EAP-RADIUS authentication.

This feature does not support the following:

- The Traffic Selector does not support a range of IPs.
- A single SA pair cannot support multiple traffic selectors.
- The IPv6 in IPv4 Tunnel with Configuration Payload (IP assignment) not supported.
- The IPv6 Configuration payload with DHCP not supported.

- The IPv6 Configuration payload with RADIUS not supported.
- The AH Protocol not supported.
- Thus, Client-to-Site VPN may not work together with SLB or DCFW.

Dynamic Routing Protocols

ACOS supports application of IPsec VPN to Border Gateway Protocol (BGP), Bidirectional Forwarding Detection (BFD), Open Shortest Path First (OSPF), and Equal-cost Multi-path routing (ECMP) traffic.

The following topics are covered:

NOTE: Reverse route injection is not supported.

BGP Overview

The routers in a BGP autonomous system (AS) advertise their routes to other BGP speakers (either internally or externally) through updates exchanged during peering sessions. These updates, or BGP route redistributions, can be passed over IPsec VPN.

NOTE: ACOS supports BGP4+ for IPv4.

BFD Overview

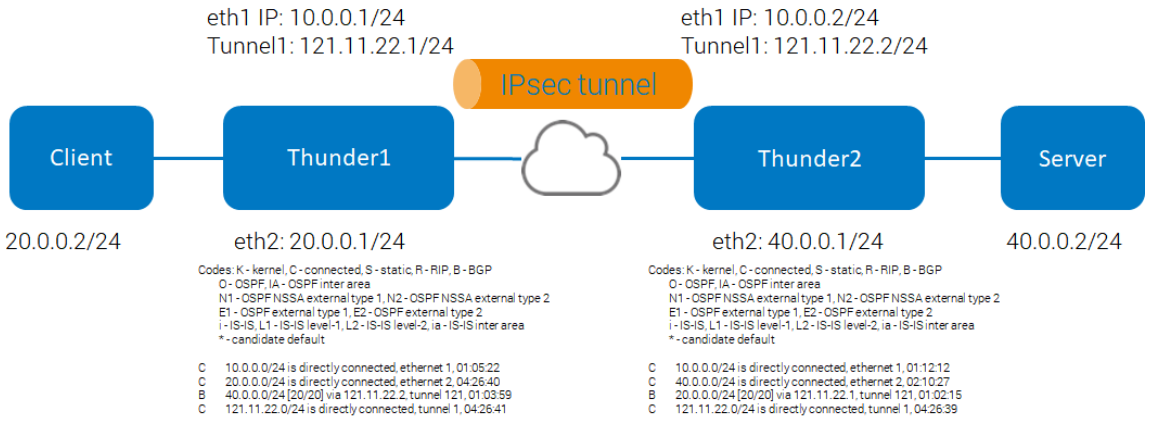
BFD provides very fast failure detection for routing protocols. BFD traffic can be passed over IPsec VPN.

NOTE: BFD provides a faster failure detection mechanism than the timeout values used by routing protocols. Routing protocol timers are multiple seconds long, whereas BFD provides sub-second failover.

Configuring BGP and BFD Traffic

The following commands configure IPsec VPN for BGP and BFD traffic:

Figure 22 : IPsec with BGP and BFD Sample Topology



ACOS-1

```
!
bfd enable
!
interface ethernet 1
  enable
  ip address 10.0.0.1 255.255.255.0
!
interface ethernet 2
  enable
  ip address 20.0.0.1 255.255.255.0
!
interface tunnel 1
  ip address 121.11.22.1 255.255.255.0
!
!
vpn ike-gateway v4
  auth-method preshare-key a10networks
  local-address ip 10.0.0.1
  remote-address ip 10.0.0.2
!
vpn ipsec v44
  ike-gateway v4
  bind tunnel 1 121.11.22.2
!
router bgp 50
  network 20.0.0.0/24
  neighbor 121.11.22.2 remote-as 121
!
```

ACOS-2

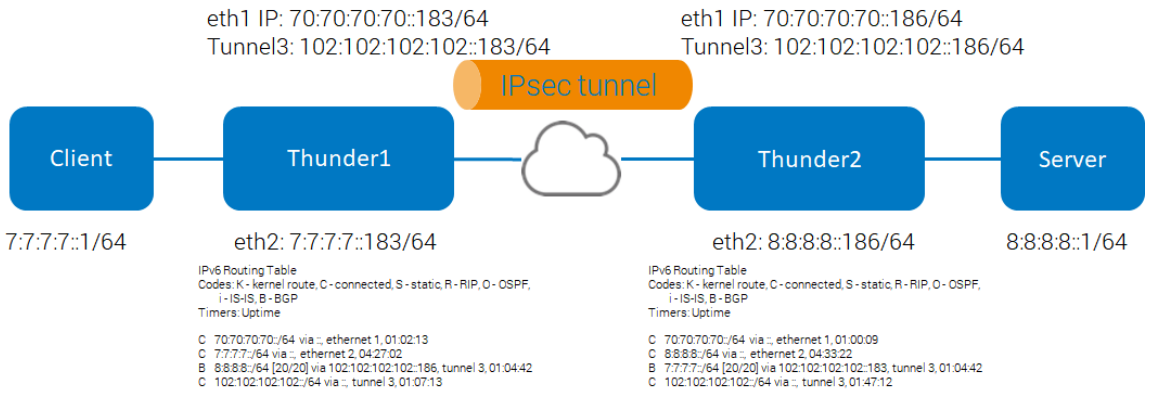
```
!
bfd enable
!
interface ethernet 1
```

```
  enable
  ip address 10.0.0.2 255.255.255.0
!
interface ethernet 2
  enable
  ip address 40.0.0.1 255.255.255.0
!
interface tunnel 1
  ip address 121.11.22.2 255.255.255.0
!
!
vpn ike-gateway v4
  auth-method preshare-key a10networks
  local-address ip 10.0.0.2
  remote-address ip 10.0.0.1
!
vpn ipsec v44
  ike-gateway v4
  bind tunnel 1 121.11.22.1
!
router bgp 121
  network 40.0.0.0/24
  neighbor 121.11.22.1 remote-as 50
!
```

Configuring IPsec IPv6 for BGP

The following commands configure IPsec IPv6 for BGP. There are two approaches to accomplish the deployment.

Figure 23 : IPsec IPv6 for BGP Sample Topology



The following topics are covered:

loopback interface

ACOS-1

```
interface ethernet 1
  enable
  ipv6 address 70:70:70:70::183/64
  ipv6 enable
!
interface ethernet 10
  enable
  ipv6 address 7:7:7:7::183/64
  ipv6 enable
!
interface loopback 1
  ipv6 address 4000::1/64
!
interface tunnel 3
  ipv6 address 102:102:102:102::183/64
  ipv6 enable
!
ipv6 route 4001::/64 tunnel 3 102:102:102:102::186
```

```
!
vpn ike-gateway 3200v6
  auth-method preshare-key 12345
  encryption aes-256 hash sha1
  local-address ipv6 70:70:70:70::183
  remote-address ipv6 70:70:70:70::186
!
vpn ipsec 3200v66
  ike-gateway 3200v6
  encryption aes-256 hash sha256
  traffic-selector ipv6 localv6 ::/64 remotev6 ::/64
  bind tunnel 3 102:102:102:102::186
!
maximum-paths 64
!
router bgp 183
  bgp router-id 192.168.230.183
  maximum-paths 64
  neighbor 4001::1 remote-as 186
  no neighbor 4001::1 activate
  neighbor 4001::1 ebgp-multihop 255
  neighbor 4001::1 update-source 4000::1
  address-family ipv6
    network 7:7:7:7::/64
    neighbor 4001::1 activate
!
```

ACOS-2

```
interface ethernet 1
  enable
  ipv6 address 70:70:70:70::186/64
  ipv6 enable
!
interface ethernet 10
  enable
  ipv6 address 8:8:8:8::186/64
  ipv6 enable
!
interface loopback 1
  ipv6 address 4001::1/64
```

```
!
interface tunnel 3
  ipv6 address 102:102:102:102::186/64
  ipv6 enable
!
ipv6 route 4000::/64 tunnel 3 102:102:102:102::183
!
vpn ike-gateway 3200v6
  auth-method 12345
  encryption aes-256 hash sha1
  local-address ipv6 70:70:70:70::186
  remote-address ipv6 70:70:70:70::183
!
vpn ipsec 3200v66
  ike-gateway 3200v6
  encryption aes-256 hash sha256
  traffic-selector ipv6 localv6 ::/64 remotev6 ::/64
 bind tunnel 3 102:102:102:102::183
!
maximum-paths 64
!
router bgp 186
  bgp router-id 192.168.230.186
  maximum-paths 64
  neighbor 4000::1 remote-as 183
  no neighbor 4000::1 activate
  neighbor 4000::1 ebgp-multihop 255
  neighbor 4000::1 update-source 4001::1
  address-family ipv6
    network 8:8:8:8::/64
    neighbor 4000::1 activate
!
```

route map

ACOS-1

```
vlan 7
  tagged ethernet 1
```

```
  router-interface ve 7
!
vlan 52
  tagged ethernet 2
  router-interface ve 52
!
ipv6 prefix-list VPN_6 seq 99 permit 58:58:58::/64
!
ipv6 prefix-list VPN_6 seq 100 permit 52:52:52::/64
!
interface ve 7
  ipv6 address 7:7:7::1/64
!
interface ve 52
  ipv6 address 52:52:52::1/64
!
interface tunnel 1
  ipv6 address 64:64:64::1/64
  ipv6 enable
!
vpn ike-gateway v6
  auth-method preshare-key A10Networks
  local-address ipv6 7:7:7::1
  remote-address ipv6 7:7:7::2
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 52:52:52::/64 remotev6 55:55:55::/64
  bind tunnel 1 64:64:64::2
!
router bgp 121
  bgp router-id 192.168.230.121
  maximum-paths 64
  neighbor 64:64:64::2 remote-as 122
  no neighbor 64:64:64::2 activate
  address-family ipv6
    network 52:52:52::/64
    neighbor 64:64:64::2 activate
    neighbor 64:64:64::2 route-map 64:64:64::-out out
!
```

```
route-map 64:64:64::-out permit 1
  match ipv6 address prefix-list VPN_6
  set ipv6 next-hop 64:64:64::1
!
```

ACOS-2

```
vlan 7
  tagged ethernet 1
  router-interface ve 7
!
vlan 58
  tagged ethernet 2
  router-interface ve 58
!
ipv6 prefix-list VPN_6 seq 99 permit 58:58:58::/64
!
ipv6 prefix-list VPN_6 seq 100 permit 52:52:52::/64
!
interface ve 7
  ipv6 address 7:7:7::2/64
!
interface ve 58
  ipv6 address 58:58:58::1/64
!
interface tunnel 1
  ipv6 address 64:64:64::2/64
  ipv6 enable
!
vpn ike-gateway v6
  auth-method preshare-key A10Networks
  local-address ipv6 7:7:7::2
  remote-address ipv6 7:7:7::1
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 55:55:55::/64 remotev6 52:52:52::/64
  bind tunnel 1 64:64:64::1
!
router bgp 122
  bgp router-id 192.168.230.122
```

```
  maximum-paths 64
  neighbor 64:64:64::1 remote-as 121
  no neighbor 64:64:64::1 activate
  address-family ipv6
    network 58:58:58::/64
    neighbor 64:64:64::1 activate
    neighbor 64:64:64::1 route-map 64:64:64::-out out
!
route-map 64:64:64::-out permit 100
  match ipv6 address prefix-list VPN_6
  set ipv6 next-hop 64:64:64::2
!
```

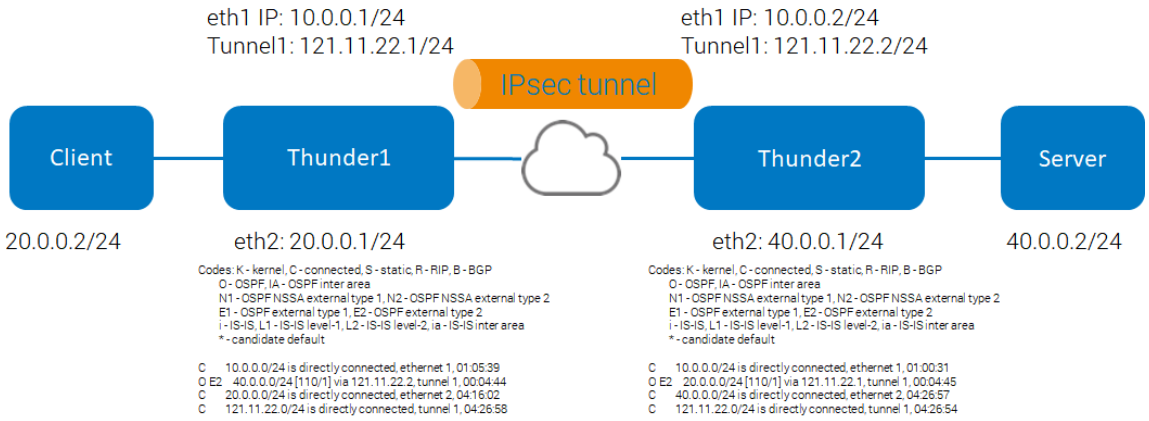
OSPF Overview

The ACOS device supports OSPF dynamic routing traffic over the IPsec VPN tunnel.

Configuring OSPF Traffic

The following commands configure IPsec VPN over OSPF traffic:

Figure 24 : IPsec VPN with OSPF Sample Topology



ACOS-1

```
!
```

```
interface ethernet 1
  enable
  ip address 10.0.0.1 255.255.255.0
!
interface ethernet 2
  enable
  ip address 20.0.0.1 255.255.255.0
!
interface tunnel 1
  ip address 121.11.22.1 255.255.255.0
!
!
vpn ike-gateway v4
  auth-method preshare-key a10networks
  local-address ip 10.0.0.1
  remote-address ip 10.0.0.2
!
vpn ipsec v44
  ike-gateway v4
  bind tunnel 1 121.11.22.2
!
router ospf
  router-id 1.1.1.1
  area 0.0.0.0 authentication
  neighbor 121.11.22.2
!
```

ACOS-2

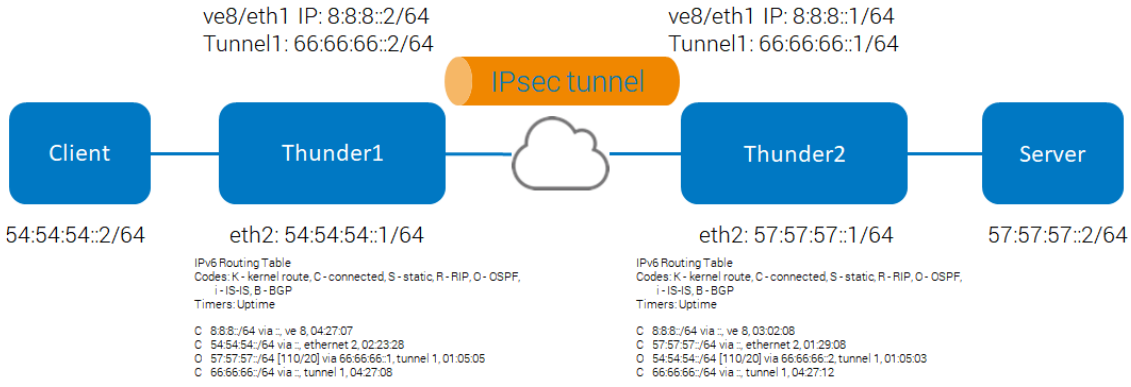
```
!
interface ethernet 1
  enable
  ip address 10.0.0.2 255.255.255.0
!
interface ethernet 2
  enable
  ip address 40.0.0.1 255.255.255.0
!
```

```
interface tunnel 1
  ip address 121.11.22.2 255.255.255.0
!
!
vpn ike-gateway v4
  auth-method preshare-key a10networks
  local-address ip 10.0.0.2
  remote-address ip 10.0.0.1
!
vpn ipsec v44
  ike-gateway v4
  bind tunnel 1 121.11.22.1
!
router ospf
  router-id 3.3.3.3
  area 0.0.0.0
  neighbor 121.11.22.1
```

Configuring IPsec IPv6 for OSPF

The following commands configure IPsec IPv6 for OSPF:

Figure 25 : IPsec IPv6 with OSPF Sample Topology



ACOS-1

```
vlan 8
  tagged ethernet 1
  router-interface ve 8
!
interface ve 8
  ipv6 address 8:8:8::2/64
!
interface tunnel 1
  ipv6 address 66:66:66::2/64
  ipv6 enable
  ipv6 router ospf area 0
!
vpn ike-gateway v6
  auth-method preshare-key A10Networks
  local-address ipv6 8:8:8::2
  remote-address ipv6 8:8:8::1
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 54:54:54::/64 remotev6 57:57:57::/64
  bind tunnel 1 66:66:66::1
!
router ipv6 ospf
  router-id 65.65.65.2
  redistribute connected route-map ospfv6
!
route-map ospfv6 permit 100
  set ipv6 next-hop 66:66:66::2
!
```

ACOS-2

```
vlan 8
  tagged ethernet 1
  router-interface ve 8
!
interface ve 8
```

```
  ipv6 address 8:8:8::1/64
!
interface tunnel 1
  ipv6 address 66:66:66::1/64
  ipv6 enable
  ipv6 router ospf area 0
!
vpn ike-gateway v6
  auth-method preshare-key encrypted
/+mboU9rpJM8EIy41dsA5zwQjLjV2wDnPBCMuNXbAOc8EIy41dsA5zwQjLjV2wDn
  local-address ipv6 8:8:8::1
  remote-address ipv6 8:8:8::2
!
vpn ipsec v66
  ike-gateway v6
  traffic-selector ipv6 localv6 57:57:57::/64 remotev6 54:54:54::/64
  bind tunnel 1 66:66:66::2
!
router ipv6 ospf
  router-id 65.65.65.1
  redistribute connected route-map ospfv6
!
route-map ospfv6 permit 100
  set ipv6 next-hop 66:66:66::1
!
```

ECMP Overview

ACOS supports ECMP routing to increase the total IPsec VPN bandwidth. ECMP, combined with BGP, allows routers to support multiple network routes simultaneously, allowing ACOS devices to load balance traffic across multiple paths to boost overall VPN capacity.

NOTE: For ECMP routing to work as designed, the underlying routing infrastructure must support ECMP hashing that includes the source port. Some routers may use just L3 data for hashing, or just the destination port.

The following topics are covered:

Multiple Tunnels for Internal Packets

For internal packets that need be sent out via an IPsec tunnel, ECMP can be used to choose among multiple IPsec tunnels based on the route. In this way, the ACOS device load balances traffic among multiple IPsec tunnels.

ACOS uses the same IPsec tunnel for all packets of one flow. This is applied when the internal flow is TCP, UDP, and ICMP. For TCP or UDP internal flow, the overlay tunnel information is based on source port and destination port. For ICMP internal flow, the overlay tunnel information is based on the ICMP ID.

NOTE: When IPsec VPN works together with other modules such as SLB and ACOS do not perform ECMP based on the overlay tunnel information.

Multiple Tunnels to VPN Peer

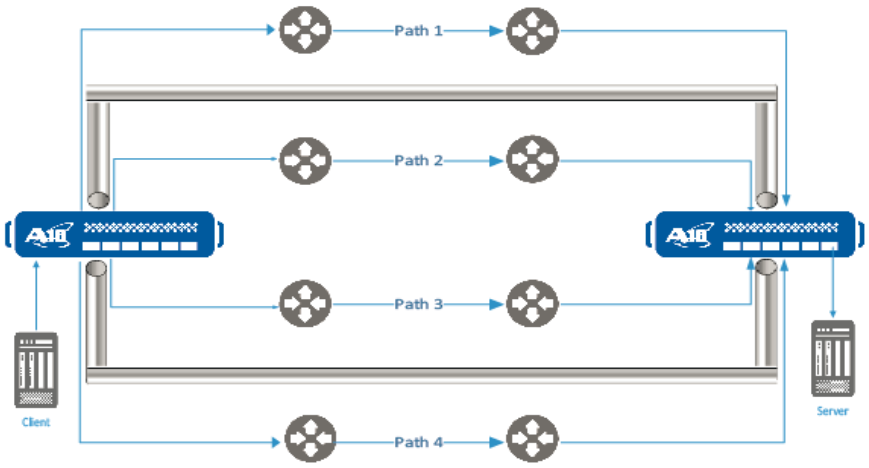
For multiple route paths to a VPN peer, ACOS uses ECMP to choose the route for the encapsulated VPN packet to load balance the VPN packet among these paths. This is done only for UDP encapsulation when the source port of the UDP encapsulated packet is chosen based on the source port and destination port of the internal flow.

Two VPN tunnels encapsulate traffic between the client and server. For a tunnel 1 packet, ACOS can use either path 1 or path 2. For a tunnel 2 packet, ACOS can use either path 3 or path 4. There are multiple VPN tunnels between the ACOS devices and multiple route paths for each tunnel. The same tunnel is used for all packets of one flow.

When IPsec VPN works together with other modules such as SLB and ACOS do not perform ECMP based on overlay tunnel information.

NOTE: ACOS supports load balancing traffic for IPsec VPN peer over ECMP routes sync on VRRP-A.

Figure 26 : IPsec with ECMP Sample Topology



NOTE: ACOS supports ECMP routes learnt using OSPF, ISIS, ESP, SCEP, and OCSP.

Configuring ECMP Traffic

The following topics are covered:

Configuring Tunnel Selection

The following topics are covered:

VPN Configuration

```
vpn ike-gateway ike1
        ike-version v2
        auth-method preshare-key a10
        encryption 3des hash md5
```

```
        dh-group 2
        local-address ip 10.11.11.100
        remote-address ip 20.11.11.218
!
vpn ike-gateway ike2
        auth-method preshare-key a10
        encryption aes-256 hash sha256
        dh-group 1
        local-address ip 10.12.11.100
        remote-address ip 20.12.11.218
!
vpn ipsec pkitest
        ike-gateway ike1
        bind tunnel 1 192.168.1.2
        encryption 3des hash sha256
        traffic-selector ipv4 local 11.0.0.1 255.255.255.255 remote 12.0.0.126
255.255.255.255
!
vpn ipsec pkitest1
        ike-gateway ike1
        bind tunnel 4 192.168.4.2
        encryption 3des hash sha256
        traffic-selector ipv4 local 11.0.0.2 255.255.255.255 remote 12.0.0.126
255.255.255.255
!
vpn ipsec pkitest2
        ike-gateway ike2
        bind tunnel 2 192.168.2.2
        encryption 3des hash sha256
        traffic-selector ipv4 local 11.0.0.1 255.255.255.255 remote 12.0.0.126
255.255.255.255
!
```

Route Configuration

```
ip route 12.0.0.126 /32 tunnel 1 192.168.1.2
ip route 12.0.0.126 /32 tunnel 4 192.168.4.2
ip route 12.0.0.126 /32 tunnel 2 192.168.2.2
ip route 20.11.11.218 /32 10.11.11.57
ip route 20.12.11.218 /32 10.12.11.57
```

Configuring Router Selection

The following commands configure IPsec VPN over ECMP traffic using Router selection.

The following topics are covered:

VPN Configuration

```
vpn nat-traversal-flow-affinity
vpn ike-gateway ike3
        auth-method preshare-key a10
        encryption aes-256 hash sha256
        dh-group 5
        local-address ip 10.11.12.100
        remote-address ip 20.11.12.218
        nat-traversal
vpn ipsec pkitest3
        ike-gateway ike3
        bind tunnel 3 192.168.3.2
        encryption aes-128 hash md5
        traffic-selector ipv4 local 11.1.1.1 255.255.255.255 remote 12.1.1.126
255.255.255.255
!
```

Route Configuration

```
ip route 12.1.1.126 /32 tunnel 3 192.168.3.2
ip route 20.11.12.218 /32 10.11.12.57
ip route 20.11.12.218 /32 10.12.12.57
```

Running RIPv2 and RIPng over IPsec SA

The following topics are covered:

Overview

This feature enables the RIPv2 routing protocol to endorse routing information through IPsec SA. The RIPv2 routing protocol enables the ability to exchange routing information through IPv4 IPsec SA, while RIPng is only capable of IPv6 IPsec SA.

Requirements

The RIPv2 exchange routing information between peers, so that the network topology behind each of these peers can be reachable by the other. The client must be able to send the traffic through the advertising routes over the IPsec SA to reach the server and the vice-versa.

Scenario

The sequence of scenarios for this feature is as the following:

- The RIPv2 packet is IPv4 multicast, UDP (17), port number 520 with **`dest IP 224.0.0.9`**.
- After the ACOS receives these RIPv2 packets, which are decapsulated from the ESP packets, it forwards to the kernel to continue this process request.

CLI Configuration

For this feature, the CLI config command changes are as the following:

There are additional interface config options for RIPv2 as the following, which the user can refer to.

```
TH4430S-121(config-if:tunnel:27)#ip rip ?
```

```
  authentication  Authentication control
  receive         Advertisement reception
  receive-packet  Enable receiving packet through the specified interface
  send            Advertisement transmission
  send-packet     Enable sending packets through the specified interface
  split-horizon   Perform split horizon
TH4430S-121(config-if:tunnel:27)#
```

Configuring the Initial Set-up

For this feature, the following are the multiple representations to configure the initial set-up of the system.

The following topics are covered:

AX1

```
vlan 3
  tagged ethernet 1
  router-interface ve 3
!
vlan 15
  tagged ethernet 2
  router-interface ve 15
!
interface ethernet 1
  enable
!
interface ethernet 2
  enable
!
interface ve 3
  ip address 3.3.3.1 255.255.255.0
!
interface ve 15
  ip address 15.15.15.1 255.255.255.0
!
```

```
interface tunnel 27
  ip address 27.27.27.1 255.255.255.0
!
vpn ike-gateway 3_v4
  auth-method preshare-key A10Networks
  encryption aes-128 hash sha1
  local-address ip 3.3.3.1
  remote-address ip 3.3.3.2
!
vpn ipsec 15to21_v4
  ike-gateway 3_v4
  encryption aes-128 hash sha1
  bind tunnel 27 27.27.27.2
!
router rip
  network 15.15.15.0/24
  network tunnel 27
!
```

AX2

```
vlan 3
  tagged ethernet 1
  router-interface ve 3
!
vlan 21
  tagged ethernet 2
  router-interface ve 21
!
interface ethernet 1
  enable
!
interface ethernet 2
  enable
!
interface ve 3
  ip address 3.3.3.2 255.255.255.0
!
interface ve 21
  ip address 21.21.21.1 255.255.255.0
!
```

```
interface tunnel 27
  ip address 27.27.27.2 255.255.255.0
!
vpn ike-gateway 3_v4
  auth-method preshare-key A10Networks
  encryption aes-128 hash sha1
  local-address ip 3.3.3.2
  remote-address ip 3.3.3.1
!
vpn ipsec 15to21_v4
  ike-gateway 3_v4
  encryption aes-128 hash sha1
  bind tunnel 27 27.27.27.1
!
router rip
  network 21.21.21.0/24
  network tunnel 27
!
```

Limitation

This feature does not support the RIPv1 routing protocol.

IPsec Configuration

This chapter describes the steps for the configuration of IPsec with SLB and CGNAT.

The following topics are covered:

The ACOS device supports IPsec VPN:

- before Server Load Balancing (SLB) (IPsec + SLB) and after SLB (SLB + IPsec).
- before Carrier-Grade NAT (CGN) (IPsec +CGN) and after CGN (CGN+IPsec).

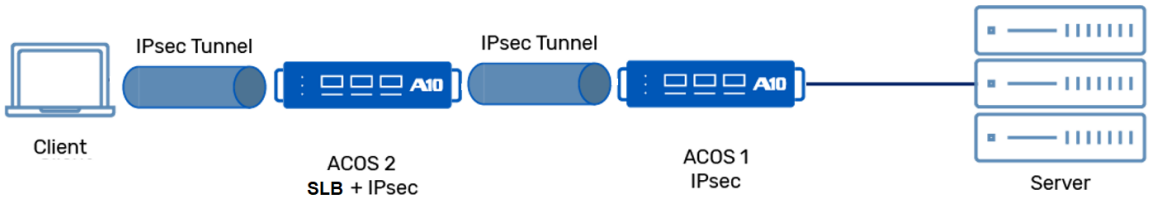
NOTE: IPsec with SLB and CGN can be performed only in the stateful mode.

Configuring IPsec with SLB

The traffic from a client passes through the IPsec VPN tunnel to enter the ACOS device that performs Server Load Balancing after VPN decapsulates the packet. ACOS creates a connection session and saves the VPN tunnel ID in its extension. The packet from the server matches SLB session in ACOS and is sent out through the VPN tunnel.

If the server chosen needs access through VPN, ACOS encapsulates the packet and sends it out through the VPN tunnel and saves the VPN tunnel ID in the SLB connection session for future use. The packets from the server pass the VPN tunnel first, then the ACOS device matches the session and performs the Server Load Balancing process after decapsulating the received packet.

Figure 27 : IPsec with SLB on Client Side Topology



The following topics are covered:

Configuring Encapsulation End

The following topics are covered:

VPN Configuration

The following is the VPN configuration for SLB:

```
vpn ike-gateway slb
  auth-method preshare-key a10
  encryption aes-256 hash sha256
  dh-group 5
  local-address ip 10.12.12.100
  remote-address ip 20.12.12.218
  nat-traversal

vpn ipsec slb
  ike-gateway slb
  bind tunnel 5 192.168.5.2
  encryption 3des hash sha256
  traffic-selector ipv4 local 11.2.2.1 255.255.255.255 remote 12.2.2.126
255.255.255.255
```

Route Configuration for SLB

The following is the Route configuration for SLB:

```
ip route 12.2.2.126 /32 tunnel 5 192.168.5.2
ip route 20.12.12.218 /32 10.11.12.57
ip route 20.12.12.218 /32 10.12.12.57
```

SLB Configuration

The following is the SLB configuration:

```
slb server s1 12.2.2.126
  port 22 tcp
    health-check-disable
!
slb service-group sg1 tcp
  health-check-disable
  member s1 22
!
slb virtual-server s1 11.2.2.101
  port 22 tcp
   service-group sg1
!
```

Configuring Decapsulation End

The following topics are covered:

VPN Configuration

The following is the VPN configuration:

```
vpn stateful-mode
vpn ike-gateway slb
  auth-method preshare-key a10
  encryption aes-256 hash sha256
  dh-group 5
```

```
  local-address ip 20.12.12.218
  remote-address ip 10.12.12.100
  nat-traversal

vpn ipsec slb
  ike-gateway slb
  bind tunnel 5 192.168.5.1
  encryption 3des hash sha256
```

Route Configuration

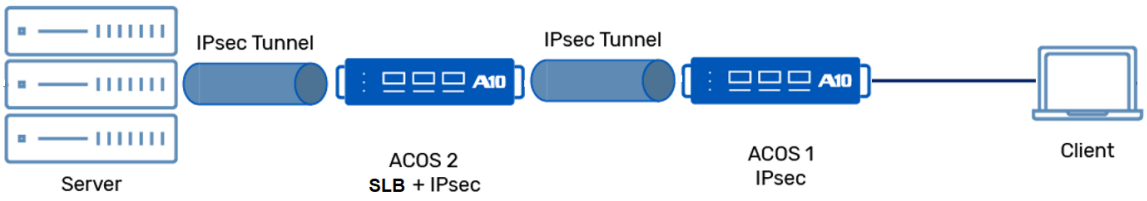
The following is the Route configuration:

```
ip route 10.12.12.100 /32 20.11.12.57
ip route 10.12.12.100 /32 20.12.12.57
ip route 11.2.2.1 /32 tunnel 5 192.168.5.1
```

Configuring SLB with IPsec

This section explains configuration of IPsec after SLB.

Figure 28 : IPsec with SLB on Server Side Topology



The following topics are covered:

Configuring Encapsulation End

The following topics are covered:

VPN configuration

The following is the VPN configuration:

```
vpn stateful-mode
vpn ike-gateway slb
  auth-method preshare-key a10
  encryption aes-256 hash sha256
  dh-group 5
  local-address ip 10.12.12.100
  remote-address ip 20.12.12.218
  nat-traversal

vpn ipsec slb-a
  ike-gateway slb
  bind tunnel 6 192.168.6.2
  encryption aes-256 hash sha256
  traffic-selector ipv4 local 11.0.0.0 255.255.255.0 remote 12.0.0.100
255.255.255.255
```

Route Configuration

The following is the Route configuration:

```
ip route 12.0.0.100 /32 tunnel 6 192.168.6.2
ip route 20.12.12.218 /32 10.11.12.57
ip route 20.12.12.218 /32 10.12.12.57
```

Configuring Decapsulation End

The following topics are covered:

VPN Configuration

The following is the VPN configuration:

```
vpn ike-gateway slb
  auth-method preshare-key a10
  encryption aes-256 hash sha256
  dh-group 5
  local-address ip 20.12.12.218
  remote-address ip 10.12.12.100
  nat-traversal

vpn ipsec slb-a
  ike-gateway slb
  bind tunnel 6 192.168.6.1
  encryption aes-256 hash sha256
  traffic-selector ipv4 local 12.0.0.100 255.255.255.255 remote 11.0.0.0
255.255.255.0
```

Route Configuration

The following is the Route configuration:

```
ip route 11.0.0.2 /32 tunnel 6 192.168.6.1
ip route 10.12.12.100 /32 20.11.12.57
ip route 10.12.12.100 /32 20.12.12.57
```

SLB Configuration

The following is the SLB configuration:

```
slb server slb-a 12.0.0.1
  port 22 tcp
   health-check-disable
!
slb service-group sg-a tcp
  health-check-disable
  member slb-a 22
!
slb virtual-server vs-a 12.0.0.100
  port 22 tcp
  source-nat auto
  service-group sg-a
```

Configuring IPsec with CGN

The CGNAT and IPsec functionality is often deployed in the same location where the data center is located (server-side), and sometimes deployed at the client-side. This functionality is needed to meet the traffic delivery requirements that combine the CGN for network address translation and IPsec for encrypting and protecting the traffic.

Based on the deployment scenario, it may be required to decapsulate the traffic first and then apply CGN or apply CGN first and then encapsulate the packets to send over the IPsec tunnel. ACOS provides the capabilities to implement both scenarios on one partition.

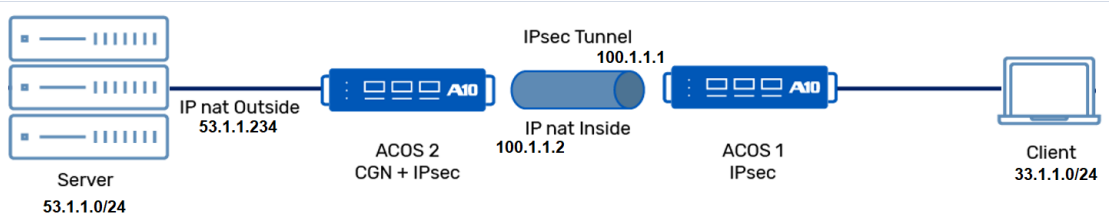
The following sections explain the configuration needed for the implementation of both use cases.

The following topics are covered:

CGN on Server Side of IPsec Topology

Figure 29 illustrates a topology when CGN is deployed at the server-side of IPsec.

Figure 29 : CGN at the Server-Side of IPsec



- The client sends a request to the server.
- On ACOS 1, IPsec encapsulates packets as the next hop is an IPsec tunnel.

- On ACOS 2, IPsec decapsulates packets before the CGN process.
- CGN will process the packets.
  For example, for NAT64, CGN creates a NAT session for the client, replaces the client's IPv6 address with an IPv4 address from the NAT pool. It also replaces the IPv6 destination address with the corresponding IPv4 address of the server.
- The server replies to the request and sends the packets.
- CGN processes packets received from the server and forwards them to the client. For NAT64, CGN translates the SYN-ACK into an IPv6 SYN-ACK and forwards it to the client.
- During the session, CGN checks for the router to send the traffic. Since the next hop is an IPsec tunnel, IPsec encapsulates the packets.
- ACOS 1 receives the encapsulated packets. IPsec decapsulates packets and forwards them to the client.

IPsec Configuration

The following configuration steps configure IPsec on ACOS 1.

1. Enter the following command to configure the ACOS device to set up sessions for the IPsec traffic:

```
ACOS(config)#vpn stateful-mode
```

2. Enter the following command to configure an IKE-based VPN gateway:

```
ACOS(config)#vpn ike-gateway t1
```

3. Enter the following command to configure the preshared key for authentication:

```
ACOS(config-ike-gateway:t1)#auth-method preshare-key 123456
```

4. Enter the following command to specify the ID value of the gateway for IKE:

```
ACOS(config-ike-gateway:t1)#local-id id_233
```

5. Enter the following command to specify the ID value of the peer gateway for IKE:

```
ACOS(config-ike-gateway:t1)#remote-id id_234
```

6. Enter the following command to configure an encryption group for IKE traffic:

```
ACOS(config-ike-gateway:t1)#encryption aes-256 hash sha256
```

7. Enter the following command to specify the interface on the ACOS device to use as the local endpoint of the tunnel:

```
ACOS(config-ike-gateway:t1)#local-address ip 34.1.1.3
```

8. Enter the following command to specify the interface on the peer VPN gateway to use as the remote endpoint of the tunnel:

```
ACOS(config-ike-gateway:t1)#remote-address ip 34.1.1.4
```

9. Enter the following command to enable NAT traversal to establish a VPN connection with an endpoint that is on the other side of a NAT device. This encapsulates ESP traffic inside UDP packets before sending them to the peer VPN gateway:

```
ACOS(config-ike-gateway:t1)#nat-traversal
```

10. Enter the following command to configure an IPsec tunnel:

```
ACOS(config)#vpn ipsec t1
```

11. Enter the following command to enable Perfect Forward Secrecy (PFS). The DH group controls the strength of the keying material exchanged during initiation of the IPsec SA. The devices at each end of the VPN tunnel use the keying materials to generate the shared secret key and their own private keys.

```
ACOS(config-ipsec:t1)#dh-group 1
```

12. Enter the following command to configure an encryption group for tunnel traffic:

```
ACOS(config-ipsec:t1)#encryption aes-256 hash sha256
```

13. Enter the following command to bind the VPN tunnel configuration to the tunnel interface. The tunnel interface number is the local tunnel interface number. The IP address is the peer tunnel interface IP:

```
ACOS(config-ipsec:t1)#bind tunnel 1 100.1.1.2
```

14. Enter the following command to specify the IPsec gateway configuration to use for the local endpoint of the VPN tunnel:

```
ACOS(config-ipsec:t1)#ike-gateway t1
```

15. Use the `show running-config` command to verify the configuration:

```
ACOS#show running-config
vpn stateful-mode
!
vpn ike-gateway t1
 auth-method preshare-key 123456
 local-id id_233
 remote-id id_234
 encryption aes-256 hash sha256
 local-address ip 34.1.1.3
 remote-address ip 34.1.1.4
 nat-traversal
!
vpn ipsec t1
 dh-group 1
 encryption aes-256 hash sha256
 bind tunnel 1 100.1.1.2
 ike-gateway t1
!
```

Similarly, you can configure the IPsec on ACOS 2 using the following commands:

```
ACOS(config)#vpn stateful-mode
ACOS(config)#vpn nat-traversal-flow-affinity

ACOS(config)#vpn ike-gateway t1
ACOS(config-ike-gateway:t1)#auth-method preshare-key 123456
ACOS(config-ike-gateway:t1)#local-id id_234
ACOS(config-ike-gateway:t1)#remote-id id_233
ACOS(config-ike-gateway:t1)#encryption aes-256 hash sha256
ACOS(config-ike-gateway:t1)#local-address ip 34.1.1.4
ACOS(config-ike-gateway:t1)#remote-address ip 34.1.1.3
ACOS(config-ike-gateway:t1)#nat-traversal
ACOS(config-ike-gateway:t1)#exit

ACOS(config)#vpn ipsec t1
ACOS(config-ipsec:t1)#dh-group 1
ACOS(config-ipsec:t1)#encryption aes-256 hash sha256
ACOS(config-ipsec:t1)#bind tunnel 1 100.1.1.1
ACOS(config-ipsec:t1)#ike-gateway t1
ACOS(config-ipsec:t1)#exit
```

CGN Configuration

The following configuration steps configure LSN on ACOS 2:

1. Enter the following commands to configure a class list, define an IP subnet address and network mask length, and the LID number.

```
ACOS(config)#class-list c2
ACOS(config-class list)#33.1.1.0/24 lsn-lid 1
```

2. Enter the following command to bind the class list to the LSN:

```
ACOS(config)#cgnv6 lsn inside source class-list c2
```

3. Enter the following command to configure LSN NAT Pool:

```
ACOS(config)#cgnv6 nat pool p1 2.2.2.1 2.2.2.255 netmask /24
```

4. Enter the following command to configure LSN Limit IDs (LIDs):

```
ACOS(config)#cgnv6 lsn-lid 1
```

5. Enter the following command to bind an LSN NAT pool to the LID:

```
ACOS(config-lsn-lid)#source-nat-pool p1
```

6. Use the `show running-config` command to verify the configuration:

```
ACOS#show running-config
!
class-list c2
 33.1.1.0/24 lsn-lid 1
!
cgnv6 lsn inside source class-list c2
!
cgnv6 nat pool p1 2.2.2.1 2.2.2.255 netmask /24
!
cgnv6 lsn-lid 1
 source-nat-pool p1
!
```

IP Configuration

The following configuration steps configure the ethernet interface, tunnel interface, and a static route on ACOS 1.

1. Enter the following commands to configure the ethernet interface:

```
ACOS(config)#interface ethernet 1
ACOS(config-if:ethernet:1)#enable
ACOS(config-if:ethernet:1)#ip address 34.1.1.3 255.255.255.0

ACOS(config)#interface ethernet 2
ACOS(config-if:ethernet:2)#enable
ACOS(config-if:ethernet:2)#ip address 33.1.1.233 255.255.255.0
```

2. Enter the following commands to configure a tunnel interface:

```
ACOS(config)#interface tunnel 1
ACOS(config-if:tunnel:1)#ip address 100.1.1.1 255.255.255.0
```

3. Enter the following commands to configure a static route that is used in the tunnel interface:

```
ACOS(config)#ip route 53.1.1.0 /24 tunnel 1 100.1.1.2
```

The following configuration steps configure the ethernet interface, tunnel interface, and a static route on ACOS 2.

1. Enter the following commands to configure the ethernet interface:

```
ACOS(config)#interface ethernet 1
ACOS(config-if:ethernet:1)#enable
ACOS(config-if:ethernet:1)#ip address 34.1.1.4 255.255.255.0

ACOS(config)#interface ethernet 2
ACOS(config-if:ethernet:2)#enable
ACOS(config-if:ethernet:2)#ip address 53.1.1.234 255.255.255.0
ACOS(config-if:ethernet:2)#ip nat outside
```

2. Enter the following commands to configure a tunnel interface:

```
ACOS(config)#interface tunnel 1
```

```
ACOS(config-if:tunnel:1)#ip address 100.1.1.2 255.255.255.0
ACOS(config-if:tunnel:1)#ip nat inside
```

3. Enter the following commands to configure a static route that is used in the tunnel interface:

```
ACOS(config)#ip route 33.1.1.0 /24 tunnel 1 100.1.1.1
```

4. Use the `show running-config` command to verify the configuration:

   On ACOS1

```
ACOS#show running-config
interface ethernet 1
 enable
 ip address 34.1.1.3 255.255.255.0
!
interface ethernet 2
 enable
 ip address 33.1.1.233 255.255.255.0
!
interface tunnel 1
 ip address 100.1.1.1 255.255.255.0
!
ip route 53.1.1.0 /24 tunnel 1 100.1.1.2
!
```

   On ACOS2

```
ACOS#show running-config
interface ethernet 1
 enable
 ip address 34.1.1.4 255.255.255.0
!
interface ethernet 2
 enable
 ip address 53.1.1.234 255.255.255.0
 ip nat outside
!
interface tunnel 1
 ip address 100.1.1.2 255.255.255.0
 ip nat inside
```

```
!
ip route 33.1.1.0 /24 tunnel 1 100.1.1.1
```

IPsec in Multi-PU Deployment

This section describes the features and configurations specific to IPsec implementation in multi-PU platforms.

For the common multi-PU implementation details, see *Application Delivery Controller Guide*.

The following topics are covered:

Supported Features and Limitation

Supported IPsec and PKI Features on Multi-PU

- Internet Key Exchange (IKE) Security Association (SA) and IPsec SA establishment and negotiation.
- IPsec traffic distribution and encryption or decryption on PU1 and PU2.
- IPsec tunnel and VPN synchronization between PU1 and PU2.
- IPsec tunnel establishment using authentication methods.

Limitation

Only the stateless mode is supported.

Key Considerations

- Traffic from an odd source IP is sent to PU1. Similarly, traffic from an even source IP is sent to PU2.
- The IKE traffic is always processed by PU1. After a connection is established, the IPSec SAs are synchronized to PU2. The data traffic will be processed by PU1 and PU2.
- Use odd sequence numbers on PU1 and even sequence numbers on PU2 to send outbound IPsec packets.

  The inbound IPsec packets are distributed to PU1 or PU2. The multi-PU will verify the anti-replay window.
- Forward and reverse packets can be processed by different PUs.
- The show counters receive the counters from PU2 while the show summary counters receive the counters from PU1 and PU2.
- All the IPsec show commands can be used to view the IPsec traffic statistics on PU1 and PU2.

CLI Configuration

This section describes the general CLI configuration guidelines for implementing IPsec in multi-PU platforms to ensure seamless traffic distribution across PU1 and PU2.

1. Configure vpn gateway and tunnel.
2. Bind the VPN tunnel configuration to the tunnel interface.
3. Verify the configuration and view the show commands for traffic distribution.

Basic Configuration

This section provides the configuration example for IPsec tunnel establishment in multi-PU platforms.

Tunnel Configuration on ACOS 1

```
ACOS(config)# vlan 30
ACOS(config-vlan:30)# tagged ethernet 3
ACOS(config-vlan:30)# router-interface ve 30
ACOS(config-vlan:30)# name client
ACOS(config-vlan:30)# traffic-distribution-mode sip
ACOS(config-vlan:30)# exit
ACOS(config)# interface ethernet 3
ACOS(config-if:ethernet:3)# speed-forced-40g
ACOS(config-if:ethernet:3)# enable
ACOS(config-if:ethernet:3)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:3)# ip nat inside
ACOS(config-if:ethernet:3)# exit
ACOS(config)# interface ethernet 5
ACOS(config-if:ethernet:5)# enable
ACOS(config-if:ethernet:5)# traffic-distribution-mode dip
ACOS(config-if:ethernet:5)# ip address 10.10.10.1 255.255.255.0
ACOS(config-if:ethernet:5)# exit
ACOS(config)# interface ve 30
ACOS(config-if:ve:30)# ip address 30.30.30.2 255.255.255.0
ACOS(config-if:ve:30)# ip allow-promiscuous-vip
ACOS(config-if:ve:30)# exit
ACOS(config)#interface tunnel 1
ACOS(config-if:1)# ip address 11.11.11.1 255.255.255.0
```

```
ACOS(config-if:1)# exit
ACOS(config)# vpn signature-authentication
ACOS(config)# exit
ACOS(config)# vpn ike-gateway psk
ACOS(config-ike-gateway:psk)# auth-method preshare-key a10
ACOS(config-ike-gateway:psk)# local-address ip 10.10.10.1
ACOS(config-ike-gateway:psk)# remote-address ip 10.10.10.2
ACOS(config-ike-gateway:psk)# exit
ACOS(config)# vpn ipsec psk
ACOS(config-ipsec:psk)# traffic-selector ipv4 local 192.168.10.0
255.255.255.0 remote 192.168.20.0 255.255.255.0
ACOS(config-ipsec:psk)# bind tunnel 1 11.11.11.2
ACOS(config-ipsec:psk)# ike-gateway psk
```

Tunnel Configuration on ACOS 2

```
ACOS(config)# vlan 40
ACOS(config-vlan:40)# tagged ethernet 4
ACOS(config-vlan:40)# router-interface ve 40
ACOS(config-vlan:40)# name server
ACOS(config-vlan:40)# traffic-distribution-mode dip
ACOS(config-vlan:40)# exit
ACOS(config)# interface ethernet 7
ACOS(config-if:ethernet:7)# traffic-distribution-mode sip
ACOS(config-if:ethernet:7)# ip address 10.10.10.2 255.255.0.0
ACOS(config-if:ethernet:7)# ip allow-promiscuous-vip
ACOS(config-if:ethernet:7)# exit
ACOS(config)# interface ve 40
ACOS(config-if:ve:40)# ip address 40.40.40.2 255.255.255.0
ACOS(config-if:ve:40)# ip allow-promiscuous-vip
ACOS(config-if:ve:40)# exit
ACOS(config)#interface tunnel 1
ACOS(config-if:1)# ip address 11.11.11.2 255.255.255.0
ACOS(config-if:1)# exit
ACOS(config)# vpn ike-gateway psk
ACOS(config-ike-gateway:psk)# auth-method preshare-key a10
ACOS(config-ike-gateway:psk)# local-address ip 10.10.10.2
ACOS(config-ike-gateway:psk)# remote-address ip 10.10.10.1
ACOS(config-ike-gateway:psk)# exit
ACOS(config)# vpn ipsec psk
```

```
ACOS(config-ipsec:psk)# traffic-selector ipv4 local 192.168.20.0
255.255.255.0 remote 192.168.10.0 255.255.255.0
ACOS(config-ipsec:psk)# bind tunnel 1 11.11.11.1
ACOS(config-ipsec:psk)# ike-gateway psk
```

IPsec VPN Management Traffic

The IPsec can protect management traffic when accessing the management from data ports.

Applying IPsec to Management Traffic on Data Ports

Simply create an IKE gateway and IPsec tunnel, and apply to the interface that accepts management traffic, such as an data Ethernet interface, or loop-back interface.

NOTE: For more information on setting up management via data ports, see *System Administration Guide*.

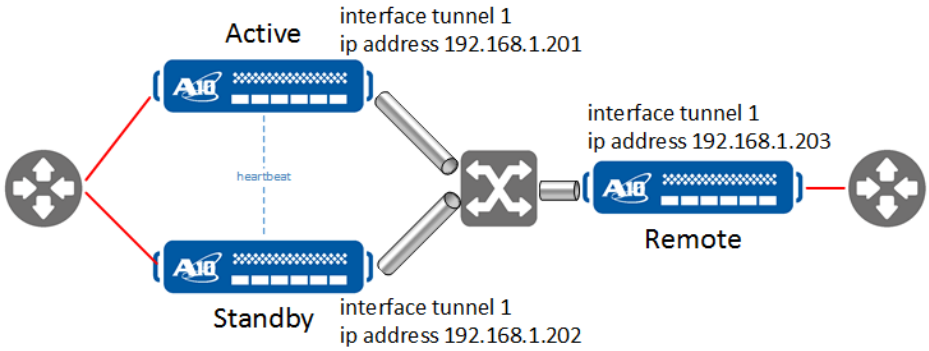
IPsec VPN and the VRRP-A Configuration

ACOS supports IPsec SA session synchronization for stateful failover of active VPN sessions. If a failover occurs, active IKE and IPsec SAs that have been synchronized continue uninterrupted.

NOTE: IPsec SA session synchronization requires certain pieces of configuration on the ACOS devices in the VRRP-A pair to be the same. The active and standby are not configured exactly the same, but the similarities are pointed out in the configuration steps.

The following commands deploy a single active/standby IPsec tunnel over ACOS devices in a site-to-site VPN. Figure 30 shows a single active/standby IPsec tunnel over ACOS devices in a site-to-site VPN configuration.

Figure 30 : VRRP-A Active-to-Standby Support



The following topics are covered:

Configuring IPsec and VRRP-A

Use the following commands to configure IPsec and VRRP-A on an Active, Standby, and Remote ACOS device.

The following topics are covered:

Active

If you change the configuration on the active device, then be sure to change it on the standby as well. There is only automatic session synchronization, but not automatic configuration synchronization.

The following topics are covered:

VRRP-A Configuration

The active VRRP-A configuration is shared for common `set-id`, `vrrp-a vrid`, and `floating-ip` addresses with the standby. Notice that the `priority` value is higher on the active device than on the standby device.

```
ACOS-Active(config)#vrrp-a common
ACOS-Active(config-common)#device-id 5
ACOS-Active(config-common)#set-id 15
ACOS-Active(config-common)#enable
ACOS-Active(config-common)#disable-default-vrid
ACOS-Active(config-common)#
ACOS-Active(config-common)#exit
```

```
ACOS-Active(config)#
ACOS-Active(config)#vlan 110
ACOS-Active(config-vlan:110)#tagged trunk 1
ACOS-Active(config-vlan:110)#router-interface ve 110
ACOS-Active(config-vlan:110)#exit
ACOS-Active(config)#vlan 2101
ACOS-Active(config-vlan:2101)#tagged trunk 1
ACOS-Active(config-vlan:2101)#router-interface ve 2101
ACOS-Active(config-vlan:2101)#exit
ACOS-Active(config)#hostname ACOS4430-1
ACOS-Active(config)#interface ethernet 1
ACOS-Active(config-ethernet:1)#enable
ACOS-Active(config-ethernet:1)#exit
ACOS-Active(config)#interface ethernet 2
ACOS-Active(config-ethernet:2)#enable
ACOS-Active(config-ethernet:2)#trunk-group 1
ACOS-Active(config-ethernet:2)#exit
ACOS-Active(config)#interface ethernet 3
ACOS-Active(config-ethernet:3)#enable
ACOS-Active(config-ethernet:3)#trunk-group 1
ACOS-Active(config-ethernet:3)#exit
ACOS-Active(config)#interface ethernet 4
ACOS-Active(config-ethernet:4)#exit
ACOS-Active(config)#interface ethernet 5
ACOS-Active(config-ethernet:5)#exit
ACOS-Active(config)#interface ethernet 6
ACOS-Active(config-ethernet:6)#exit
ACOS-Active(config)#interface ethernet 7
ACOS-Active(config-ethernet:7)#exit
ACOS-Active(config)#interface ve 110
ACOS-Active(config-if:ve:110)#ip address 50.0.1.1 255.255.255.0
ACOS-Active(config-if:ve:110)#exit
ACOS-Active(config)#interface ve 2101
ACOS-Active(config-if:ve:2101)#ip address 211.201.201.8 255.255.255.0
ACOS-Active(config-if:ve:2101)#exit
ACOS-Active(config)#exit
ACOS-Active(config)#vrrp-a vrid 31
ACOS-Active(config-vrid:31)#floating-ip 50.0.1.10
ACOS-Active(config-vrid:31)#floating-ip 211.201.201.100
ACOS-Active(config-vrid:31)#floating-ip 192.168.1.100
```

```
ACOS-Active(config-vrid:31)#blade-parameters
ACOS-Active(config-vrid:31-blade-parameters)#priority 254
ACOS-Active(config-vrid:31-blade-parameters)#exit
ACOS-Active(config-vrid:31)#vrrp-a interface ethernet 1
ACOS-Active(config-ethernet:1)#vlan 4000
ACOS-Active(config-ethernet:1)#exit
ACOS-Active(config)#ip route 50.0.0.0 /24 tunnel 1 192.168.1.203
ACOS-Active(config)#exit
```

Tunnel Interface Configuration

Configure a tunnel interface for later VPN IPsec binding. Notice that the `ip address` is different than on the standby device.

```
ACOS-Active(config)#interface tunnel 1
ACOS-Active(config-if:tunnel:1)#ip address 192.168.1.201 255.255.255.0
ACOS-Active(config-if:tunnel:1)#exit
```

Heartbeat Configuration

Configure a vlan for heartbeat checking of the connectivity between the active and standby. Notice that the `interface ve ip address` is different than on the standby device.

```
ACOS-Active(config)#vlan 4000
ACOS-Active(config-vlan:4000)#tagged ethernet 1
ACOS-Active(config-vlan:4000)#router-interface ve 4000
ACOS-Active(config-vlan:4000)#name vrrp-a-heartbeat
ACOS-Active(config-vlan:4000)#exit
ACOS-Active(config)#interface ve 4000
ACOS-Active(config-if:ve:4000)#ip address 4.0.0.1 255.255.255.252
ACOS-Active(config-if:ve:4000)#exit
```

VPN Configuration

Configure the vpn ike-gateway, and vpn ipsec encryption. The standby device shares these same settings.

```
ACOS-Active(config)#
ACOS-Active(config)#vpn ike-gateway 1
ACOS-Active(config-ike-gateway:1)#ike-version v1
ACOS-Active(config-ike-gateway:1)#vrid 31
```

```
ACOS-Active(config-ike-gateway:1)#auth-method preshare-key "password"
ACOS-Active(config-ike-gateway:1)#local-id ACOS4430-IKEv1-GW1
ACOS-Active(config-ike-gateway:1)#remote-id ACOS3030-IKEv1-GW1
ACOS-Active(config-ike-gateway:1)#encryption aes-256 hash sha256
ACOS-Active(config-ike-gateway:1)#dh-group 18
ACOS-Active(config-ike-gateway:1)#local-address ip 211.201.201.100
ACOS-Active(config-ike-gateway:1)#remote-address ip 211.201.201.5
ACOS-Active(config-ike-gateway:1)#lifetime 300
ACOS-Active(config-ike-gateway:1)#dpd interval 10 retry 3
ACOS-Active(config-ike-gateway:1)#exit
ACOS-Active(config)#vpn ipsec 1
ACOS-Active(config-ipsec:1)#ike-gateway 1
ACOS-Active(config-ipsec:1)#dh-group 18
ACOS-Active(config-ipsec:1)#encryption aes-256 hash sha256
ACOS-Active(config-ipsec:1)#bind tunnel 1 192.168.1.203
ACOS-Active(config-ipsec:1)#exit
```

NOTE: Use the `show hardware` command to verify the number of cores for SSL cards that are available to use with the `system ipsec crypto-core` command.

Verifying the Configuration

Use the `show running-config` command to verify the configuration.

```
ACOS-Active#show running-config
vrrp-a common
        device-id 5
        set-id 15
        enable
        disable-default-vrid
!
vlan 110
        tagged trunk 1
        router-interface ve 110
!
vlan 2101
        tagged trunk 1
        router-interface ve 2101
!
vlan 4000
```

```
        tagged ethernet 1
        router-interface ve 4000
        name vrrp-a-heartbeat
!
hostname ACOS4430-1
!
interface ethernet 1
        enable
!
interface ethernet 2
        enable
        trunk-group 1
!
interface ethernet 3
        enable
        trunk-group 1
!
interface ethernet 4
!
interface ethernet 5
!
interface ethernet 6
!
interface ethernet 7
!
interface ve 110
        ip address 50.0.1.1 255.255.255.0
!
interface ve 2101
        ip address 211.201.201.8 255.255.255.0
!
interface ve 4000
        ip address 4.0.0.1 255.255.255.252
!
interface tunnel 1
        ip address 192.168.1.201 255.255.255.0
!
!
vrrp-a vrid 31
        floating-ip 50.0.1.10
```

```
        floating-ip 211.201.201.100
        floating-ip 192.168.1.100
        blade-parameters
                priority 254
!
vrrp-a interface ethernet 1
        vlan 4000
!
ip route 50.0.0.0 /24 tunnel 1 192.168.1.203
!
!
vpn ike-gateway 1
        ike-version v1
        auth-method preshare-key "Password"
        vrid 31
        local-id ACOS4430-IKEv1-GW1
        remote-id ACOS3030-IKEv1-GW1
        encryption aes-256 hash sha256
        dh-group 18
        local-address ip 211.201.201.100
        remote-address ip 211.201.201.5
        lifetime 300
        dpd interval 10 retry 3
!
vpn ipsec 1
        ike-gateway 1
        dh-group 18
        encryption aes-256 hash sha256
        bind tunnel 1 192.168.1.203
!!
```

Standby

The standby device takes over the IPsec SA for active sessions when it no longer receives the heartbeat from the active device. It also starts accepting IKE and SA for new sessions.

To fail-over manually to the standby device, you can also change the VRRP-A priority on the active device to a smaller value than is currently configured on the standby device.

The following topics are covered:

VRRP-A Configuration

The standby VRRP-A configuration shares a common `set-id`, `vrrp-a vrid`, and `floating-ip` addresses with the active. Notice that the `priority` value is lower on the standby device than on the active device.

```
ACOS-Standby(config)#vrrp-a common
ACOS-Standby(config-common)#device-id 8
ACOS-Standby(config-common)#set-id 15
ACOS-Standby(config-common)#enable
ACOS-Standby(config-common)#disable-default-vrid
ACOS-Standby(config-common)#exit
ACOS-Standby(config)#
ACOS-Standby(config)#vlan 110
ACOS-Standby(config-vlan:110)#tagged trunk 1
ACOS-Standby(config-vlan:110)#router-interface ve 110
ACOS-Standby(config-vlan:110)#exit
ACOS-Standby(config)#vlan 2101
ACOS-Standby(config-vlan:2101)#tagged trunk 1
ACOS-Standby(config-vlan:2101)#router-interface ve 2101
ACOS-Standby(config-vlan:2101)#exit
ACOS-Standby(config)#hostname ACOS4430-1
ACOS-Standby(config)#interface ethernet 1
ACOS-Standby(config-ethernet:1)#enable
ACOS-Standby(config-ethernet:1)#exit
ACOS-Standby(config)#interface ethernet 2
ACOS-Standby(config-ethernet:2)#enable
ACOS-Standby(config-ethernet:2)#trunk-group 1
```

```
ACOS-Standby(config-ethernet:2)#exit
ACOS-Standby(config)#interface ethernet 3
ACOS-Standby(config-ethernet:3)#enable
ACOS-Standby(config-ethernet:3)#trunk-group 1
ACOS-Standby(config-ethernet:3)#exit
ACOS-Standby(config)#interface ethernet 4
ACOS-Standby(config-ethernet:4)#exit
ACOS-Standby(config)#interface ethernet 5
ACOS-Standby(config-ethernet:5)#exit
ACOS-Standby(config)#interface ethernet 6
ACOS-Standby(config-ethernet:6)#exit
ACOS-Standby(config)#interface ethernet 7
ACOS-Standby(config-ethernet:7)#exit
ACOS-Active(config)#interface ve 110
ACOS-Active(config-if:ve:110)#ip address 50.0.1.2 255.255.255.0
ACOS-Active(config-if:ve:110)#exit
ACOS-Active(config)#interface ve 2101
ACOS-Active(config-if:ve:2101)#ip address 211.201.201.4 255.255.255.0
ACOS-Active(config-if:ve:2101)#exit
ACOS-Active(config)#exit
ACOS-Standby(config)#vrrp-a vrid 31
ACOS-Standby(config-vrid:31)#floating-ip 50.0.1.10
ACOS-Standby(config-vrid:31)#floating-ip 211.201.201.100
ACOS-Standby(config-vrid:31)#floating-ip 192.168.1.100
ACOS-Standby(config-vrid:31)#blade-parameters
ACOS-Standby(config-vrid:31-blade-parameters)#priority 255
ACOS-Standby(config-vrid:31-blade-parameters)#exit
ACOS-Standby(config-vrid:31)#vrrp-a interface ethernet 1
ACOS-Standby(config-ethernet:1)#vlan 4000
ACOS-Standby(config-ethernet:1)#exit
ACOS-Standby(config)#ip route 50.0.0.0 /24 tunnel 1 192.168.1.203
ACOS-Standby(config)#exit
```

Tunnel Interface Configuration

Configure a tunnel interface for later VPN IPsec binding. Notice that the `ip address` is different than on the active device.

```
ACOS-Standby(config)#interface tunnel 1
ACOS-Standby(config-if:tunnel:1)#ip address 192.168.1.202 255.255.255.0
ACOS-Standby(config-if:tunnel:1)#exit
```

Heartbeat Configuration

Configure a vlan for heartbeat checking of the connectivity between the active and standby. Notice that the `interface ve ip address` is different than on the active device.

```
ACOS-Standby(config)#vlan 4000
ACOS-Standby(config-vlan:4000)#tagged ethernet 1
ACOS-Standby(config-vlan:4000)#router-interface ve 4000
ACOS-Standby(config-vlan:4000)#name vrrp-a-heartbeat
ACOS-Standby(config-vlan:4000)#exit
ACOS-Standby(config)#interface ve 4000
ACOS-Standby(config-if:ve:4000)#ip address 4.0.0.2 255.255.255.252
ACOS-Standby(config-if:ve:4000)#exit
```

VPN Configuration

Configure the vpn ike-gateway, and vpn ipsec encryption. The active device shares these same settings.

```
ACOS-Standby(config)#vpn ike-gateway 1
ACOS-Standby(config)#ike version v1
ACOS-Standby(config-ike-gateway:1)#vrid 31
ACOS-Standby(config-ike-gateway:1)#auth-method preshare-key "password"
ACOS-Standby(config-ike-gateway:1)#local-id ACOS4430-IKEv1-GW1
ACOS-Standby(config-ike-gateway:1)#remote-id ACOS3030-IKEv1-GW1
ACOS-Standby(config-ike-gateway:1)#encryption aes-256 hash sha256
ACOS-Standby(config-ike-gateway:1)#dh-group 18
ACOS-Standby(config-ike-gateway:1)#local-address ip 211.201.201.100
ACOS-Standby(config-ike-gateway:1)#remote-address ip 211.201.201.5
ACOS-Standby(config-ike-gateway:1)#lifetime 300
ACOS-Standby(config-ike-gateway:1)#dpd interval 10 retry 3
ACOS-Standby(config-ike-gateway:1)#exit
ACOS-Standby(config)#vpn ipsec 1
ACOS-Standby(config-ipsec:1)#ike-gateway 1
ACOS-Standby(config-ipsec:1)#dh-group 18
ACOS-Standby(config-ipsec:1)#encryption aes-256 hash sha256
ACOS-Standby(config-ipsec:1)#bind tunnel 1 192.168.1.203
ACOS-Standby(config-ipsec:1)#exit
```

NOTE: Use the `show hardware` command to verify the number of cores for SSL cards that are available to use with the `system ipsec crypto-core` command.

Verifying the Configuration

Use the show running-config command to verify the configuration.

```
Standby#show running-config
vrrp-a common
        device-id 8
        set-id 15
        enable
        disable-default-vrid
!
vlan 110
        tagged trunk 1
        router-interface ve 110
!
vlan 2101
        tagged trunk 1
        router-interface ve 2101
!
vlan 4000
        tagged ethernet 1
        router-interface ve 4000
        name vrrp-a-heartbeat
!
hostname ACOS4430-2
!
interface ethernet 1
        enable
!
interface ethernet 2
        enable
        trunk-group 1
!
interface ethernet 3
        enable
        trunk-group 1
```

```
!
interface ethernet 4
!
interface ethernet 5
!
interface ethernet 6
!
interface ethernet 7
!
interface ve 110
        ip address 50.0.1.2 255.255.255.0
!
interface ve 2101
        ip address 211.201.201.4 255.255.255.0
!
interface ve 4000
        ip address 4.0.0.2 255.255.255.252
!
interface tunnel 1
        ip address 192.168.1.202 255.255.255.0
!
!
vrrp-a vrid 31
        floating-ip 50.0.1.10
        floating-ip 211.201.201.100
        floating-ip 192.168.1.100
        blade-parameters
                priority 255
!
vrrp-a interface ethernet 1
        vlan 4000
!
ip route 50.0.0.0 /24 tunnel 1 192.168.1.203
!
!
vpn ike-gateway 1
        ike-version v1
        auth-method preshare-key "Password"
        vrid 31
        local-id ACOS4430-IKEv1-GW1
```

```
        remote-id ACOS3030-IKEv1-GW1
        encryption aes-256 hash sha256
        dh-group 18
        local-address ip 211.201.201.100
        remote-address ip 211.201.201.5
        lifetime 300
        dpd interval 10 retry 3
!
vpn ipsec 1
        ike-gateway 1
        dh-group 18
        encryption aes-256 hash sha256
        bind tunnel 1 192.168.1.203
!
end
```

Remote

The remote device terminates the tunnel.

The following topics are covered:

Tunnel Interface Configuration

Configure a tunnel interface for later VPN IPsec binding.

```
ACOS-Remote(config)#interface tunnel 1
ACOS-Remote(config-if:tunnel:1)#ip address 192.168.1.203 255.255.255.0
ACOS-Remote(config-if:tunnel:1)#exit
```

VPN Configuration

Configure the vpn ike-gateway, and vpn ipsec encryption.

```
ACOS-Remote(config)#vpn ike-gateway 1
ACOS-Remote(config-ike-gateway:1)#ike-version v1
ACOS-Remote(config-ike-gateway:1)#auth-method preshare-key "password"
```

```
ACOS-Remote(config-ike-gateway:1)#local-id ACOS3030-IKEv1-GW1
ACOS-Remote(config-ike-gateway:1)#remote-id ACOS4430-IKEv1-GW1
ACOS-Remote(config-ike-gateway:1)#encryption aes-256 hash sha256
ACOS-Remote(config-ike-gateway:1)#dh-group 18
ACOS-Remote(config-ike-gateway:1)#local-address ip 211.201.201.5
ACOS-Remote(config-ike-gateway:1)#remote-address ip 211.201.201.100
ACOS-Remote(config-ike-gateway:1)#lifetime 300
ACOS-Remote(config-ike-gateway:1)#dpd interval 10 retry 3
ACOS-Remote(config-ike-gateway:1)#exit
ACOS-Remote(config)#vpn ipsec 1
ACOS-Remote(config-ipsec:1)#ike-gateway 1
ACOS-Remote(config-ipsec:1)#dh-group 18
ACOS-Remote(config-ipsec:1)#encryption aes-256 hash sha256
ACOS-Remote(config-ipsec:1)#bind tunnel 1 192.168.1.100
ACOS-Remote(config-ipsec:1)#exit
```

Verifying the Configuration

Use the show running-config command to verify the configuration.

```
ACOS-Remote#show running-config
vlan 110
        untagged ethernet 2
        router-interface ve 110
!
vlan 2101
        tagged ethernet 1
        router-interface ve 2101
!
hostname ACOS3030
!
interface management
        ip address 10.255.255.86 255.255.0.0
!
interface ethernet 1
        enable
!
interface ethernet 2
        enable
!
interface ve 110
```

```
        ip address 50.0.0.1 255.255.255.0
!
interface ve 2101
        ip address 211.201.201.5 255.255.255.0
!
interface tunnel 1
        ip address 192.168.1.203 255.255.255.0
!
!
ip route 50.0.1.0 /24 tunnel 1 192.168.1.100
!
!
vpn ike-gateway 1
        ike-version v1
        auth-method preshare-key "Password"
        local-id ACOS3030-IKEv1-GW1
        remote-id ACOS4430-IKEv1-GW1
        encryption aes-256 hash sha256
        dh-group 18
        local-address ip 211.201.201.5
        remote-address ip 211.201.201.100
        lifetime 300
        dpd interval 10 retry 3
!
vpn ipsec 1
        ike-gateway 1
        dh-group 18
        encryption aes-256 hash sha256
        bind tunnel 1 192.168.1.100
!
end
```

NOTE: Use the `show hardware` command to verify the number of cores for SSL cards that are available to use with the `system ipsec crypto-core` command.

IKE Cipher Options

The following IKE (Internet Key Exchange) and IP security cipher options are provided during VPN setup:

- ECDSA authentication
- AES GCM encryption
- SHA-384 hashing

Use the vpn ike-gateway and vpn ipsec commands to configure the IKE and IP Security options.

For example:

```
#vpn ike-gateway
ACOS(config-ike-gateway:ike)#auth-method ecdsa-signature
ACOS(config-ike-gateway:ike)#dh-group 19
```

Use the `import cert` and `pki create cert` commands to create or import ECDSA certificates and keys.

For example:

```
ACOS(config)#pki create csr mycsr certtype ecdsa
```

IPsec Scaleout

In the topology of IPsec scaleout, upstream and downstream router uses ECMP to balance the traffic. By configuring the IKE gateway local address to the same loopback IP, all devices have the same Ike configuration. By synchronizing IKE/IPsec SA in scaleout, all devices have all the IKE/IPsec SA, so in stateless mode, all data traffic arriving at any device can be processed directly without redirected.

IPsec Scaleout is used for client-to-site scenarios. The same IKE gateway is configured on each device. The ECMP of inbound router was configured as per-flow, and the inbound traffic of one SA will be distributed to one device. This device will establish SA with client. The SA information is then synchronized to other devices, followed by outbound traffic distributed to devices.

IPsec uses sequence numbers that are sent as a part of data packets to provide anti-replay services. The inbound traffic (client-to-server) always reaches the same node in the Scaleout topology as the distribution from the ECMP router is based on the tunnel source and tunnel destination IP addresses. The sequence number handles the flow of the traffic.

The outbound traffic (server-to-client) may reach any nodes in the Scaleout topology and the sequence numbers must be distributed properly for a given tunnel to manage the flow of traffic. The sequence numbers are assigned to all the nodes in the Scaleout cluster. Two nodes maintain the per tunnel sequence number information across the nodes. The node with which the initial IKE negotiation is made becomes the master node for a given tunnel, and it maintains and provides a range of sequence numbers to the nodes to send traffic. All the other nodes store the backup information of the sequence number. When the master node is removed, a new node is selected as the new master and helps the traffic of the IPsec SA to flow seamlessly.

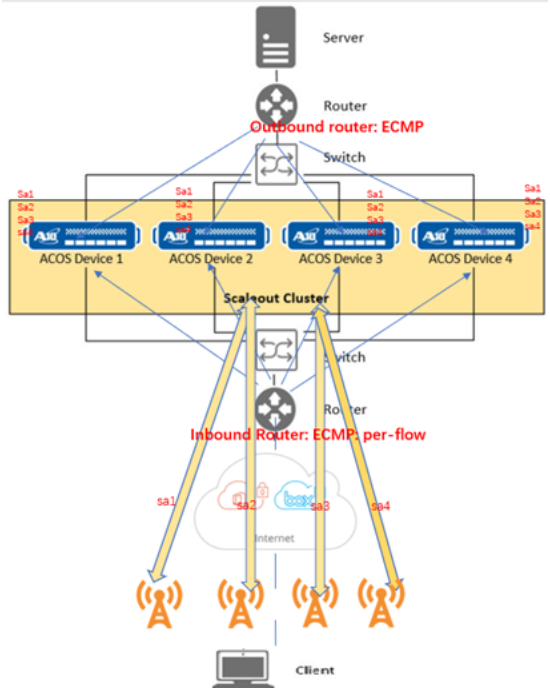
When a new node is added to the Scaleout cluster, the node receives the SA synchronize message, and gets a new sequence number range. When a node is removed from the Scaleout cluster the sequence numbers are updated too.

The outbound traffic may reach any one of the nodes where it is decrypted and forwarded to the client with the sequence number. The Anti-replay window that

protects against replay attacks is enhanced to support window sizes of 2048, 3072, 4096, and 8192. It is recommended to set a larger window size for the high traffic.

For more information about the Anti-reply window, see Anti-Replay Window.

Table 8 : IPsec Scaleout



Configuring VCS

To configure VCS , use the following commands:

ACOS1

```
ACOS(config)# vrrp-a common
ACOS1(config)# set-id 5
ACOS1(config-common)# vcs enable
ACOS1(config-common)# vcs device 1
ACOS1(config-common)# interfaces management
ACOS1(config-common)# enable
```

ACOS2

```
ACOS(config)# vrrp-a common
ACOS1(config)# set-id 5
ACOS1(config-common)# vcs enable
ACOS1(config-common)# vcs device 2
ACOS1(config-common)# interfaces management
ACOS1(config-common)# enable
```

ACOS3

```
ACOS(config)# vrrp-a common
ACOS1(config)# set-id 5
ACOS1(config-common)# vcs enable
ACOS1(config-common)# vcs device 3
ACOS1(config-common)# interfaces management
ACOS1(config-common)# enable
```

Configuring vMaster

Configure the Scaleout cluster. In a VCS environment, you can use the command to use the device ID and priority settings from your VCS configuration.

```
ACOS1-vMaster[8/1](config:3)# scaleout 1
ACOS1-vMaster[8/1](config:3)# device-context 1
ACOS1-vMaster[8/1](config:3)# local-device
ACOS1-vMaster[8/1](config:3)# priority 120
```

```
ACOS1-vMaster[8/1](config:3)# id 1
ACOS1-vMaster[8/1](config:3)# device-context 2
ACOS1-vMaster[8/1](config:3)# local-device
ACOS1-vMaster[8/1](config:3)# priority 100
ACOS1-vMaster[8/1](config:3)# id 2
ACOS1-vMaster[8/1](config:3)# device-context 3
ACOS1-vMaster[8/1](config:3)# local-device
ACOS1-vMaster[8/1](config:3)# priority 100
ACOS1-vMaster[8/1](config:3)# id 3
ACOS1-vMaster[8/1](config:3)# cluster-devices
ACOS1-vMaster[8/1](config:3)# device-id 1
ACOS1-vMaster[8/1](config:3)# ip 33.1.1.1
ACOS1-vMaster[8/1](config:3)# device-id 2
ACOS1-vMaster[8/1](config:3)# ip 33.1.1.2
ACOS1-vMaster[8/1](config:3)# device-id 3
ACOS1-vMaster[8/1](config:3)# ip 33.1.1.3
ACOS1-vMaster[8/1](config:3)# device-groups
ACOS1-vMaster[8/1](config:3)# device-group 1
ACOS1-vMaster[8/1](config:3)# device-id 1 to 3
ACOS1-vMaster[8/1](config:3)# scaleout apps enable
```

Scaleout is now active:

```
TH1040S-30-21-vMaster[5/1](config:1)#show scaleout
Device Role   : Cluster Master
Cluster Mode : Layer-2
Device 1 - Active (Local) (Master)
Device 2 - Active
Device 3 - Active
```

Configuring IKE and Loopback

Configure the Scaleout cluster. In a VCS environment, you can use the command to use the device ID and priority settings from your VCS configuration.

```
ACOS(config)# vpn ike-gateway t1
ACOS(config)# auth-method preshare-key 123456
ACOS(config)# encryption aes-256 hash sha256
ACOS(config)# local-address ip 2.2.2.2
ACOS(config)# vpn ipsec t1
```

```
ACOS(config)# encryption aes-256 hash sha256
ACOS(config)# sequence-number-disable
ACOS(config)# bind tunnel 1
ACOS(config)# ike-gateway t1
```

Configure Loopback IP

```
ACOS(config)# interface loopback 1
ACOS(config)# ip address 170.1.1.19 255.255.255.0
```

Limitations

- In ACOS 6.0.0,
  - only VPN stateless mode is supported.
  - trigger IKE and IPsec negotiation from Scaleout side is not supported.
  - start rekey from Scaleout is not supported.
- Local IP address in VPN IKE-gateway should be configured as loopback IP, and all the devices in cluster should have the same loopback IP address.
- For stateless mode in Scaleout, the packet for one tunnel will be processed in different devices in cluster, so ESP sequence number of outbound IPsec traffic maybe not in order.
- For counters of IPsec SA, the counters of multiple devices will not be aggregated. When the show command of IPsec is executed, the counter of current device can be viewed.
- The newly added device does not contain the information of IKE SA that had negotiated up before, show vpn ike-sa will not display these IKE SAs.
- Since each device has to store all ike-sa, so the IPsec max specification supported by Scaleout is that of a single device.
- Addition of Chassis platform into Scaleout is not supported.
- When IKE/IPsec SA information is synchronized between cluster, the synchronization message is not encrypted.
- Management Interface VPN IPsec is not supported.

Performance and Scalability

The following topics are covered:

Processing the IPsec Modes

The IPsec VPN supports both stateful mode and stateless mode. Stateless mode results in better performance than stateful mode. Stateful mode is slower but is used for services such as SLB where state information is needed.

The following topics are covered:

NOTE: Enabling stateless mode results in better ACOS performance. By default, the stateless mode is enabled.

Stateful Mode

For IPsec stateful mode, a setup session is needed for IPsec traffic. If no session is present, the IPsec module sets up the session, on its own. In this mode, the performance is slower as compared to stateless but more functions are available.

Use the following CLI command to configure the stateful mode:

```
ACOS(config)#vpn stateful-mode
```

Stateless Mode

If IPsec stateless mode is enabled, no setup sessions are needed for IPsec traffic. For outbound traffic, ACOS just does a route lookup, encapsulates the IP packet, and sends packet out. For inbound traffic, ACOS decapsulates the IP packet, performs route lookup, and sends the packet out. By default, stateless mode is enabled.

IPsec Acceleration

Thunder devices with a cryptographic hardware accelerator speeds up IPsec processing and provide a higher throughput for IPsec traffic. IPsec acceleration

supports IPsec packet encryption/decryption and NAT encapsulation/decapsulation. IPsec acceleration provides faster response time and higher server scalability.

NOTE: The cryptographic hardware accelerator is shared between SSL and IPsec. Some cores of the accelerator are reserved for IPsec, due to which SSL performance may be impacted.

Configuring Core Allocation for IPsec Acceleration

The number of cores and memory are primarily used because the hardware cryptographic accelerators are shared with other modules, such as SSL. If only IPsec is being used, set the crypto-cores to the maximum value, and crypto-mem to 100 percent.

NOTE: If no core is allocated for IPsec, IPsec encryption/decryption is software supported. If however, no core is allocated for SSL, the SSL request fails.

Use the following command to define the amount of core and memory for IPsec acceleration:

```
ACOS(config)#system ipsec ?
  crypto-core          Crypto cores assigned for IPsec processing
  crypto-mem           Crypto memory percentage assigned for IPsec
processing
```

NOTE: The `crypto-core` and `crypto-mem` options are not supported if the SSL module mode is set to ‘QAT.’ Similarly, these commands do not take effect if the SSL module mode is set to ‘N5-New.’ To view the SSL module mode, use 'show hardware' command in 'Global Configuration Mode', see *Command Line Interface Reference*.

Speed on Tunnel Interface

Reading IPsec tunnel interface speeds dynamically via SNMP MIB interface objects of the ifTable and ifXTable is supported.

To configure speed on the tunnel interface:

```
ACOS#conf
ACOS(config)#interface tunnel 1
ACOS(config-if:tunnel:1)#speed ?
  <1-200>  Speed in Gbit/Sec (Default 10 Gbps)
```

To verify the configured speed:

The following topics are covered:

Using the CLI

```
ACOS(config-if:tunnel:1)#show interface tunnel 1
Tunnel 1 is up, line protocol is up
  Tunnel interface, Address is 001f.a005.c590
  Internet address is 10.1.1.1, Subnet mask is 255.255.255.0
  IP MTU is 1500 bytes, Speed is 5 Gb per second
  Received    0 packets   0 bytes
    Error 0 packets
  Transmitted 0 packets   0 bytes
    Error 0 packets
```

Using the AXAPI

```
{
  "tunnel": {
    "ifnum":1,
    "mtu":1500,
    "speed":5,
    "uuid":"8e1b341c-ed63-11e5-b0cb-001fa0047f5c",
    "ip": {
      "address": {
        "dhcp":0,
        "ip-cfg": [
          {
            "ipv4-address":"10.1.1.1",
            "ipv4-netmask":"255.255.255.0"
```

```
          }
        ]
      },
      "generate-membership-query":0,
      "uuid":"97611ee6-f06b-11e5-b0c7-001fa0047f5c",
      "a10-url":"/axapi/v3/interface/tunnel/1/ip"
    }
  }
}
MIB Browser
.1.3.6.1.2.1.2.2.1.5.5201
Name/OID: ifSpeed.5201; Value (Gauge): 4294967295
.1.3.6.1.2.1.31.1.1.1.15.5201
Name/OID: ifHighSpeed.5201; Value (Gauge): 5000
```

Improved CPU Utilization for Single IPsec Tunnel

On Thunder devices with FPGA, when IPsec uses AES256 for encryption and SHA256 for HMAC, the FPGA handles the decryption and HMAC validation and sends the inner packet to the upper layer applications. This improvement is for a single ESP tunnel using AES-256 and SHA-256 cryptographic algorithms only and the number of tunnels should be 16 or less.

Use the following `system ipsec fpga-decrypt <enable | disable>` command to enable/disable this feature:

```
ACOS(config)#system ipsec fpga-decrypt ?
  enable   Enable FPGA decryption offload
  disable  Disable FPGA decryption offload
ACOS(config)#system ipsec fpga-decrypt enable
IPsec FPGA decryption assistance is on
Settings will take effect on reboot.
ACOS(config)#system ipsec fpga-decrypt disable
IPsec FPGA decryption assistance is off
Settings will take effect on reboot.
```

The `system ipsec fpga-decrypt <enable | disable>` command is an operational command and not for configuration. The user can use it to enable/disable the feature, but it does not appear in show running configuration.

NOTE: Disable CPU load sharing using the `system cpu-load-sharing disable` command. Do not use IPsec packet round-robin with this feature.

Enhanced IPsec Tunnel Traffic for L3-DSCP

This feature helps in enhancing the IPsec tunnel traffic delivering capabilities for Layer-3 DiffServ/Differentiated Services Code Point (L3-DSCP), which allows in categorizing or prioritizing the traffic by next-hop devices.

The upgraded capabilities help in determining the following:

- Preserving the incoming marking, which is already supporting.
- Overriding the incoming marking with global value, which helps in resetting the marking.
- Overriding the incoming marking individually per tunnel.
- When the ACOS receives a packet and decides that the packet must come out of an IPsec tunnel, then the ACOS automatically encrypts the packet and adds a new IP header, also known as the outer IP header.
- The outer IP header sets the source IP address as the IP of the VPN IKE-gateway, and the destination IP address as the IP of the peer VPN gateway.

Despite this, there are some other IP header fields that need to be considered, such as the DSCP.

- The DSCP is a higher 6-bit field in the 8-bit "*Priority and Type of Service*" field, which is used for classifying the layer 3 traffics.
- Different DSCP values indicate different L3 packet drop probabilities, which directly affects the quality of service (QoS).

As the original IP header is encrypted and unreadable by the next-hop routers, it is recommended to copy the DSCP field from the original IP header to the new IP header.

This behavior is called as the IPsec DSCP preserving, which is already supported by ACOS.

For classifying and prioritizing the IPsec traffic, it is necessary to manually set the DSCP value for each of these individual tunnels.

It is to be noted, as a restriction, this feature is not yet supported by ACOS and the design is proposed to address this.

The following topics are covered:

Feature Description

This feature and its descriptions help the user to perform and understand the following:

- It helps the user in enabling the DSCP set for each individual IPsec tunnel when creating an outer IP header during encapsulation.
- The DHCP requires both IPv4 and IPv6 traffic support to perform this feature, which must be preserved during the packet fragmentation.

CLI Configuration

The following topics are covered:

Configuration Commands

User Recommendation

The following important points must be considered by the user for this feature:

- By default, the value of the DSCP of the outer IP header is copied from the inner IP header.
- The user can also manually set the DSCP value of the outer IP header with the following commands.

CLI Options

The following are the steps and representations of some of the new commands, which are now added to the parameter VPN IPsec:

1. Go to a VPN IPsec module:
```
(config)# vpn ipsec test_ipsec
```
2. Set DSCP to a fixed value:
```
(config-ipsec:test_ipsec)# dscp [value]
```
3. Unset the DSCP. And the Outer DSCP is copied from the original IP header.
```
(config-ipsec:test_ipsec)# no dscp [value]
```

Show Commands

The following is the new show command set for this feature.

```
show running-config vpn ipsec ipsec_test
```

This result displays the DSCP values if it is set.

GUI Configuration

This section describes how to configure a VPN tunnel by using the ACOS GUI.

To create a VPN Tunnel:

1. Navigate to **Security > IPsec VPN > VPN Tunnels**.
2. Click **Create**.
   The **Create VPN Tunnel** page is displayed.
3. Enter the following details to create a VPN Tunnel.
   - Name
   - IKE Gateway

- Traffic Selector
- IP Version
- Local IP/IPv6
- Netmask
- Local Port
- Remote IP/IPv6
- Netmask (Remote)
- Remote Port
- Remote Protocol
- Lifetime
- Encryption
- Reference: Encryption Algorithm
- Binding
- Tunnel Interface
- Nexthop IP
- Diffie-Hellman Group
- Lifebytes
- Encap Mode
- Encap Protocol
- Disable Sequence Number
- Anti Replay Window Size

4. Click **Create**.
   VPN Tunnel is created.

Licensing and Platforms

The following topics are covered:

Supported Platforms

All those platforms which support registration with the harmony controller must be supported.

Upgrading or Downgrading Results

The following are the important points, referring to either upgrading or downgrading the system, as per the impact of this feature:

- The command is expected to be ported to all future releases.
- Upgrading is an expected and applicable mode, which is not an issue.
- Downgrading will wipe out the command and malfunctioning.

Risks or Assumptions

The following is a list of risks or assumptions for this feature.

- The feature is only enabled for outbound IPsec traffic, which means, only those IPsec traffic elements that require encryption and encapsulation.
- The feature supports DSCP preserving and DSCP set for each individual tunnel.
- In the current phase of this feature, DSCP set for all tunnels (global reset) is not implemented.
- The feature is applicable to both IPv4 and IPv6.
- This feature is applied to both hardware and software encryption routines.

Limitations

The following is a list of limitations:

- As a process, the DSCP global reset is not implemented and each IPsec tunnel has its own DSCP value.
- As a requirement, the user needs to add CM configuration for DSCP, within the node “*vpn.ipsec*”.

IPsec Fragmentation

The ACOS device supports packet fragmentation for IPsec VPN.

The following topics are covered:

Pre-Encap Fragmentation Overview

Packets are fragmented into equal sizes before encapsulating in IPsec tunnel headers.

- Packet sizes are more uniform; remote gateway is not involved in reassembly
- Packet input is not the same as output
- Requires destination host to reassemble

Post-Encap Fragmentation Overview

Packets are fragmented after encapsulating in IPsec tunnel headers.

- Packets in are exactly the same as packets out
- Requires peer gateway to reassemble

Configuring Pre-Encap Fragmentation

Pre-encap fragmentation is enabled by default.

Configuring Post-Encap Fragmentation

Use the following command to configure:

```
ACOS(config)#vpn fragment-after-encap
```

TCP Maximum Segment Size Clamping

The TCP MSS Adjustment feature enables the configuration of the maximum segment size (MSS). Disable auto adjust TCP MSS to avoid fragmentation overhead.

```
ACOS(config)#vpn tcp-mss-adjust-disable
```

Maximum IPsec SA Based on the System Memory

This design is optimized for the maximum IPsec SA number that can be supported based on the system memory size. It also includes the support for configuring the maximum number of IPsec SA through CLI mode to provide a way to limit the IPsec memory usage by the user.

In the existing implementation, the following are the observations, which needed a reconsideration.

- In the current implementation, the maximum number of VPN IPsec SA is fixed.
- The user can at the maximum configure 20,000 VPN IKE gateway and IPsec configurations.
- It resulted in only supporting 20,000 IPsec SAs now.

In the revised implementation, the focus is on the following enhancements.

- Supporting a dynamic maximum number of VPN IPsec SA based on system memory size.
- Supporting the configuration of the maximum number of VPN IPsec SA by the user.
- Supporting a high increment in the maximum number of VPN IPsec SA from 20,000 to 100,000 (100k).

The following topics are covered:

Dependencies

In the existing system, the following are the points of restrictive actions.

- Both inbound/outbound SAs are counted as one.
- The current IPsec SA includes the inbound SA (key-in) and outbound SA (key-out).
- When the rekey action is performed, the older key-in and key-out are deleted after the termination time.
- The fixed maximum VPN IPsec SA number may not be matching with the ability of some of the low or middle-end platforms.

In the revised system, the focus is on the following enhancements.

- There is a need to dynamically set the maximum number based on the memory of a platform.
- If the customer has a large unit, but only in need of a limited number of IPsec SAs, then the allocation of the memory for that IPsec is based on the maximum number of IPsec SA which are decided or calculated by the system memory size.
- A CLI command is added to configure the maximum number of IPsec SA, through which the user can limit the IPsec memory usage based on the actual requirement.

Configuring IPsec SA and IKE Gateway

The following topics are covered:

Maximum Values

The following Table 9 tabulates the maximum default values of the IPsec SA and IKE Gateway configurations in various options for all the applicable system memory sizes.

Table 9 : Maximum Default Values of the IPsec SA and IKE Gateway Configurations

System Memory Size	MAX IPsec SA Config	MAX IKE Gateway Config	MAX IKE Gateway in Shared Config	MAX IKE Gateway in L3V Config	MAX IPsec Config	MAX IPsec in Shared Config	MAX IPsec in L3V Config
8G	2,000	2,000	2,000	8	2,000	2,000	8
16G	5,000	5,000	5,000	20	5,000	5,000	20
32G	10,000	10,000	10,000	40	10,000	10,000	40
64G	20,000	20,000	20,000	80	20,000	20,000	80
128G	64,000	64,000	64,000	256	64,000	64,000	256
256G	100,000	100,000	100,000	400	100,000	100,000	400
384G	100,000	100,000	100,000	400	100,000	100,000	400
512G	100,000	100,000	100,000	400	100,000	100,000	400

Reference for the Maximum Values

MAX IPsec SA

This is the maximum IPsec SA number of the whole system. This cannot be extended beyond as it is the maximum available value and is also a limitation.

MAX IKE Gateway Config

This is the maximum VPN IKE-Gateway configuration number of the whole system.

MAX IKE Gateway in Shared

This is the maximum VPN IKE-Gateway configuration number allowed in the shared partition.

MAX IKE Gateway in L3V

This is the maximum VPN IKE-Gateway configuration number allowed in the L3V partition. When using the CLI command to configure the max IPsec SA number, the system allows a value higher than this limit for the L3V partition.

NOTE: You can configure only four tunnel interfaces in the L3V partition and 128 in the shared partition.

MAX IPsec Config

This is the maximum VPN IPsec configuration number of the whole system.

MAX IPsec in Shared

This is the maximum VPN IPsec configuration number allowed in the shared partition.

MAX IPsec in L3V

This is the maximum VPN IPsec configuration number allowed in L3V partition. When using the CLI command to configure the max IPsec SA number, the system allows a value higher than this limit for the L3V partition.

Cavium N3 and Cavium N5

The following Table 10 tabulates the corresponding Cavium N3 and Cavium N5 context memory usage information.

Table 10 : Cavium N3 and Cavium N5 Context Memory Usage Information

System Memory Size	Max IPsec SA	N3 Context Memory Usage	N5 Context Memory Usage
8G	2,000	2,000 * 1K = 2M	2,000 * 2K = 4M
16G	5,000	5,000 * 1K = 5M	5,000 * 2K = 10M
32G	10,000	10,000 * 1K = 10M	10,000 * 2K = 20M
64G	20,000	20,000 * 1K = 20M	20,000 * 2K = 40M
128G	64,000	64,000 * 1K = 64M	64,000 * 2K = 128M
256G	100,000	100,000 * 1K = 100M	100,000 * 2K = 200M
384G	100,000	100,000 * 1K = 100M	100,000 * 2K = 200M
512G	100,000	100,000 * 1K = 100M	100,000 * 2K = 200M

Reference for Cavium N3 and Cavium N5

Cavium N3

The standard one N3 context memory block size is 512, in which the user needs one for key-in and one for key-out, resulting into the usage of 512 * 2 = 1K context memory for one IPsec-SA.

Cavium N5

The standard one N5 context memory block size is 1024 (1K), in which the user needs one for key-in and one for key-out, resulting into the usage of 1024 * 2 = 2K context memory for one IPsec-SA.

Resource Manager Infra

The default number for IPsec SA and its configuration is set by using the resource manager infra. The implementation is also done for the maximum IPsec SA number to configure in the CLI command on the resource manager infra as well.

The following Table 11 tabulates the minimum, the maximum and the default IPsec SA number values, based on the system memory size, which are supported by the resource manager.

Table 11 : System Memory Size, Supported by the Resource Manager Infra

System Memory Size	Minimum	Maximum	Default
less or equal 8G	8	2,000	2,000
12G	20	5,000	5,000
16G	20	5,000	5,000
24G	40	10,000	10,000
32G	40	10,000	10,000
64G	80	20,000	20,000
128G	256	64,000	64,000

CLI Configuration Commands

The following topics are covered:

Configuration Commands

The following CLI command configuration is added to allow the user to configure the current maximum IPsec SA number.

```
AX1030-105.33(config)#system resource-usage ?
  aflex-table-entry-count            Total aFleX table entry in the system
  auth-portal-html-file-size        Specify maximum html file size for
each html page in auth portal (in KB)
  auth-portal-image-file-size        Specify maximum image file size for
default portal (in KB)
  authz-policy-number                Specify the maximum number of
authorization policies
  class-list-ac-entry-count          Total entries for AC class-list
  class-list-ipv6-addr-count         Total IPv6 addresses for class-list
  ipsec-sa-number                  Specify the maximum number of IPsec SA
  l4-session-count                   Total Sessions in the System
  max-aflex-authz-collection-number  Specify the maximum number of
collections supported by the aFleX authorization
  max-aflex-file-size                Set maximum aFleX file size
  nat-pool-addr-count                Total configurable NAT Pool addresses
in the System
  radius-table-size                  Total configurable CGNV6 RADIUS Table
entries
  ssl-context-memory                  Total SSL context memory needed in units of
MB. Will be rounded to the closest multiple of 2MB
  ssl-dma-memory                     Total SSL DMA memory needed in units
of MB. Will be rounded to the closest multiple of 2MB
  visibility                         Configure System Resource Usage for
visibility
AX1030-105.33(config)#system resource-usage ipsec-sa-number ?
  <20-5000>  Specify the maximum number of IPsec SA
AX1030-105.33(config)#system resource-usage ipsec-sa-number 1000
Changes will come into effect next time you reload the Software.
AX1030-105.33(config)#
```

Show Command

The user can find the details of all the show command that are needed for operation support in the IPsec SA number information as in `show system resource-usage`.

```
AX1030-105.33(config)#show system resource-usage

Resource                              Current    Default    Minimum
Maximum
------------------------------------------------------------------------------
---
l4-session-count                      33554432   33554432   8388608
67108864
nat-pool-addr-count                   500        500        500        4000
class-list-ipv6-addr-count            2048000    2048000    2048000
4096000
class-list-ac-entry-count             1024000    1024000    1024000
2048000
auth-portal-html-file-size            20         20         4          120
auth-portal-image-file-size           6          6          1          80
max-aflex-file-size                   32         32         16         256
aflex-table-entry-count               102400     102400     102400
8388608
max-aflex-authz-collection-number     512        512        256        4096
radius-table-size                     4000000    4000000    2000000
4000000
monitored-entity-count                14048      14048      1520       28384
authz-policy-number                   128        128        32         2000
ipsec-sa-number                       1000       5000       20         5000
```

Limitations

The following topics are covered:

ACOS

The ACOS only supports at the maximum 4k static router configuration and at the maximum 64k entries in FIB table. The IPsec VPN is routing based, it can only distribute the traffic depending on the route. Hence, not enough route record to

distribute the traffic for 100k IPsec SAs. There is a need to enlarge the static route and FIB table limitation.

VPN

With over 10k IKE gateway and IPsec configurations, the system boot-up speed needs an increase.

Monitoring

The following topics are covered:

IPsec Tunnel Interface Statistics for SNMP

ACOS provides SNMP monitoring for IPSec tunnel interfaces. The tunnel interfaces are included in the following SNMP MIB tables:

- A10-AX-MIB::axInterface
- A10-AX-MIB::axInterfaceStat
- ifTable (RFC 1213)
- ifXTable (RFC 2863)

Supported ifTable MIBs

The following ifTable MIBs provide in/out packet statistics for IPSec tunnel interfaces:

```
ifEntry OBJECT-TYPE
              SYNTAX  IfEntry
              ACCESS  not-accessible
              STATUS  mandatory
              DESCRIPTION
                      "An interface entry containing objects at the
                      subnetwork layer and below for a particular
                      interface."
              INDEX   { ifIndex }
              ::= { ifTable 1 }
          IfEntry ::=
              SEQUENCE {
                  ifIndex
                      INTEGER,
                  ifDescr
                      DisplayString,
                  ifType
                      INTEGER,
                  ifMtu
                      INTEGER,
                  ifAdminStatus
                      INTEGER,
                  ifOperStatus
                      INTEGER,
```

```
                ifInOctets
                    Counter,
                ifInUcastPkts
                    Counter,
                ifInErrors
                    Counter,
                ifOutOctets
                    Counter,
                ifOutUcastPkts
                    Counter,
                ifOutErrors
                    Counter,
            }
```

Supported ifXTable MIBs

The following ifXTable MIBs provide in/out packet statistics for IPSec tunnel interfaces:

```
ifXEntry        OBJECT-TYPE
    SYNTAX      IfXEntry
    MAX-ACCESS  not-accessible
    STATUS      current
    DESCRIPTION
            "An entry containing additional management information
            applicable to a particular interface."
    AUGMENTS    { ifEntry }
    ::= { ifXTable 1 }
IfXEntry ::=
    SEQUENCE {
        ifName                  DisplayString,
        ifHCInOctets            Counter64,
        ifHCInUcastPkts         Counter64,
        ifHCInMulticastPkts     Counter64,
        ifHCOutOctets           Counter64,
        ifHCOutUcastPkts        Counter64,
        ifHCOutMulticastPkts    Counter64,
    }
```

NOTE: The **ifHCInMulticastPkts** and **ifHCOutMulticastPkts** counters will always be zero, as tunnel interfaces do not support multicast packets.

SNMP Trap for IPsec Tunnel Up/Down

The SNMP trap for IPsec Tunnel Up/Down sends an event when the state of the IPsec tunnel interface changes. For example, when the state of an IPsec tunnel interface changes from up to down or vice versa, the SNMP trap event is sent.

IPsec Bandwidth Command

A bandwidth rate statistic command is available on a per-tunnel basis. See the **/axapi/v3/vpn/ipsec/<tunnel>/rate** endpoint in the *aXAPI Reference* and `show counters-rate` command for further information.

64-bit SNMP Bandwidth Counters

64-bit bandwidth packet and byte counters for tunnel interface MIB are supported. The following counters are supported:

Bandwidth - 64-bit Counter:

```
ifHCInOctets
1.3.6.1.2.1.31.1.1.1.6
Bandwidth - 64-bit Counter
ifHCInUcastPkts
1.3.6.1.2.1.31.1.1.1.7
Bandwidth - 64-bit Counter
ifHCInMulticastPkts
1.3.6.1.2.1.31.1.1.1.8
Bandwidth - 64-bit Counter
ifHCOutOctets
1.3.6.1.2.1.31.1.1.1.10
Bandwidth - 64-bit Counter
ifHCOutUcastPkts
1.3.6.1.2.1.31.1.1.1.11
```

```
Bandwidth - 64-bit Counte
ifHCOutMulticastPkts
1.3.6.1.2.1.31.1.1.1.12
```

Using the GUI for IPsec VPN Monitoring

To use the GUI to view statistics for System, IKE Global, IKE Gateway, IPsec Tunnel and Tunnel Interface resources:

1. Navigate to **Security** > **IPsec VPN**
2. Select the **Statistics** tab, click on **System**, **IKE Global**, **IKE Gateway**, **IPsec Tunnel**, or **Tunnel Interface** tab for the specified view.
3. For IKE Gateway, IPsec Tunnel, and Tunnel Interface, choose a valid object from the drop-down list.

Troubleshooting

The following topics are covered:

If the IPsec VPN Tunnel is Not Up

NOTE: Four levels of IKE debug logs are present. In most cases, level 1 (Basic Packet Negotiation) must suffice. The debug commands do not bring down the tunnel.

The information contained in the VPN logs provide detailed VPN messages as well as state changes depend upon the log level. Once the VPN log level is set and tunnel negotiation is triggered, the VPN log file is generated. If the VPN log file exists, new log messages are appended to the file. VPN log files are persistent across reload and reboot.

NOTE: VPN log is disabled by default, and no log files are generated.

The following is a list of common issues and solutions:

1. IKE proposal mismatch

   If you see a log that says “No proposal chosen,” the peer device is making the decision for the tunnel to fail. In this case, IKE SA negotiation cannot initiate, and traffic continues to flow unencrypted. To fix this issue, login to the peer ACOS device and enter the `show vpn log` command. In the output, check the “received proposals” and “configured proposals” and make sure the encryption, hash and key exchange (DH) parameters and traffic selectors on both sides are exactly the same.

2. IKE Authentication fail

   For preshared key, the exact same key must be configured on both sides. For certificates, check certificate configuration on both sides. The debug level for certificate validation, certificate revocation, and caching is level 2 when running the `debug vpn level` command.

3. IKE version mode mismatch

   Make sure both sides use the same IKE version.

If the IPsec VPN Tunnel is Up

Go through the following steps to troubleshoot:

1. Isolate where the packet is not getting forwarded to correctly (device 1 or 2).
2. Check the routing for the inner destination IP on both gateways.
3. Use `ping` and `traceroute` commands from client to server.
4. See which interfaces the packet is coming in and going out of the device. Make sure the input and output interfaces are configured as expected. Make sure the local and remote IP addresses specified on both devices are correct. Use the following commands to debug:
   - debug packet
   - debug monitor

   NOTE: When NAT-T is disabled, ESP packets use IP protocol 50. In this case, the NAT device may inappropriately apply NAT to the packets. When NAT-T is enabled, IPsec packets use UDP (protocol 17) port 4500.
5. Use the `debug tunnel` along with `debug monitor` command to display the relevant flow paths for the tunnel. The `debug tunnel` command shows output only if the packet is flowing through the tunnel.

VPN Log Filter for Troubleshooting an Individual IKE Gateway

The VPN configuration and negotiation process generates logs, which can be viewed by using the show VPN log command. This feature is used to filter these VPN logs by using the name of the gateway.

This feature all about the filtering process through the name of Ike-gateway. The following is a list of important points of for this feature:

- As the user demand is increasing, there is a possibility of excess deployment of over 1,000 tunnels of IPsec VPN.
- As it is a large exercise, with an excess amount of same process time required to debug all the tunnels to resolve the problem, there is a scope for implementing the VPN debugging.
- Currently, debugging a VPN by configuring tunnel or Ike -gateway names, as a filter is not supported.
- When the number of tunnel or Ike gateway configured is large, if the user wants to troubleshoot one tunnel or Ike gateway, the log printed by VPN contains the debug of all tunnels.

Analyzing the entire VPN log takes more time to resolve problems. Thus, there is a need to add filtering conditions to enable the users to filter debug information of VPN.

The following topics are covered:

Limitations

The following is a list of limitations for this feature:

- For one partition, at most one Ike-gateway name is supported at the same time.
- In the `show debug` command, displaying the configured ike -gateway name is not supported.

Feature Description

The following topics are covered:

Configurations

When more than one Ike gateway is configured, to get more efficient logs, the user can use Ike-gateway as a filter. The user must configure the following on ACOS to get the filtered IPsec VPN logs:

- Open debug VPN level and Ike-gateway filter.
- Configure more than one Ike-gateway and IPsec tunnel.
- Show a VPN log.

Example

The user can refer to the following configuration example:

```
debug vpn level 1
debug vpn ike-gateway SJ_to_LA_GW
!
interface tunnel 1
  ip address 100.100.10.2 255.255.255.0
interface tunnel 2
  ip address 200.200.20.2 255.255.255.0
vpn ike-gateway SJ_to_LA_GW
  auth-method preshare-key 123456
  local-id Site-SJ
  remote-id Site-LA
  encryption des hash md5
  local-address ip 34.1.1.3
  remote-address ip 34.1.1.4
vpn ike-gateway test2
  auth-method preshare-key 123456
  local-id site-1
  remote-id site-2
  encryption des hash md5
  local-address ip 35.1.1.3
  remote-address ip 35.1.1.4
vpn ipsec SJ_to_LA_TUN1
  dh-group 1
  encryption aes-128 hash sha1
```

```
  traffic-selector ipv4 local 53.1.1.0 255.255.255.0 remote 46.1.1.0
255.255.255.0
  bind tunnel 1 100.100.10.3
  ike-gateway SJ_to_LA_GW
vpn ipsec ipsec_test_2
  dh-group 2
  encryption aes-128 hash sha1
  traffic-selector ipv4 local 53.1.1.0 255.255.255.0 remote 46.1.1.0
255.255.255.0
  bind tunnel 2 200.200.20.3
  ike-gateway test2
ping 200.200.20.3
ping 100.100.10.3
show vpn log from-start
VMServer-105.233(config)#show vpn log from-start
…
…
…
VMServer-105.233(config)#
```

Processing Cached Logs

The following is a list of important points for this feature:

- Some of the negotiation messages cannot find the name of the ike-gateway due to some configuration errors.
- These logs do not belong to any ike-gateway, but they are helpful for troubleshooting, so these logs, by default, are recorded.
- The user can control whether these logs are recorded or not by adding “**strict**” after the ike-gateway name.

CLI Configuration

The following topics are covered:

Configuration Commands

The following is the configuration command set for this feature. The user must add the new option “Ike-gateway” to debug the VPN command.

```
VMServer-105.233#debug vpn ?
  level  Debug Level
  ike-gateway Specify a ike gateway name
VMServer-105.233#debug vpn ike-gateway ?
  WORD<length:1-31>  IKE gateway name
VMServer-105.233#debug vpn ike-gateway SJ
  strict  Only record the logs that can match Ike-gateway name
  <cr>
VMServer-105.233#debug vpn ike-gateway SJ
```

The following observations are also noted:

It is also noticed that some of the logs cannot be determined, which Ike-gateway belongs to.

But they are considered as useful logs for troubleshooting and are recorded in the log by default.

If these logs records are not needed, then the user can opt to configure the “strict” command, after Ike-gateway name to ignore them.

```
VMServer-105.233#debug vpn ike-gateway SJ
  strict  Only record the logs that can match Ike-gateway name
VMServer-105.233#debug vpn ike-gateway SJ strict
```

Show Commands

The following is the show command set for this feature.

```
VMServer-105.233#show debug
debug vpn (level 4) is on
debug vpn ike-gateway is on
```

Licensing and Platforms

The following topics are covered:

Upgrading or Downgrading Results

The following are the important points, referring to either upgrading or downgrading the system, as per the impact of this feature:

The command is expected to be ported to all future releases.

Upgrading is an expected and applicable mode, which is not an issue.

Downgrading will wipe out the command and malfunctioning.

Glossary

A

anti-replay

An IPsec sub-protocol used for preventing hackers from injecting or modifying packets being sent from a source to a destination. It is part of Internet Engineering Task Force (IETF) suite.

C

cipher

An information security algorithm that performs encryption or decryption of code. It follows a series of well-defined steps called as the encipherment procedure.

clamping

The process of changing the maximum segment size (MSS) of TCP connections passing through links with an MTU lower than the default ethernet value.

core allocation

The process of assigning a set of all feasible properties to the core.

CRL

Certificate Revocation List. A list of web certificates being revoked by the issuing Certificate Authority (CA) prior to their scheduled expiration date. Certificates in the CRL can no longer be trusted.

D

decapsulation

The process of unlocking encapsulated data sent in the form of packets across a communication network. It is observed as opening of a capsule of wrapped-up data.

E

encapsulation

The process of retrieving data from one protocol and converting it into another protocol, thereby allowing the data to be sent across a network.

F

FIPS

Federal Information Processing Standards. A set of standards describing encryption algorithms, document processing, and related IT standards that can be used within a network.

fragmentation

An IP process which dismantles packets into smaller pieces called fragments. It helps in passing these fragments through a link with a smaller MTU than the actual packet size. The fragments are further reassembled by the receiving host.

I

IKE

Internet Key Exchange. An IPSec standard protocol used for securing VPN negotiation and remote hosts or accessing network.

IPv4

The fourth version of the Internet Protocol used as a core protocol in standardized internetworking methods over the Internet and packet-switched networks.

IPv6

The sixth and most recent version of Internet Protocol, that works as a communications protocol to provide location system and identification for computers on networks and to route traffic over the Internet.

K

keying

A process where two entities authenticate each other and exchange the required key material.

L

L3V

Layer 3 Virtualization. A virtualization layer that allows organizations to utilize the same IP address ranges for ensuring that the multi-tenant data center architecture gets the flexibility similar to that of a independently-deployed device.

M

MTU

Maximum Transmission Unit (MTU) is the largest packet size, defined in the octets or eight-bit bytes, which can be sent in a packet-based network or the internet. TCP uses the MTU for determining the maximum size of each packet in a given transmission.

N

NAT

Network Address Translation. A method of reallocating one IP address space into another by changing the network address information in the IP header when the packets are still being transmitted across a traffic routing device.

O

OCSP

Online Certificate Status Protocol. An Internet protocol that obtains the revocation status of an X.509 digital certificate according to RFC 6960. It is on the track of Internet standards.

P

packet flow

A sequence of packets sent from a source to a destination over packet-switching networks, across a host, a broadcast domain, or a multicast group.

peer

A equally privileged and equipotent participant in the application or over the network.

R

re-keying

The process of modifying the session key, which is the encryption key of an active communication, and limiting the amount of data being encrypted with the same key.

S

SCEP

Simple Certificate Enrollment Protocol. A protocol standard for certificate management, primarily used for Certificate-based authentication and providing access to services through encryption via certificates.

SLB

The process of distributing high-traffic sites among multiple servers by using a network-based hardware or software-based device. SLB intercepts the traffic for a website and reroutes it to different servers for attaining data-load equilibrium.

SNMP

Simple Network Management Protocol. A standard Internet Protocol used for collecting

and managing information on managed devices over IP networks and for changing the information to modify device behavior.

standby

A action which provides redundancy for a local subnet.

T

TCP

Transmission Control Protocol. Key part of the main IP suite protocols used during initial network implementation.

tunnel

The component of tunneling process where a protocol allows the secure transmission of data from one network to another. It enables communication of a private network communications with a public network through encapsulation.

V

VPN

Virtual Private Network. A private network extended across a public network, which enables users to transmit data across public or shared networks in a manner where their computing devices were being directly-connected to the private network.

VRRP-A

Virtual Router Redundancy Protocol-A. A computer networking protocol for providing an automated assignment of available IP routers to the participants (hosts). It boosts the reliability and availability of routing paths through automatic selection of default gateways on an IP subnetwork.

Contact Us