Technical Guide

Internet Protocol Security (IPsec)

Feature Overview and Configuration Guide

Introduction

This guide describes Internet Protocol Security (IPsec) and its configuration. IPsec is a protocol suite for securing IP networks by authenticating and encrypting IP packets. IPsec protects one or more paths between a pair of hosts, a pair of security gateways, or a security gateway and a host. A security gateway is an intermediate device, such as a router or firewall that implements IPsec. The connection between two devices using IPsec to protect data is called a VPN (Virtual Private Network).

Products and software version that apply to this guide

This guide applies to AlliedWare™ Plus products running version **5.4.5** (IPsec basic features) or later.

To see whether a product supports IPsec, see the following documents:

- The product's Datasheet
- The product's Command Reference

These documents are available from the above links on our website at alliedtelesis.com.

These features are available in later releases:

- Version 5.5.3-0.x and later support certificate-based authentication
- Version 5.5.2-1.x and later support the **tunnel inline-processing** command.
- Version 5.5.1-2.x and later support the **tunnel oper-status-control** command.
- Version 5.4.9-2.1 and later lets you disable rekeying of unused IPsec SAs
- Version 5.4.9-0.1 and later support IPv4 traffic selectors on IPsec IPv6 tunnels.
- Version 5.4.8-1.1 and later support strict selector pairing.
- Version 5.4.8-0.x and later support the **tunnel security-reprocessing** command.
- Version 5.4.7-2.1 and later support index range for interface tunnels from 0-255 to 0-65535.
- Version 5.4.6-1.x and later support advanced IPsec specific features.

Related documents

For more information about creating firewall rules, see:

- Getting Started with the Device GUI on UTM Firewalls
- Firewall and Network Address Translation (NAT) Feature Overview and Configuration

Content

IPsec introduction

This section describes IPsec functionality and operation. It provides an explanation about default and custom profiles, how to work with dynamically assigned IP addresses, and how traffic is routed via the Virtual Tunnel Interface (VTI) and automatically encrypted.

What does IPsec do?

IPsec provides the following security services for traffic at network Layer 3 (IP):

- Data origin authentication—Identifying who sent the data
- Confidentiality (encryption)—Ensuring that the data has not been read en route
- Connectionless integrity (authentication)—Ensuring the data has not been changed en route
- Replay protection—Detecting packets received more than once to help protect against denial of service attacks

The operation of IPsec is based upon negotiated connections between peer devices. These connections are called Security Associations.

A Security Association (SA) is a one-way connection that provides security services between IPsec peers. For example, SAs determine the security protocols and the keys. An SA is uniquely identified by a combination of:

- A random number called the Security Parameter Index (SPI)
- An IP destination address
- A security protocol header, either AH (Authentication Header) or ESP (IPsec Encapsulating Security Payload)

You can choose IPsec in tunnel mode to implement a site-to-site VPN. A site-to-site VPN connects two sites together, for example a branch office to a head office, by providing a communication channel over the Internet. This saves a company having to pay for expensive leased lines. Employees gain full access to all company resources as if they were physically in the office connected to the corporate LAN.

IPsec provides secure protection of IPv4, IPv6, GRE, L2TP/PPP traffic (by using IPsec in transport mode) that traverses the VTI. AlliedWare Plus firewalls and routers support the following IPsec features:

- IPsec Encapsulating Security Payload (ESP)
- IKEv2 (Internet Key Exchange version 2) The default profile is now exclusively IKEv2 and it will not respond to IKEv1 requests. Custom ISAKMP profiles for IKEv1 peers need to be explicitly created.
- Pre-defined default ISAKMP (Internet Security Association and Key Management Protocol) and IPsec profiles based on current recommended parameters
- IKEv1 Main and Aggressive modes
- IKEv2 INIT, AUTH, CREATE_CHILD_SA and INFORMATIONAL Exchanges
- Configurable phase 1 local and remote IDs using IP address or Fully Qualified Domain Name (FQDN)
- Pre-shared key authentication using optionally encrypted shared keys identified by hostname or IPv4 or IPv6 address
- Dead Peer Detection (DPD) with a default 30-second polling timer
- Automatic NAT-Traversal negotiation
- Trace debugging of ISAKMP and IPsec negotiation
- Counters for both ISAKMP and IPsec
- Display of ISAKMP and IPsec SAs
- IPsec profile can be specified per VTI

The following types of tunnels can be used with IPsec:

- IPsec VTI using IPsec in IPv4 tunnel mode (IPv4 in IPv4)
- IPsec VTI using IPsec in IPv6 tunnel mode (IPv6 in IPv6)
- Protection of GRE based VTI traffic using IPsec in transport mode
- Protection of GRE IPv6 based VTI traffic using IPsec in transport mode
- Protection of L2TPv3 Ethernet Pseudowires based VTI traffic using IPsec in transport mode
- Protection of L2TPv2 PPP based VTI traffic using IPsec in transport mode

Default profiles

The processes that bring up and operate secure VPNs involve a number of different algorithms. These are encryption algorithms, key-exchange methods, anti-tamper checking algorithms, and so on. When two ends of a VPN are establishing their secure connection, they need to go through a negotiation, in which they agree which algorithms they will use for each of the component processes. This is a matter of them proposing options, choosing their preference from the proposed options, and confirming each other's choices.

A particular collection of algorithms, offered as an option in a proposal, is referred to as a **Transform**. The full set of transforms that are offered is referred to as a **Profile**.

The default profile used by AlliedWare Plus includes only the more secure FIPs 140-2 compliant algorithms to protect VPN traffic, and so does not support weaker non-compliant algorithms that may still be used by some legacy VPN devices.

The default profile contains a large set of pre-defined IPsec and ISAKMP transforms containing a wide variety of options that it can offer when negotiating an SA to a peer. This enables AlliedWare Plus firewalls and routers to inter-operate easily with a broad range of other vendors VPN equipment. No specific configuration is required to enable them to offer this large collection of options, it simply happens by default.

The negotiation process works down from the most secure cryptographic options through progressively less strong FIPs 140-2 compliant options until a match is agreed to. This process ensures the flexibility to inter-operate with all manner of modern peers with minimal configuration effort.

Default ISAKMP profiles

The default ISAKMP profiles are listed in order of preference:

Table 1: Default ISAKMP profiles

ATTRIBUTE	ENCRYPTION	INTEGRITY	GROUP	AUTHENTICATION
Transform 1	AES256	SHA256	14	Pre-shared
Transform 2	AES256	SHA256	16	Pre-shared
Transform 3	AES256	SHA1	14	Pre-shared
Transform 4	AES256	SHA1	16	Pre-shared
Transform 5	AES128	SHA256	14	Pre-shared
Transform 6	AES128	SHA256	16	Pre-shared
Transform 7	AES128	SHA1	14	Pre-shared
Transform 8	AES128	SHA1	16	Pre-shared
Transform 9	3DES	SHA256	14	Pre-shared
Transform 10	3DES	SHA256	16	Pre-shared
Transform 11	3DES	SHA1	14	Pre-shared
Transform 12	3DES	SHA1	16	Pre-shared

The entries in the default ISAKMP profiles table are:

- **Transform**: A transform specifies a set of algorithms to be used to protect ISAKMP messages, such as ISAKMP Key exchanges.
- **Encryption**: Symmetric key ciphers used for bulk data encryption. The Data Encryption Standard (DES) algorithm is no longer considered secure and was replaced by 3DES and now the Advanced Encryption Standard (AES). Encryption algorithms are used in order of preference: AES256, AES128, 3DES.
- **Integrity**: Secure Hash Algorithm (SHA) is used to check data integrity. Hash algorithms are used in order of preference: SHA256 then SHA1.
- **Group**: Diffie-Hellman (DH) groups determine the strength of the key used in the key exchange process. The DH groups are used in order of preference: 14 then 16.
- **Authentication**: Pre-shared key is a shared secret between peers that is used to authenticate each peer. This key is communicated to the peers by a separate process (possibly via a phone call).
- One other significant parameter is **Expiry**: Negotiate ISAKMP SA lifetime with a default of 24 hours.

Default IPsec profiles

The default IPsec profiles are listed in order of preference:

ATTRIBUTE	PROTOCOL	ENCRYPTION (ALL CBC)	INTEGRITY (ALL HMAC)
Transform 1	ESP	AES256	SHA256
Transform 2	ESP	AES256	GCM16
Transform 3	ESP	AES256	GCM8
Transform 4	ESP	AES256	SHA1
Transform 5	ESP	AES128	SHA256
Transform 6	ESP	AES128	GCM16
Transform 7	ESP	AES128	GCM8
Transform 8	ESP	AES128	SHA1
Transform 9	ESP	3DES	SHA256
Transform 10	ESP	3DES	SHA1

The entries in the default IPsec profiles table are:

- **Protocol**: The Encapsulating Security Payload (ESP) provides confidentiality (encryption) of data within IP packets.
- **Encryption**: Symmetric key ciphers used for bulk data encryption. The Data Encryption Standard (DES) algorithm is no longer considered secure and was replaced by 3DES and now the AES (Advanced Encryption Standard). Encryption algorithms are used in order of preference: AES256, AES128, 3DES.

- **Integrity (all HMAC)**: Secure Hash Algorithm (SHA) is used to check data integrity. Hash algorithms are used in order of preference: SHA256, Galois/Counter Mode (GCM), SHA1.

Other significant parameters for the transforms are:

- **Mode**: Protection of GRE- and L2TP/PPP based VTI traffic using IPsec in transport mode. Transport mode encapsulates the upper layer payload (such as Transmission Control Protocol (TCP) or User Datagram Protocol (UDP) of the original IP datagram. AH and ESP intercept packets from the Transport Layer that are intended for the Network Layer, protect the Transport header, and provide a configured security. Transport mode provides end-to-end security where the communications endpoint is also the cryptographic endpoint. The alternative to Transport mode is Tunnel mode. When Tunnel mode is used, IPsec encrypts the IP header and the payload, whereas Transport mode only encrypts the IP payload. All the transforms offered in the default profiles use Transport mode.
- **Expiry**: Negotiate IPsec SA lifetime with a default of 8 hours.

Custom profiles

There are cases where it is necessary for a VPN to use something other than the default profile.

These non-default cases could include:

- Peering to a device that may not support up to date secure cryptographic algorithms that are outside the range included in the default profile. For example a legacy peer device may be using older and weaker cryptographic options such as:
  - AES128 encryption algorithm
  - IKEv1 Main mode, or alternatively IKEv1 Aggressive mode (for key exchange with peers with dynamic IP addresses)
  - weaker Diffie-Hellman groups, such as DH group 2, for determining the strength of the key used in the key exchange process
  - older Secure Hash Algorithms, such as SHA-1 for checking data integrity
- If a business has a security policy that requires the negotiation of only a narrow set of cryptographic options - the default profile may offer too many options, and the set of offered options needs to be reduced to just the options that comply with the business security policy.

To set up VPNs in these non-default situations, AlliedWare Plus firewalls and routers provide **Custom Profiles** for the configuration of the VPNs. These are profiles that inform the software to offer a non-default set of options for the processing of the packets passing through the VPN.

Custom profiles are configured for both IPsec and ISAKMP. Each profile can be configured to contain a specific set of cryptographic options to offer to the peer. Each profile can be configured with multiple encryption algorithms. This can include weaker cryptographic options that are not FIPs 140-2 compliant, to allow inter-operation with legacy devices.

When a custom profile is being used, the AlliedWare Plus firewalls and routers will offer the specific ISAKMP and IPsec transform options that are included in that profile. The custom profile replaces the default profile, rather than adding options to the default profile.

These custom profiles also support additional options, such as specific SA lifetimes, and PFS.

Custom ISAKMP profiles

Each custom ISAKMP profile is named, and contains a set of transforms. The options that can be configured on the profiles are:

- Mode:
  - IKEv2 as an initiator and responder
  - IKEv1 Main mode as an initiator and responder
  - IKEv1 Aggressive mode as initiator and responder
- IKE version is configurable for the profile as a whole
- Pre-Shared Key (PSK) Authentication is configurable as a whole
- DPD interval (time between messages) is configurable for the profile as a whole (default 30 seconds)
- ISAKMPv1 DPD timeout (after which all peer SAs are deleted) is configurable for the profile as a whole (default 150 seconds)
- Lifetime in seconds for each profile. This should be two-three times longer than the IPsec profile lifetime to ensure a stable network.
- Integrity algorithm (SHA1, SHA256 and SHA512) for each transform
- Encryption algorithm (3DES, AES128, AES192, AES256) for each transform
- Diffie-Hellman group (2,5,14,15,16,18) for each transform

An ISAKMP profile may be specified per peer IP address, and another ISAKMP profile may be specified for all dynamic peers. The default ISAKMP profile is used for all ISAKMP peers not otherwise specified.

Custom IPsec profiles

Each IPsec custom profile is named, and contains a configurable list of IPsec transforms in priority order. The parameters that can be configured on each transform are:

- SA lifetime in seconds
- SHA-1, SHA256, or SHA512 integrity algorithms
- GCM8, GCM12, or GCM16 integrity algorithms. These are well-suited for hardware optimization, thereby improving throughput.
- Encryption algorithm
- AES128, AES192, AES256 or 3DES
- Optional Diffie-Hellman groups 2, 5, 14, 15, 16, 18 for PFS
- Extended Sequence Numbers (ESN) are supported and will be automatically negotiated if supported by the peer device.

Perfect Forward Secrecy (PFS) ensures generated keys, e.g. IPsec SA keys, are not compromised if any other keys, such as ISAKMP SA keys, are compromised. This configurable option is disabled by default but can be configured with the groups above.

Working with dynamically assigned IP addresses

It is not unusual, in a hub-and-spoke network, for the main site to have a fixed static IP address on its WAN interface, whereas the remote site WAN interfaces have dynamically allocated IP addresses.

In this situation, the remote site devices will initiate the formation of the IPsec VPN. The remote sites know the main office's fixed IP to which they can initiate the connection, once the remote site WAN interface becomes operational, and the WAN IP is dynamically allocated.

On the remote site, the destination address of the virtual tunnel is the static WAN IP address of the main office router. The main office VPN firewall is configured with the command **tunnel destination dynamic**, since the destination IP address is dynamically allocated to the remote site peer is unknown.

The main office device will identify the incoming peer with the local name that the incoming peer provides. On the main office device, this will be configured as the tunnel remote name. On the remote office device this will be configured as the tunnel local name. The main office device will then learn the dynamic IP address of the remote office.

Disabling rekeying of unused IPsec SAs

From version 5.4.9-2.1 onwards, you can specify a rekey policy for an IPsec profile. This policy will be used to make a decision on whether the SA will rekey at its expiry.

The options are **always**, **never**, and **on-demand**. The on-demand option makes its decision based on whether the link has seen any traffic since the SA's last rekey. Note that the default behavior remains unchanged and is to always rekey.

The new options may be useful if you have a hub and spoke VPN topology and need to provision more than the maximum number of concurrent active VPNs supported by your AR-Series device. The new options age out unused VPNs more quickly, making more efficient use of the number of available VPNs.

To specify the rekey policy, use the following command in IPsec Profile Configuration mode:

```
awplus(config-ipsec-profile)# rekey {always|never|on-demand}
```

For example, to only rekey when traffic is detected over the interface, in the profile named ‘myprofile’, use the commands:

```
awplus(config)# crypto ipsec profile myprofile
awplus(config-ipsec-profile)# rekey on-demand
```

Traffic selectors

By default AlliedWare Plus uses a route based VPN, where the VPN is terminated via a VTI and any traffic that is routed via the VTI is automatically encrypted. This means that a single IPsec SA will be negotiated with the device at the other end of the tunnel and that all traffic being sent down this tunnel will be encrypted by this SA.

Specific traffic selectors for different remote address ranges

There are circumstances in which it may be desirable to be selective about which traffic trying to go into the tunnel is accepted and encrypted. This means it may be necessary to create multiple SAs within the tunnel, so that different streams of traffic within the tunnel are encrypted by different SAs.

The latter case is necessitated by connections with some legacy devices that may not support route based VPNs. It may instead attempt to negotiate the use of IP address traffic selectors to match, filter, and transport only a specific range of local and remote IP addresses in each SA.

To deal with these requirements, AlliedWare Plus VTI tunnel interfaces can be configured to negotiate one or more pairs of local and remote network traffic selectors. This enables the negotiation of different SAs for different streams of traffic. When using IKEv1 a single IPsec SA is created for each negotiated pair. With IKEv2 multiple pairs of traffic selectors can be negotiated on a single IPsec SA.

Main mode or Aggressive mode

Main mode or Aggressive mode selection only applies to legacy devices using IKEv1, not IKEv2. In Main mode, the Phase 1 parameters are exchanged within multiple message exchanges, containing encrypted authentication information. This ensures maximum security, at the cost of multiple message exchanges to fully negotiate the security association.

In Aggressive mode, the Phase 1 parameters are instead exchanged within a single message, including with **unencrypted** authentication information, such as the IDs. This is at the cost of slightly lower security—because additional information, such as hostname IDs are shared to the peer in unencrypted clear text. The advantage, however, is that this mode reduces negotiation of the IKE security association to three messages, allowing for faster formation of the encrypted VPN.

Therefore, if one end of the VPN link is assigned a dynamic IP address, and VPNs are matched on hostname ID <***fqdn***> instead of a statically-configured peer IP address, then it is appropriate to ensure the peer is configured to use Aggressive mode instead of Main mode, if using IKEv1.

This is necessary especially when there are multiple remote sites within a hub-and-spoke VPN topology whose WAN IP addresses are all dynamically assigned. Using Aggressive mode ensures that the peer supplies the ISAKMP ID (hostname) within the initial (unencrypted) offer to the peer, to allow the peer to be identified and authenticated. AlliedWare Plus firewalls and routers use IKEv2 as their default profile, and will not respond to incoming IKEv1 requests. If a peer device does not support IKEv2, then in order to configure the device to support IKEv1, you need to configure a custom ISAKMP profile with those specific options selected. You can use either the Main mode or Aggressive mode option as required to ensure interoperability with legacy devices.

Step-by-step configuration

This section describes how to configure IPsec.

Basic IPsec protection

The configuration steps to enable IPsec protection are:

- Configure the pre-shared key for ISAKMP and associate the key with a peer address
- Set up the tunnel to which IPsec protection will be applied
- Apply IPsec protection to the traffic in the tunnel
- Configure one or more routes to the IP subnets on the network at the far end of the tunnel

Follow these steps to enable IPsec protection for traffic:

Step 1: **Configure the pre-shared key for ISAKMP**

```
awplus# configure terminal
awplus(config)# crypto isakmp key <key>  {hostname <host-name>|address
{<ipv4-addr>|<ipv6-addr>}|policy <policy-name>} type [eap|psk]
```

Enter the pre-shared key and peer IP address. The key is associated with:

- hostname or UFQDN
- ipv4-addr (destination IPv4 address, format A.B.C.D)
- ipv6-addr (destination IPv6 address, format X:X::X:X)
- policy name (the local policy name)
- type isakmp key type is optional.

Step 2: **Set up the tunnel to apply IPsec protection**

- Enter Interface mode and specify a tunnel name (e.g. tunnel1) and IP address for the tunnel interface.

```
awplus(config)# interface tunnel <0-65535>
awplus(config-if)# ip address <ip-address>
```

- Enter the name of the interface whose IP address is used as the source IP for traffic in the tunnel. The tunnel source can also be an IP address on the device.

```
awplus(config-if)# ip address <interface-name>
```

- Enter the IP address for the peer tunnel destination.

```
awplus(config-if)# tunnel destination <ip-address>
```

Enter the tunnel mode, where mode can be one of:

- IPsec IPv4, IPsec IPv6, L2TP v3, L2TP v3 IPv6, GRE, GRE IPv6

```
awplus(config-if)# tunnel mode <mode>
```

Step 3: **Apply IPsec protection to traffic in the tunnel**

```
awplus(config-if)# tunnel protection ipsec
```

- Exit Configuration mode:

```
awplus(config-if)# exit
```

Step 4: **Configure routes to the IP subnets at the receiving end of the tunnel**

```
awplus(config)# ip route <far-end-subnet> <tunnel-name>
```

How to use custom profiles

The configuration tasks to use custom profiles are:

- Define and name profiles
- Set up Global parameters (optional)
- Add transforms to each profile
- Associate an ISAKMP profile with one or more peers
- Apply an IPsec profile to a tunnel

ISAKMP profiles

Follow these steps to configure your custom profiles for ISAKMP:

Step 1: **Define and name the profiles**

```
awplus# configure terminal
```

- Enter the custom ISAKMP profile name. Profile names are case insensitive and can be up to 64 characters long composed of printable ASCII characters. Profile names can have only letters from a to z and A to Z, numbers from 0 to 9, - (dash), or _ (underscore). After you have entered this command, you will be in Profile Configuration mode.

```
awplus(config)# crypto isakmp profile <profile-name>
```

Step 2: **Set up Global parameters (optional)**

- Enter the lifetime in seconds.This is optional and the default is 86400 seconds (24 hours). Lifetime measures how long the IPsec SA can be maintained before it expires. Lifetime prevents a connection from being used too long.

```
awplus(config-isakmp-profile)# lifetime <lifetime>
```

To set the ISAKMP protocol version specify the version and mode:

- version 1 (IKEv1) or version 2 (IKEv2). This is optional and the default is version 2.
- mode aggressive or mode main.

```
awplus(config-isakmp-profile)# version {1 mode {aggresssive|main}|2}
```

- Enter the DPD interval in seconds. The default is 30 seconds.
- DPD (Dead Peer Detection) is an IKE mechanism using a form of keep-alive to determine if a tunnel peer is still active.
- The interval parameter specifies the amount of time the device waits for traffic from its peer before sending a DPD acknowledgment message.

```
awplus(config-isakmp-profile)# dpd-timeout <wait-time>
```

Step 3: **Add transforms to each profile**

- Specify the following:
  - transform priority (1 is the highest)
  - integrity (Secure Hash Standard)
  - encryption (Advanced Encryption Standard or 3DES)
  - Diffie-Hellman group

```
awplus(config-isakmp-profile)# transform <1-255> integrity [sha1|sha256|
sha512] encryption [3des|aes128|aes192|aes256] group [2|5|14|15|16|18]
```

Step 4: **Associate with a peer**

- Enter Global Configuration mode.

```
awplus# configure terminal
```

Associate your ISAKMP custom profile with a peer.

Enter the following:

- dynamic (remote endpoint with a dynamic IP address
- ipv4-addr (destination IPv4 address, format A.B.C.D)
- ipv6-addr (destination IPv6 address, format X:X::X:X)
- hostname (remote endpoint with a host name as the destination)
- policy name (the local policy name)
- profile name.

```
awplus(config)# crypto isakmp peer {dynamic|address}
{<ipv4-addr>|<ipv6-addr>}|hostname <hostname>|policy <policy-name>}
profile <profile_name>
```

IPsec profiles

Follow these steps to configure your custom profiles for IPsec:

Step 1: Define and name the profiles

```
awplus# configure terminal
```

- Enter the custom IPsec profile names. Profile names are case insensitive and can be up to 64 characters long composed of printable ASCII characters. Profile names can have only letters from a to z and A to Z, numbers from 0 to 9, - (dash), or _ (underscore). After you have entered this command, you will be in Profile Configuration mode.

```
awplus(config)# crypto ipsec profile <profile-name>
```

Step 2: Set up Global parameters (optional)

- Enter the lifetime in seconds.This is optional and the default is 28800 seconds (8 hours).

```
awplus(config-ipsec-profile)# lifetime seconds <lifetime>
```

- Enter the PFS (Perfect Forward Security) number. The numbers represent the Diffie-Hellman group. PFS is disabled by default.

```
awplus(config-ipsec-profile)# pfs <2|5|14|16|18>
```

Step 3: Add transforms

- Specify the following:
  - transform priority (1 is the highest)
  - protocol (which has only ESP as an option)
  - integrity (Secure Hash Standard)
  - encryption (Advanced Encryption Standard or 3DES or GCM).

Note: See "Default IPsec profiles" on page 7 for more information.

```
awplus(config-ipsec-profile)# transform <1-255> protocol esp integrity
[sha1|sha256|sha512] encryption [3des|aes128|aes192|aes256]
```

or

```
awplus(config-ipsec-profile)# transform <1-255> protocol esp integrity
[gcm8|gcm12|gcm16] encryption [aes128|aes192|aes256]
```

Step 4: Associate with a tunnel

- Enter Interface mode and specify a tunnel name. For example, tunnel1.

```
awplus(config)# interface tunnel <0-65535>
```

- Enter your custom profile name. By default IPsec protection for packets encapsulated by tunnel is disabled.

```
awplus(config-if)# tunnel protection ipsec {profile <profile-name>}
```

How to use traffic selectors

For the tunnel being protected, use Interface mode to enter the commands for selecting the traffic to be associated with different IPsec SAs. There are separate commands to match the local source address and the remote destination address of the packets.

Selectors operate in pairs – one matching the source address and one matching the destination address. ID numbers indicate which selectors are paired with each other. For example, a **local** and **remote** selector that both have the same ID are a pair.

Use the following commands to configure your traffic selectors:

Step 1: Enter Global Configuration mode

```
awplus# configure terminal
```

Step 2: Enter Interface mode and specify a tunnel name. For example tunnel1

```
awplus(config)# interface tunnel <0-65535>
```

- Enter the local address range for this selector pair ID. The local and remote selectors must use the same ID. This identifies the range of source addresses on outgoing traffic (or destination addresses on incoming traffic) to which the selector applies.

```
awplus(config-if)# tunnel local selector {id-number} <address-range>
```

- Enter the remote address range for this selector pair ID. This must have the same ID of the local selector. This identifies the range of destination addresses on outgoing traffic (or source addresses on incoming traffic) to which the selector applies.

```
awplus(config-if)# tunnel remote selector {id-number} <address-range>
```

Strict pairing for IPsec tunnels

By default, if you specify multiple address selector pairs, the tunnel can permit any combination of matching sources and/or destinations. While this conforms to the RFC, it may not be the expected behavior and may cause the IPsec SA to either fail negotiation or fail to pass traffic correctly.

Version 5.4.8-1.1 adds a new optional command **tunnel selector paired**. This command forces ISAKMP to create individual Phase 2 IPsec SAs for each pair of source and destination selectors that have the same selector ID. Only traffic that matches a selector pair is permitted to flow via the associated SA.

For example, when creating a tunnel between 172.16.1.0/24 and 172.16.2.0/24, and also between 172.16.1.0/24 and any other destination, you can use the following tunnel selector commands:

```
awplus(config)# interface tunnel0
awplus(config-if)# tunnel local selector 2 172.16.1.0/24
awplus(config-if)# tunnel remote selector 2 172.16.2.0/24
awplus(config-if)# tunnel local selector 3 172.16.1.0/24
awplus(config-if)# tunnel remote selector 3 0.0.0.0/0
awplus(config-if)# tunnel selector paired
```

How to identify a peer by name rather than IP address

When a peer is dynamically allocated an IP address, it is not possible to know its address in advance. So, when a connection comes in from the peer, the recipient of the connection needs some way to identify who the connection came from. This is done by using a local tunnel name that is embedded in the packets, that initiates the connection.

The commands to do this are entered in Interface mode for the tunnel being protected.

Use these commands to configure a **local** tunnel name for the peer:

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter Interface mode and specify a tunnel interface index identifier (from 0-65535). By default no tunnel interfaces exist. For example tunnel1.

```
awplus(config)# interface tunnel <0-65535>
```

- Enter the local tunnel name that is sent in IPsec setup packets.

```
awplus(config-if)# tunnel local name <local-name>
```

A peer receiving the connection configures a **remote** name, to identify the name it expects to see in connections from the remote peer:

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter Interface mode and specify a tunnel interface index identifier (from 0-65535). By default no tunnel interfaces exist.

```
awplus(config)# interface tunnel <0-65535>
```

- Enter the local tunnel name that is sent in IPsec setup packets.

```
awplus(config-if)# tunnel remote name <name-expected-to-be-received-in-ipsec-connections>
```

Configuration examples

Example 1: IPsec tunnel between two AlliedWare Plus firewalls

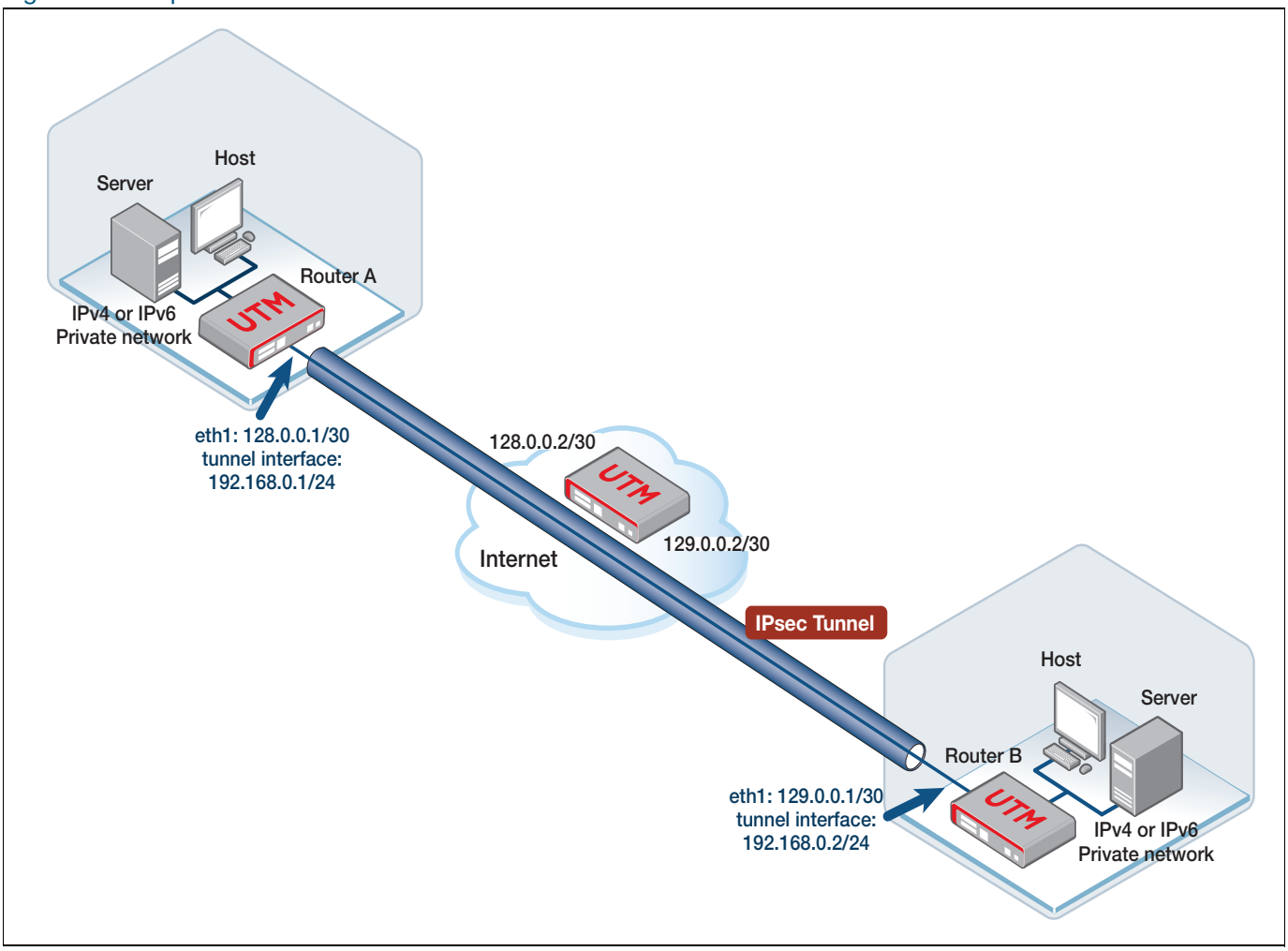
This example provides step-by-step instructions for configuring an IPsec tunnel between two AlliedWare Plus firewalls. It assumes that IP has been correctly configured and is operational on both devices. The following table lists the parameter values in the example:

Note: Public IP addresses are used in this example.

Table 2: IP address allocation

	DEVICE A	DEVICE B
IP address of Ethernet interface eth1	128.0.0.1/30	129.0.0.1/30
tunnel source IP address	128.0.0.1/30	129.0.0.1/30
tunnel destination IP address	129.0.0.1/30	128.0.0.1/30
IP address of tunnel interface	192.168.0.1/24	192.168.0.2/24

Figure 1: Example for an IPsec tunnel between two AlliedWare Plus firewalls



Configuring an IPsec tunnel between two AlliedWare Plus firewalls

Step 1: **Configure Device A**

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter Interface Configuration Mode

```
awplus(config)# interface eth1
```

- Assign an IP address for interface eth1

```
awplus(config-if)# ip address 128.0.0.1/30
```

- Exit Interface Configuration mode and enter Global Configuration mode.

```
awplus(config-if)# exit
```

- Create a virtual tunnel called tunnel1

```
awplus(config)# interface tunnel1
```

- Assign an IP address to tunnel1

```
awplus(config-if)# ip address 192.168.0.1/24
```

- Designate the interface or IP address that will be used as the source IP of the tunnel.

```
awplus(config-if)# tunnel source eth1
```

- Designate the tunnel destination address, which is the IP address of interface eth1 on Device B.

```
awplus(config-if)# tunnel destination 129.0.0.1
```

- Specify the tunnel mode

```
awplus(config-if)# tunnel mode ipsec ipv4
```

- To securely route packets through the tunnel, you need to use the **tunnel protection ipsec** command to encrypt and authenticate its packets. This is required for IPsec mode tunnels. It is optional for other tunnel modes.

```
awplus(config-if)# tunnel protection ipsec
```

Step 2: **Configure Device B**

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter Interface Configuration Mode

```
awplus(config)# interface eth1
```

- Assign an IP address for interface eth1

```
awplus(config-if)# ip address 129.0.0.1/30
```

- Exit Interface Configuration mode and enter Global Configuration mode.

```
awplus(config-if)# exit
```

- Create a virtual tunnel called tunnel1

```
awplus(config)# interface tunnel1
```

- Assign an IP address to tunnel1

```
awplus(config-if)# ip address 192.168.0.2/24
```

- Designate the interface whose IP address that will be used as the source IP of the tunnel.

```
awplus(config-if)# tunnel source eth1
```

- Designate the tunnel destination address, which is the IP address of interface eth1 on Device A.

```
awplus(config-if)# tunnel destination 128.0.0.1
```

- Specify the tunnel mode

```
awplus(config-if)# tunnel mode ipsec ipv4
```

- To securely route packets through the tunnel, you need to use the **tunnel protection ipsec** command to encrypt and authenticate its packets.

```
awplus(config-if)# tunnel protection ipsec
```

Step 3: **Configure authentication key on Device A**

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter the tunnel key tunnelkey.

```
awplus(config)# crypto isakmp key tunnelkey address 129.0.0.1
```

Step 4: **Configure authentication key on Device B**

- Enter Global Configuration mode.

```
awplus# configure terminal
```

- Enter the tunnel key tunnelkey.

```
awplus(config)# crypto isakmp key tunnelkey address 128.0.0.1
```

Step 5: **Verify the configuration**

- You can use the ping command to verify that the tunnel is established. Log into Device A and ping the interface IP address of Device B.

```
awplus# ping 192.168.0.2
```

Note: Be aware that at least one echo request will not succeed because it is dropped. Whether any other echo requests are dropped depends on how quickly ISAKMP finishes the negotiation and the ISAKMP and IPsec SAs are set. Normal ping, with a one second delay between echo requests, is expected to have the next four echo requests all responded to.

Example ping from the console

```
awplus#ping 192.168.0.2
PING 192.168.0.2 (192.168.0.2) 56(84) bytes of data.
From 192.168.0.1 icmp_seq=1 Destination Host Unreachable
64 bytes from 192.168.0.2: icmp_req=2 ttl=64 time=0.590 ms
64 bytes from 192.168.0.2: icmp_req=3 ttl=64 time=0.462 ms
64 bytes from 192.168.0.2: icmp_req=4 ttl=64 time=0.452 ms
64 bytes from 192.168.0.2: icmp_req=5 ttl=64 time=0.452 ms
```

Example 2: ISAKMP and IPsec profiles

This example shows how to configure a named IPsec profile and a named ISAKMP profile in a single device.

The named IPsec profile is configured to use weaker cryptographic algorithms (AES128, 3DES), SHA1 and non-default SA lifetimes. The named ISAKMP profile is configured to use aggressive mode IKEv1 and DH group 2. VLAN1 interface is private. Eth1 interface is public.

Example configuration for ISAKMP and IPsec custom profiles

```
!
crypto ipsec profile remote-office-phase2
 lifetime seconds 3600
 transform 1 protocol esp integrity SHA1 encryption AES128
 transform 2 protocol esp integrity SHA1 encryption 3DES
!
crypto isakmp profile remote-office-phase1
 version 1 mode aggressive
 transform 1 integrity SHA1 encryption AES128 group 2
 transform 2 integrity SHA1 encryption 3DES group 2
 lifetime 10800
!
crypto isakmp key SAMPLEKEY address 16.1.0.2
!
crypto isakmp peer address 16.1.0.2 profile remote-office-phase1
!
interface eth1
 ip address 16.0.0.1/30
!
interface vlan1
 ip address 192.168.1.0/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination 16.1.0.2
 tunnel protection ipsec profile remote-office-phase2
 tunnel mode ipsec ipv4
 ip address 192.168.3.1/30
!
 ip route 192.168.2.0/24 tunnel1
!
```

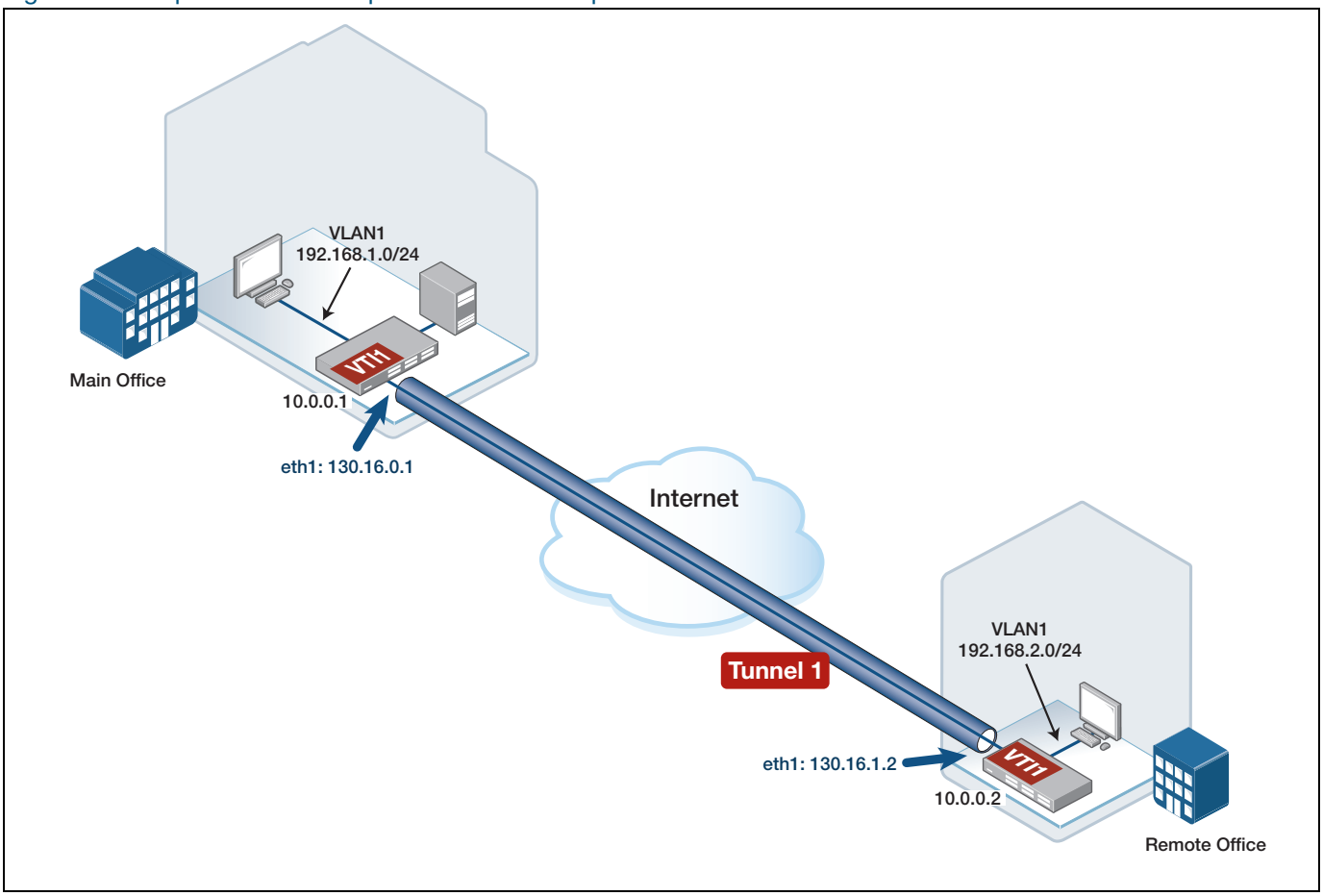
Example 3: A custom profile with a PFS option

This example shows how to configure a custom profile that sets Main mode IKEv1 in the ISAKMP configuration as well as Perfect Forward Secrecy (PFS) Diffie-Hellman (DH) group 5.

The PFS group option ensures a new Diffie-Hellman key exchange occurs whenever an SA is re-negotiated (for example, when an SA lifetime expires) to offer an addition layer of protection in the case where a private key has been compromised. Perfect Forward Secrecy (PFS) ensures generated keys (e.g. IPsec SA keys) are not compromised if any other keys (e.g. ISAKMP SA keys) are compromised. This comes at the cost of additional processing overhead, so most vendors disable this option by default. Similarly, this option is not enabled in the AlliedWare Plus default profile.

Therefore, if you wish to use PFS, you do need to configure a custom profile that has PFS enabled.

Figure 2: Example for a custom profile with a PFS option



Example **Main Office** configuration for a custom profile with a PFS option

```
!
crypto ipsec profile phase2
 transform 1 protocol esp integrity SHA256 encryption AES256
 pfs 5
!
crypto isakmp profile phase1
 transform 1 integrity SHA256 encryption AES256 group 5
 version 1 mode main
!
crypto isakmp key SAMPLEKEY address 130.16.1.2
!
crypto isakmp peer address 130.16.1.2 profile phase1
!
interface vlan1
 ip address 192.168.1.254/24
!
interface eth1
 ip address 130.16.0.1/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination 130.16.1.2
 tunnel protection ipsec profile phase2
 tunnel mode ipsec ipv4
 ip address 10.0.0.1/30
!
 ip route 192.168.2.0/24 tunnel1
!
```

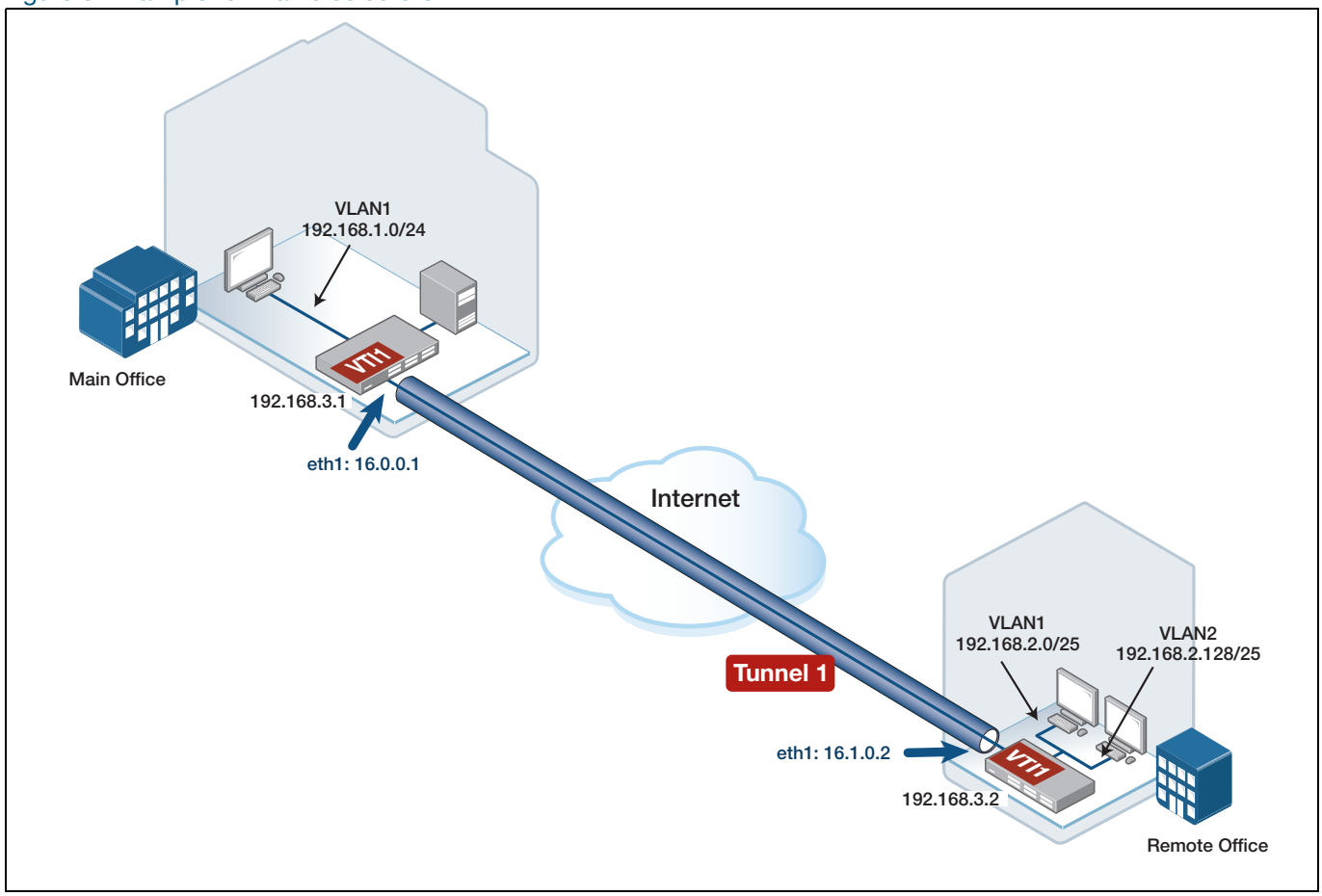
Example 4: Traffic selectors

In this configuration example, the network at the remote end of the tunnel has multiple non-contiguous network address ranges. The legacy VPN gateway at the remote site is configured with multiple traffic selectors. These traffic selectors are configured to match traffic to or from each of the individual subnet address ranges at that site.

An IPsec SA is formed for each individual set of local and remote traffic selectors that are configured.

Traffic routed over the VTI, that does not also match the optional local and remote address selectors, is discarded. Only traffic that matches one of the traffic selectors is permitted through the associated SA:

Figure 3: Example for Traffic selectors



In the configuration you can see that:

- Selector 10 matches traffic between 192.168.1.0/24 and 192.168.2.0/26
- Selector 20 matches traffic between 192.168.1.0/24 and 192.168.2.128/26

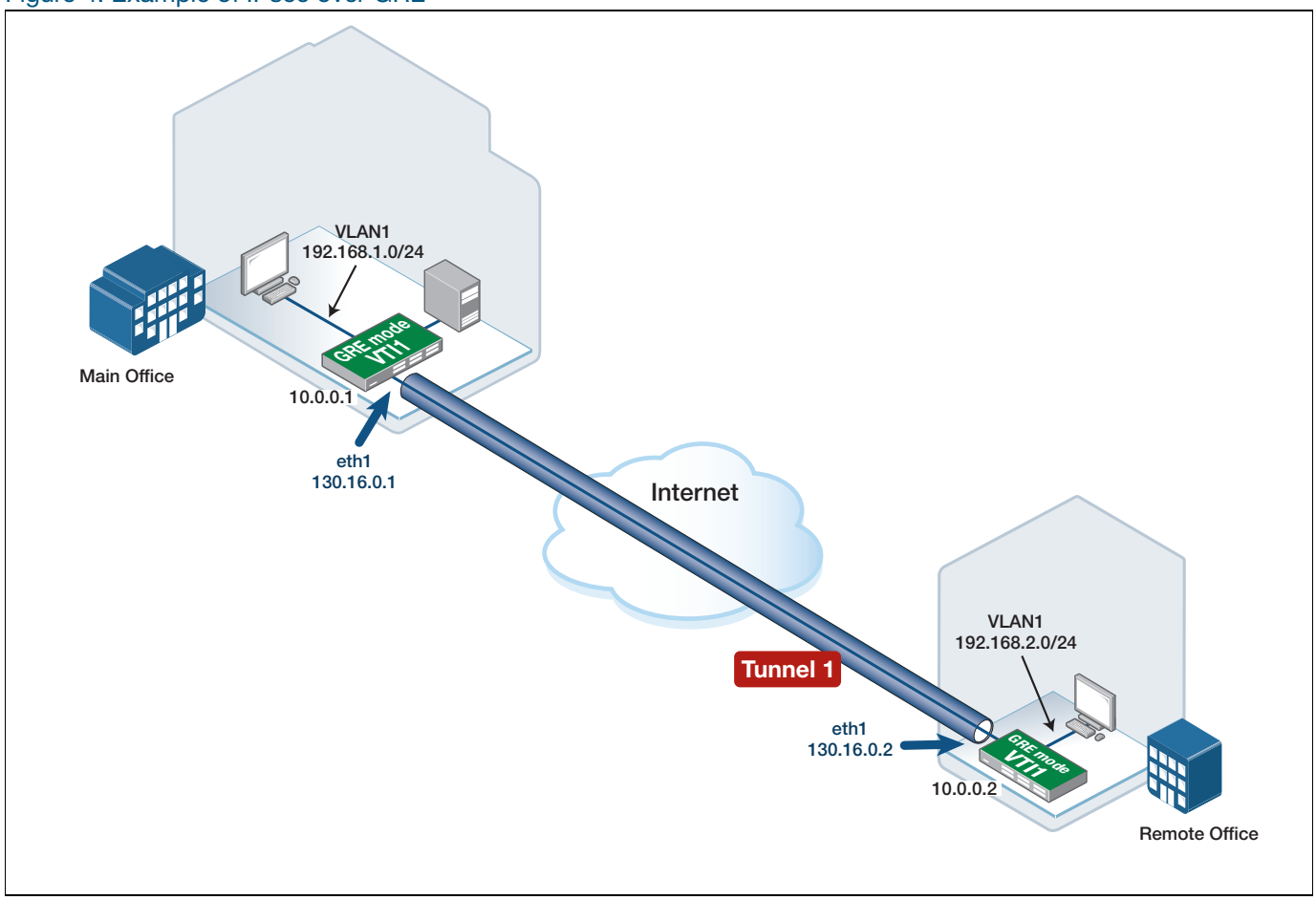
Example **Main Office** configuration for traffic selectors

```
!
crypto ipsec profile remote-office-phase2
 lifetime seconds 28800
 transform 1 protocol esp integrity SHA1 encryption AES128
 transform 2 protocol esp integrity SHA1 encryption 3DES
!
crypto isakmp profile remote-office-phase1
 version 1 mode aggressive
 transform 1 integrity SHA1 encryption AES128 group 2
 transform 2 integrity SHA1 encryption 3DES group 2
 lifetime 86400
!
crypto isakmp key SAMPLEKEY address 16.1.0.2
!
crypto isakmp peer address 16.1.0.2 profile remote-office-phase1
!
interface eth1
 ip address 16.0.0.1/30
!
interface vlan1
 ip address 192.168.1.0/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination 16.1.0.2
 tunnel local selector 10 192.168.1.0/24
 tunnel remote selector 10 192.168.2.0/26
 tunnel local selector 20 192.168.1.0/24
 tunnel remote selector 20 192.168.2.128/26
 tunnel selector paired
 tunnel protection ipsec profile remote-office-phase2
 tunnel mode ipsec ipv4
 ip address 192.168.3.1/30
!
 ip route 192.168.2.0/26 tunnel1
 ip route 192.168.2.128/26 tunnel1
!
```

Example 5: IPsec over GRE

AlliedWare Plus firewalls and routers can use IPsec VPNs to protect GRE tunnels. This example shows how to configure a Layer 3 (GRE) VPN tunnel that is protected by custom ISAKMP and IPsec profiles using IKEv2 SHA256, AES256 encryption, and Diffie Hellman (DH) group 15.

Figure 4: Example of IPsec over GRE



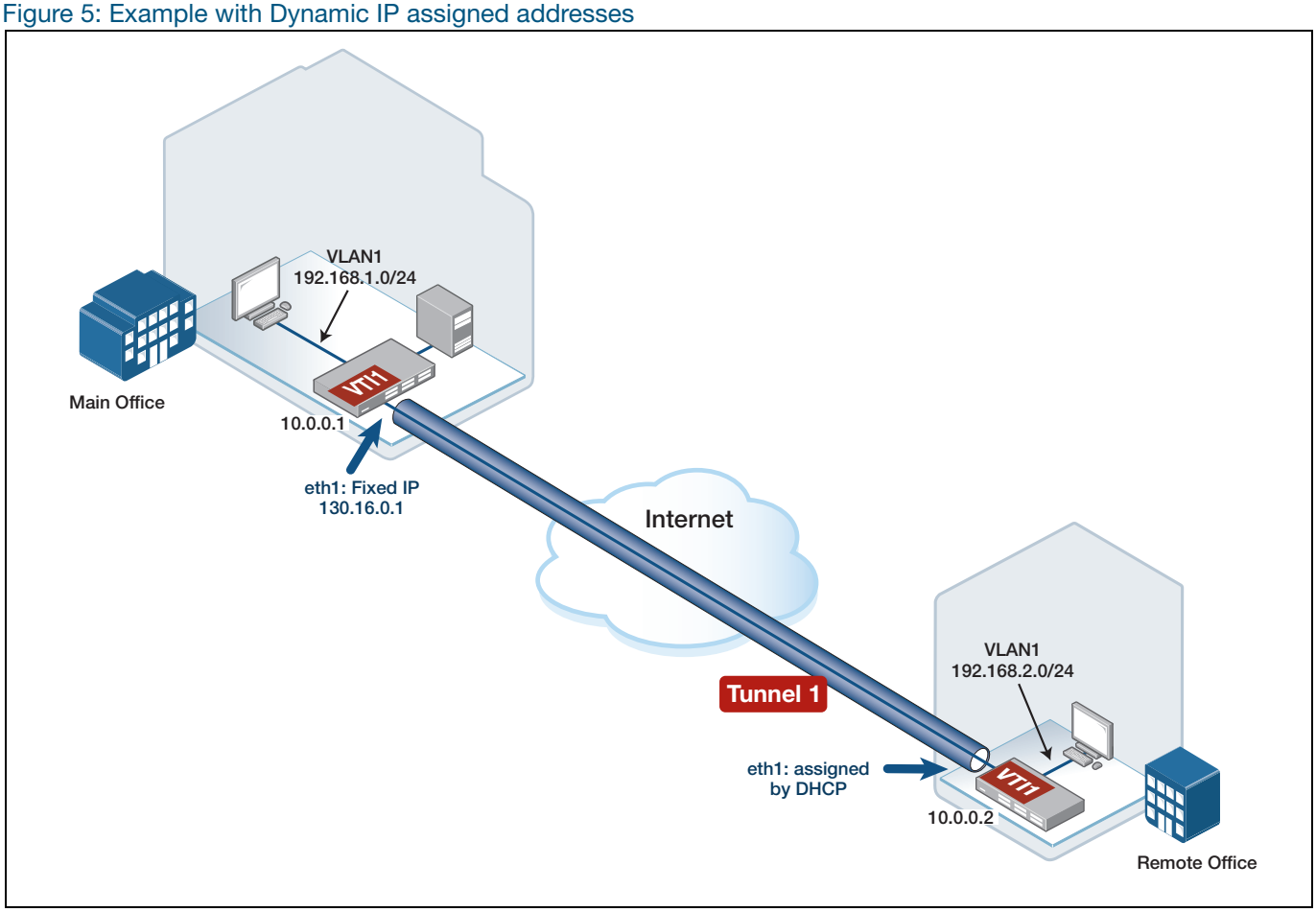
Example **Main Office** configuration for IPsec over GRE

```
crypto ipsec profile phase2
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile phase1
 transform 1 integrity SHA256 encryption AES256 group 14
 version 2
!
crypto isakmp key remote address 130.16.0.2
!
crypto isakmp peer address 130.16.0.2 profile phase1
!
interface vlan1
 ip address 192.168.1.254/24
!
interface eth1
 ip address 130.16.0.1/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination 130.16.0.2
 tunnel protection ipsec profile phase2
 tunnel mode gre
 ip address 10.0.0.1/30
!
 ip route 192.168.2.0/24 tunnel1
```

Example 6: Dynamically assigned IP addresses

As discussed previously (see,"Working with dynamically assigned IP addresses" on page 10) the IPsec configuration needs to use **local** and **remote** names to identify the connecting peer if the peer has a dynamically assigned IP address.

Figure 5: Example with Dynamic IP assigned addresses



In this configuration, the remote office WAN interface address is dynamically allocated via DHCP.

Therefore, the remote office VTI is configured to supply the text string **Remote_Site_1** as its local identifier, which allows the main office to match and identify the incoming VPN traffic from the remote office.

Similarly, the main office VTI is configured with the command **tunnel destination dynamic**, and the pre-shared crypto ISAKMP key is matched, based on the local hostname identifier text string **Remote_Site_1** (supplied by the remote office).

This example also uses custom IPsec profiles, although there is no requirement to use custom profiles when remote site WAN addresses are dynamically allocated. The custom profile is used purely to show how to configure an alternative set of non-default crypto options, such as IKEv2, DH group 5, and PFS.

Example **Main Office** configuration with dynamically assigned addresses

```
!
crypto ipsec profile phase2
 pfs 5
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile phase1
 version 2
 transform 1 integrity SHA256 encryption AES256 group 5
!
crypto isakmp key SAMPLEKEY hostname Remote_Site_1
!
crypto isakmp peer dynamic profile phase1
!
interface eth1
 ip address 130.16.0.1/24
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination dynamic
 tunnel remote name Remote_Site_1
 tunnel protection ipsec profile phase2
 tunnel mode ipsec ipv4
 ip address 10.0.0.1/30
!
 ip route 192.168.2.0/24 tunnel1
!
```

Example **Remote Office** configuration with dynamically assigned addresses

```
crypto ipsec profile phase2
 pfs 5
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile phase1
 version 2
 transform 1 integrity SHA256 encryption AES256 group 5
!
crypto isakmp key SAMPLEKEY address 130.16.0.1
!
crypto isakmp peer address 130.16.0.1 profile phase1
!
interface eth1
 ip address dhcp
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel1
 tunnel source eth1
 tunnel destination 130.16.0.1
 tunnel local name Remote_Site_1
 tunnel protection ipsec profile phase2
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
 ip route 192.168.1.0/24 tunnel1
```

Example 7: IPsec with NAT-Traversal

When peers form a secure VPN, firstly an ISAKMP SA is negotiated to provide a framework for secure key exchange using IKE. Typically ISAKMP uses UDP port 500 to transport the data between the two peers.

Subsequently, an IPsec SA is formed between the two peers, this time using ESP as the mechanism to protect the private inter-office IP data streams. The entire IP datagram headers of the private data streams are encrypted and encapsulated inside the ESP headers. This includes the private source IP/destination IP, and TCP/UDP port numbers.

ESP uses IP protocol number 50, and does not contain additional information, such as source and destination port numbers used commonly by other IP protocols, such as TCP or UDP.

This lack of port numbers makes it difficult to pass ESP data through any intermediate devices performing Network Address Port Translation (NAPT), on the path between the VPN peers. This is because the intermediate NAPT devices may not support session tracking based on alternative fields that are contained in the ESP datagram VPN headers, such as the SPI (Security Parameter Index).

To resolve this issue, the NAT-Traversal (NAT-T) option can be negotiated between the two VPN peers. The first step in NAT-T negotiation occurs during the initial ISAKMP SA negotiation, whereby the two peers automatically detect that each other supports the NAT-T option. If both devices support NAT-T, then the two peers perform NAT-Discovery (NAT-D) to detect the presence (or not) of an intermediate device performing IP address and/or port translation.

NAT-D works by each peer internally calculating a unique hash value based on the source and destination IP, and port numbers used for IKE. NAT-D messages are then sent between the two peers containing the unique hash value payload. Each peer extracts the hash values from the received NAT-D messages, and compares them to the hash values that were previously calculated. If the internally calculated and received HASH values differ, then the two peers know there is an intermediate device performing some form of network address and/or port translation. If the two compared hash values are the same, then the two peers know that there is no intermediate device performing IP address and/or port translation.

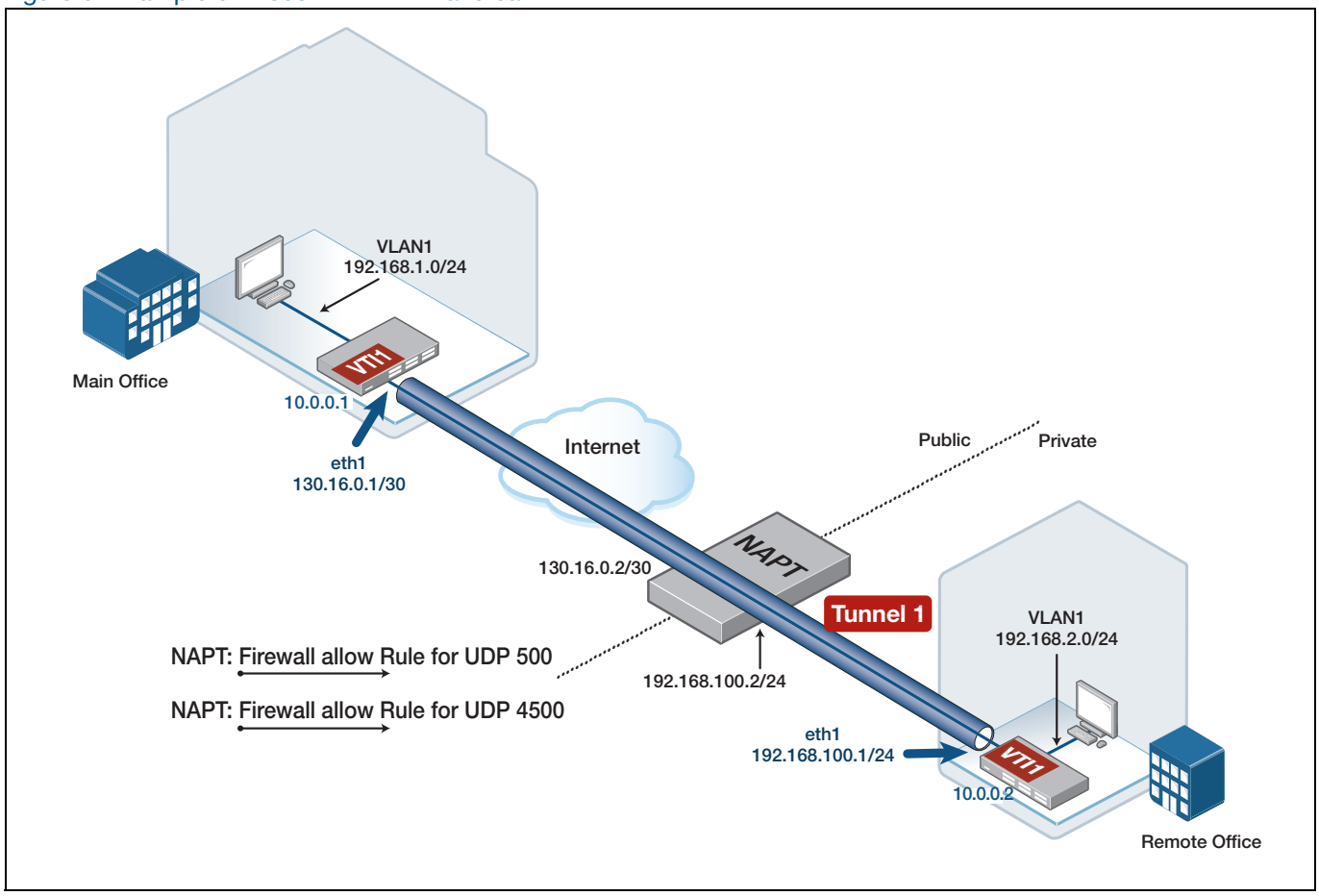
If an intermediate NAPT device is detected, NAT-T will change the UDP port numbers used by ISAKMP from UDP 500, to become UDP 4500, for all subsequent ISAKMP communications, and the ISAKMP SA is then formed.

The ESP messages used by IPsec are also encapsulated inside NAT-T, allowing them to pass seamlessly through the intermediate NAPT device as part of the existing UDP 4500 session.

The ESP traffic is only encapsulated inside NAT-T (UDP 4500) if the two NAT-T capable peers detect the presence of an intermediate NAPT device. Otherwise ISAKMP will continue to use default UDP 500, and IPsec will continue to use IP protocol 50.

The intermediate NAPT device will need to be configured with rules to allow UDP 500/UDP 4500 to pass through.

Figure 6: Example of IPsec with NAT-Traversal



In this configuration, the remote office WAN uses a private IP address, as the remote office router is located on the private side of the intermediate NAPT device. Traffic originating from the remote office has its source IP (and source port) translated by the intermediate NAPT device as the data is routed to the Internet. VPN traffic arriving at the main office therefore appears to have originated from the public Internet IP address of the NAPT device, not the private IP address of the remote office WAN.

The main office VPN tunnel destination IP configured on the peer is therefore the public IP address of the NAPT device, not the remote office eth1 private WAN IP address.

The intermediate NAPT device needs to be configured with firewall NAT port forwarding rules for ISAKMP traffic (UDP port 500), and NAT-T traffic (UDP port 4500) to allow the VPN traffic arriving from the main office to be forwarded to the private side WAN IP address of the remote office (eth1).

In this configuration, the remote office VTI is configured to point to the WAN IP address of the main office directly.

The main office VTI is configured to supply the text string **office1** as its local identifier, which allows the remote office to match and identify the incoming VPN traffic from the main office.

Similarly, the remote office VTI is configured to supply the text string **office2** as its local identifier, which allows the main office to match and identify the incoming VPN traffic from the remote office.

The main office and remote office pre-shared crypto ISAKMP keys are matched based on the local hostname identifier configured in each remote peer (instead of peer IP address).

Example **Main Office** configuration with NAT-T

```
!
crypto isakmp key SAMPLEKEY hostname office2
!
interface eth1
 ip address 130.16.0.1/30
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel1
 tunnel source 130.16.0.1
 tunnel destination 130.16.0.2
 tunnel local name office1
 tunnel remote name office2
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
!
```

Example **Remote Office** configuration with NAT-T

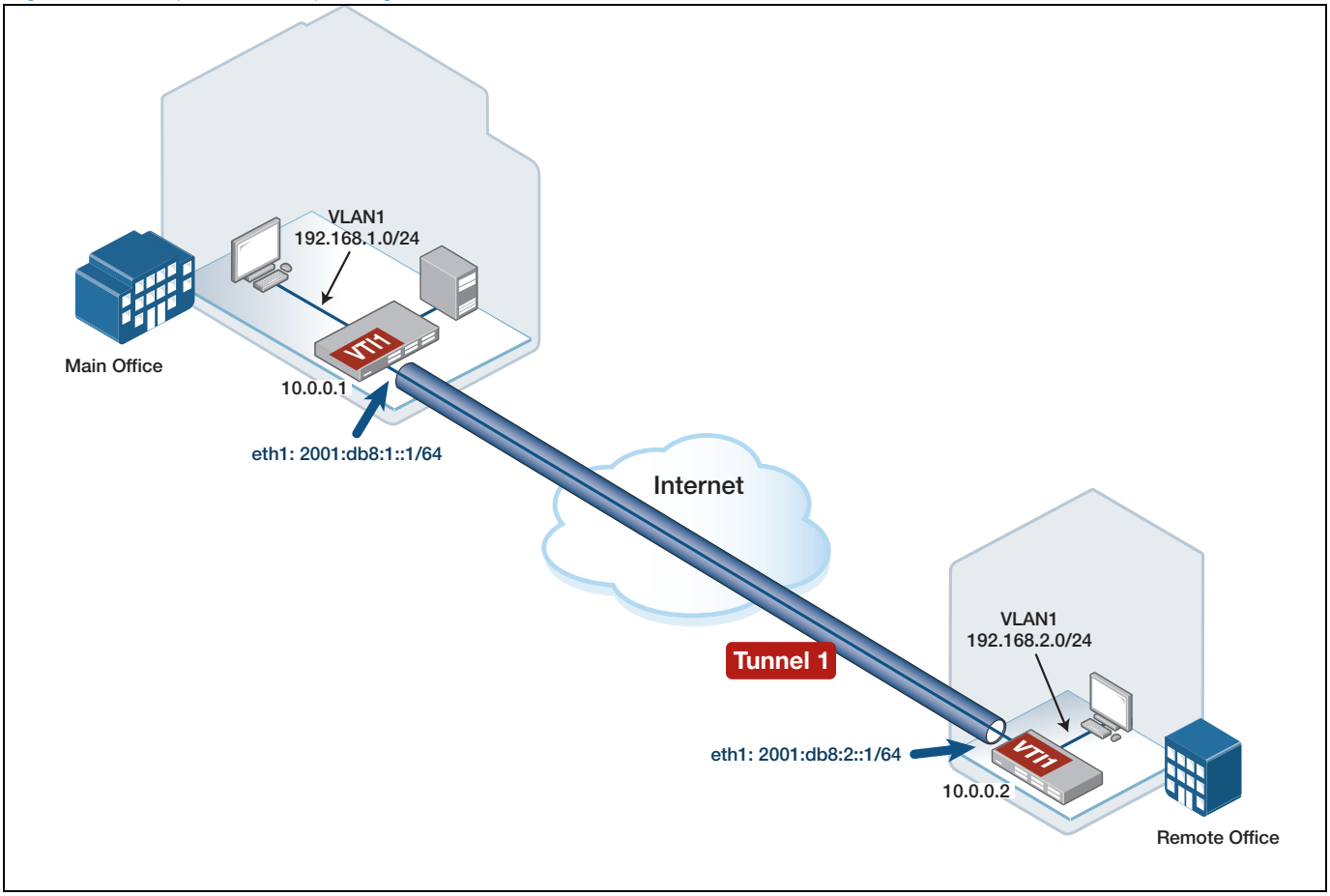
```
!
crypto isakmp key SAMPLEKEY hostname office1
!
interface eth1
 ip address 192.168.100.1/24
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel1
 tunnel source 192.168.100.1
 tunnel destination 130.16.0.1
 tunnel local name office2
 tunnel remote name office1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
ip route 192.168.1.0/24 10.0.0.1
ip route 130.16.0.0/30 192.168.100.2
!
```

Example 8: IPv4 over IPv6 tunnel

The following example shows how to configure an IPv6 VPN between the Main Office and Remote Office. The eth WAN interfaces are configured with IPv6 addresses. The tunnel and VLAN interfaces are configured with IPv4 addresses. In this example, IPv4 traffic is encapsulated and transported within the IPv6 VPN.

It is possible to optionally configure IPv4 traffic selectors on IPsec IPv6 tunnels (from version 5.4.9-0.1 onwards). In this example tunnel selectors are configured to match IPv4 traffic to be encrypted and transported via the IPv6 IPsec VPN. The default selectors for this tunnel type will only match IPv6 traffic if selectors are not configured.

Figure 7: Example for transporting IPv4 traffic over an IPv6 tunnel



Example **Remote Office** configuration for ipv4 over ipv6 IPsec tunnel

```
!
crypto isakmp key SAMPLEKEY address 2001:db8:1::1
!
interface eth1
 ipv6 address 2001:db8:2::1/64
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel1
 description VPN_to_Office
 tunnel source eth1
 tunnel destination 2001:db8:1::1
 tunnel local selector 1 192.168.2.0/24
 tunnel remote selector 1 192.168.1.0/24
 tunnel protection ipsec
 tunnel mode ipsec ipv6
 ip address 10.0.0.2/30
!
ip route 192.168.1.0/24 10.0.0.1
!
ipv6 route ::/0 2001:db8:2::2
!
```

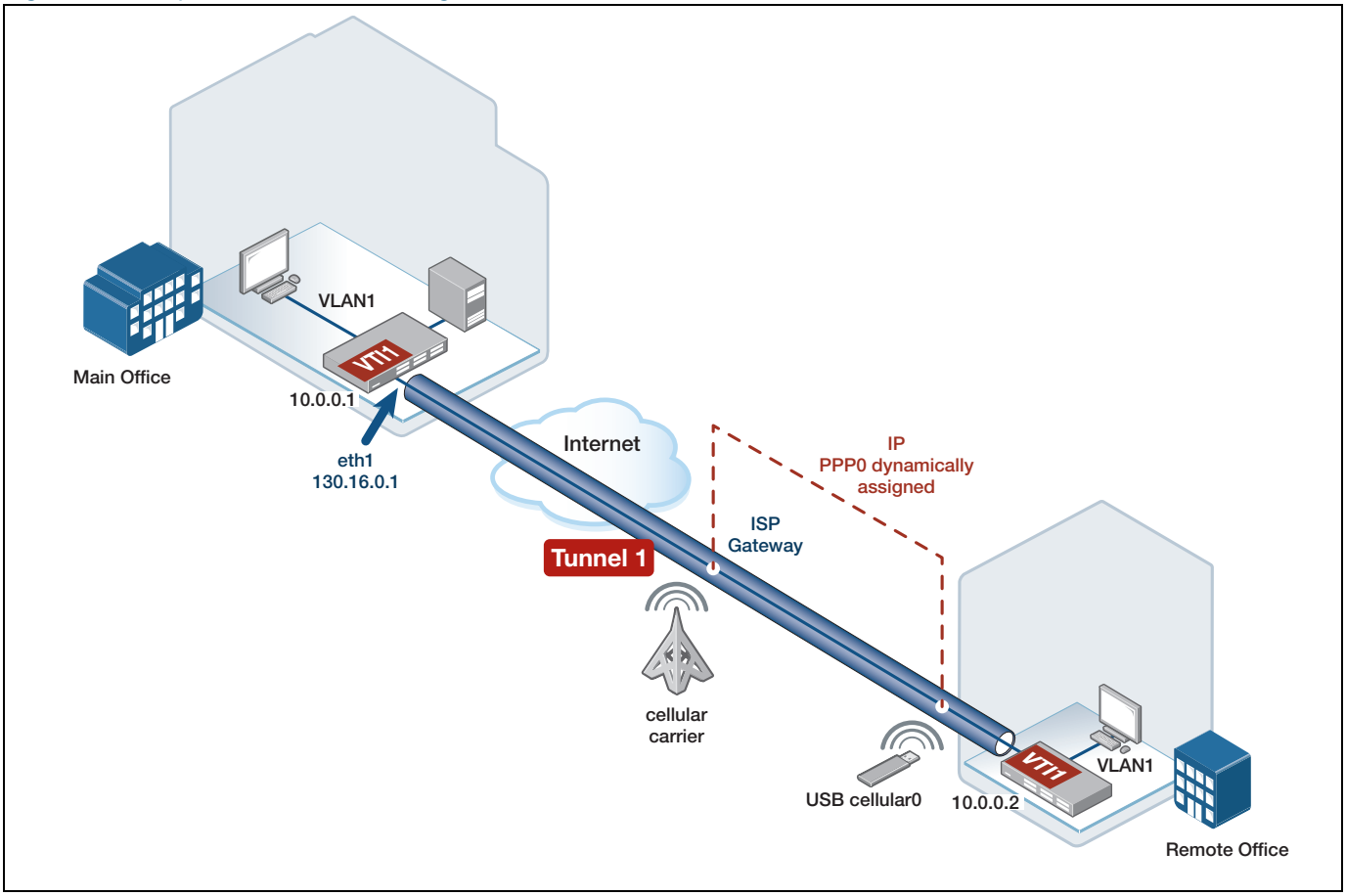
Example **Main Office** configuration for ipv4 over ipv6 IPsec tunnel

```
!
crypto isakmp key SAMPLEKEY address 2001:db8:2::1
!
interface eth1
 ipv6 address 2001:db8:1::1/64
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel1
 description VPN_to_Office
 tunnel source eth1
 tunnel destination 2001:db8:2::1
 tunnel local selector 1 192.168.1.0/24
 tunnel remote selector 1 192.168.2.0/24
 tunnel protection ipsec
 tunnel mode ipsec ipv6
 ip address 10.0.0.1/30
!
ip route 192.168.2.0/24 10.0.0.2
!
ipv6 route ::/0 2001:db8:1::2
!
```

Example 9: A VPN connecting over either a 3G or 4G/LTE cellular interface

Part 1 In this first part, the main office IPsec VPN is via an eth WAN interface, and the remote office IPsec VPN is via a USB 3G cellular interface, whose associated serial PPP WAN IP address and DNS information is dynamically assigned by the carrier. The remote office router operates as the IPsec VPN initiator, and the main office router operates as the IPsec responder to the incoming VPN.

Figure 8: Example remote office configuration for VPN with a 3G cellular/PPP interface



The main site router identifies the incoming VPN based on the tunnel name instead of the peer destination IP address. An Access Point Name (APN) is configured as part of the cellular interface. The APN information is supplied by the carrier that the cellular modem (with its inserted SIM card) connects to. This information is used by the carrier to form a valid Internet connection via its cellular network and the public Internet. The APN allows the cellular carrier to ensure the correct WAN IP address is assigned to the serial PPP interface over the USB 3G Modem, and thereby enabling Internet connectivity via that cellular connection.

For more information about APNs, see the Wikipedia entry.

Example **Remote Office** configuration for VPN with a cellular interface

```
!
crypto ipsec profile remote
 pfs 5
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile remote1
 version 2
 transform 1 integrity SHA256 encryption AES256 group 5
!
crypto isakmp key SAMPLEKEY address 130.16.0.1
!
crypto isakmp peer address 130.16.0.1 profile remote1
!
interface tunnel1
 description VPN_to_Office
 tunnel source ppp0
 tunnel destination 130.16.0.1
 tunnel local name Remote
 tunnel protection ipsec profile remote
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
interface cellular0
 encapsulation ppp 0
 apn <value>
!
interface ppp0
 ppp ipcp dns request
 keepalive
 ip address negotiated
 ip tcp adjust-mss pmtu
!
```

Example **Main Office** configuration for a VPN with a cellular interface

```
!
crypto ipsec profile remote
 pfs 5
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile remote1
 version 2
 transform 1 integrity SHA256 encryption AES256 group 5
!
crypto isakmp key SAMPLEKEY hostname Remote
!
crypto isakmp peer dynamic profile remote1
!
interface eth1
 ip address 130.16.0.1/30
!
interface tunnel1
 tunnel source eth1
 tunnel destination dynamic
 tunnel remote name Remote
 tunnel protection ipsec profile remote
 tunnel mode ipsec ipv4
 ip address 10.0.0.1/30
!
```

Part 2 In this second part, a 4G/LTE USB cellular MODEM is plugged into the device.

The main office configuration remains unchanged.

A private IP address is dynamically allocated via DHCP from the 4G cellular MODEM to the 4G cellular wireless wide area network (WWAN) interface.

The remote office router operates as the IPsec VPN initiator, and the main office operates as the IPsec responder to the incoming VPN.

The IPsec VPN connection will automatically negotiate to use NAT-T, allowing the VPN to traverse any intermediate routing equipment that may performing NAT, such as the 4G MODEM itself, or intermediate carrier equipment performing carrier grade NAT (CGNAT).

VPN traffic initiated from the remote office WWAN interface (sent via the 4G cellular MODEM towards the main office), and subsequent and associated VPN responder traffic (sent from the main office towards the remote office) is automatically passed-through the 4G MODEM to reach the WWAN interface.

Example **Remote Office** configuration for VPN with a 4G cellular/WWAN interface

```
!
crypto ipsec profile remote
 pfs 5
 transform 1 protocol esp integrity SHA256 encryption AES256
!
crypto isakmp profile remote1
 version 2
 transform 1 integrity SHA256 encryption AES256 group 5
!
crypto isakmp key SAMPLEKEY address 130.16.0.1
!
crypto isakmp peer address 130.16.0.1 profile remote1
!
interface wwan0
ip address dhcp
!
interface tunnel1
 description VPN_to_Office
 tunnel source wwan0
 tunnel destination 130.16.0.1
 tunnel local name Remote
 tunnel protection ipsec profile remote
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
```

Note: The 3G PPP interface and the 4G WWAN interface types, both have dynamically assigned addresses. Also, both types of cellular interface can be optionally configured as host entities used by associated NAT and/or firewall rules.

Example 10: IPsec pairing to legacy device with firewall and dynamic IP

This example shows how to configure an AlliedWare Plus firewall to be installed at a remote spoke site and integrated into an existing legacy hub-and-spoke network topology.

Customized IPsec and ISAKMP profiles using legacy crypto transform options, as well as IPsec traffic selectors are configured. This is to allow the firewall to successfully negotiate a VPN using legacy crypto options with the Main Office.

The firewall is connected to the Internet via a PPPoE client WAN link to an ISP PPPoE Access concentrator, in this example using PPPoE service name **any**.

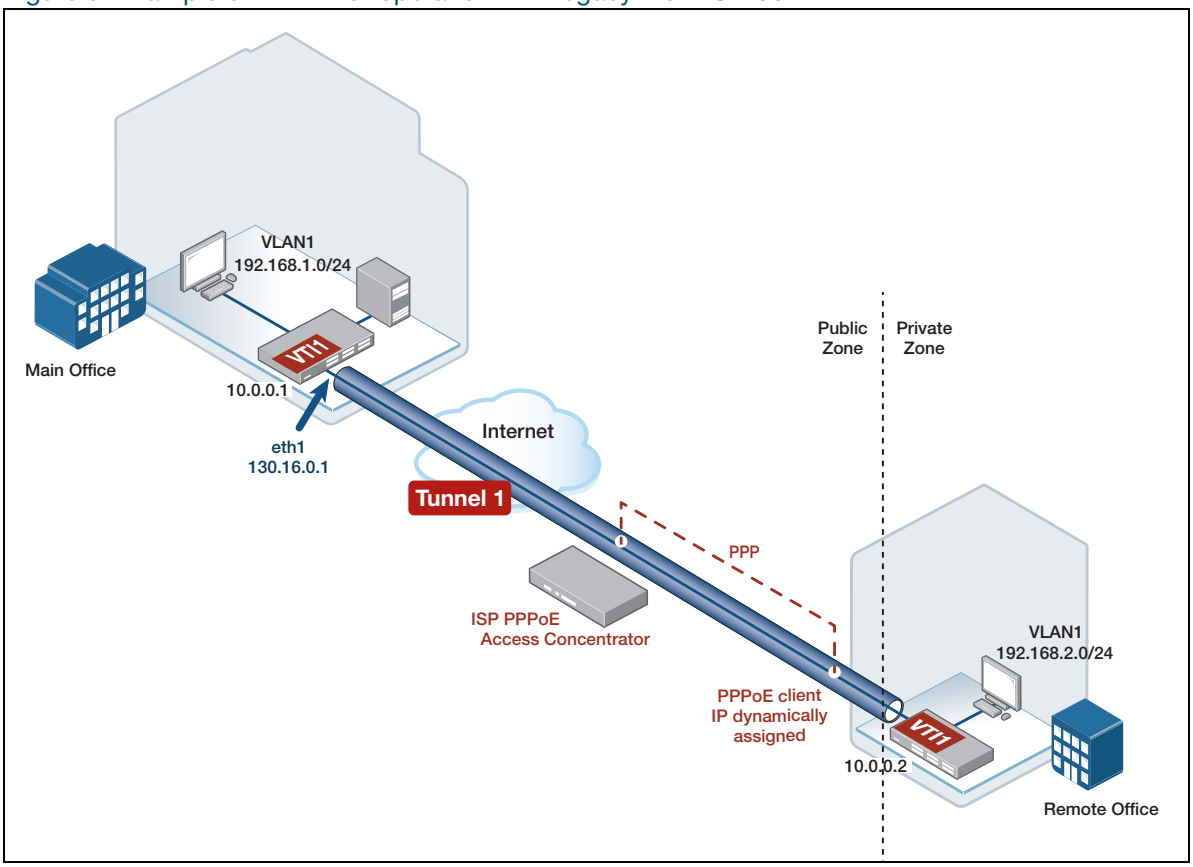
The PPPoE client WAN interface IP address is dynamically assigned. The Main Office router has fixed IP address on its WAN interface.

The PPPoE WAN interface is located in the firewall Public zone. The Main and Remote Office LAN networks, and also VPN traffic terminated at the VTI are located within the firewall private zone.

Traffic flows from private to public zones have NAT masquerade applied, so that the source IP address of traffic sent to the Internet uses the dynamically assigned PPP WAN IP address.

Firewall application rules are configured to allow the IPsec ESP, and ISAKMP traffic to be sent towards the Main office device through the firewall.

Figure 9: Example of VPN inter-operation with legacy **Main Office**



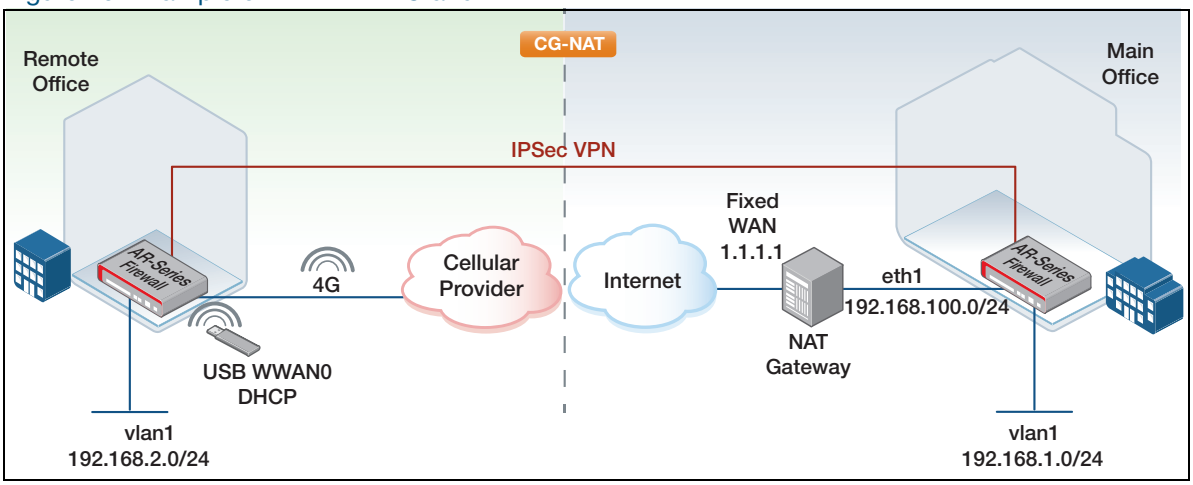
Example: **Remote Office** configuration for VPN inter-operation with **legacy Main Office**

```
!
zone private
network local
  ip subnet 192.168.2.0/24
network remote
  ip subnet 192.168.1.0/24
network tun1
  ip subnet 10.0.0.0/30
!
zone public
network wan
  ip subnet 0.0.0.0/0 interface ppp1
  host router
   ip address dynamic interface ppp1
!
application esp
 protocol 50
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private to public
 rule 30 permit isakmp from public.wan.router to public
 rule 40 permit esp from public.wan.router to public
 protect
!
nat
 rule 10 masq any from private to public
 enable
!
crypto ipsec profile legacy-phase2
 transform 1 protocol esp integrity SHA1 encryption 3DES
!
crypto isakmp profile legacy-phase1
 version 1 mode main
 transform 1 integrity SHA1 encryption 3DES group 2
!
crypto isakmp key samplekey address 130.16.0.1
!
crypto isakmp peer address 130.16.0.1 profile legacy-phase1
!
interface eth1
 encapsulation ppp 1
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel1
 tunnel source ppp1
 tunnel destination 130.16.0.1
 tunnel local name remote_site
 tunnel local selector 192.168.2.0/24
 tunnel remote selector 192.168.1.0/24
 tunnel protection ipsec profile legacy-phase2
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
interface ppp1
 ip address negotiated
 ppp service-name <any>
 ppp username <username>
 ppp password <password>
!
ip route 0.0.0.0/0 ppp1
ip route 192.168.1.0/24 tunnel1
!
```

Example 11: VPN via 4G with NAT traversal between main and remote sites

In this example a 4G USB modem is plugged into the remote office AlliedWare Plus device, acting as the IPsec VPN initiator.

Figure 10: Example of VPN with 4G and NAT



The remote office router wireless WAN interface (WWAN0) operates as a DHCPv4 client, and is dynamically assigned a private 192.168.x.x/x IPv4 address from the RFC 1918 reserved range from an inserted and compatible 4G/LTE USB modem. Via DHCPv4, the 4G USB modem allocates the private inside address and default route (gateway IP) to the cellular WWAN interface of the remote office router.

The IPsec responder is the AlliedWare Plus device and is attached to the internal network of the intermediate NAT gateway device, located at the main office. The main office firewall is configured with fixed IP addresses.

NAT is applied at multiple locations. Both devices are configured with firewall and NAT rules for:

- ISAKMP traffic (UDP port 500)
- NAT-T traffic (UDP port 4500).

NAT-Traversal (NAT-T) encapsulates ESP packets inside UDP and assigns both the source and destination ports as UDP 4500. Therefore, firewall/NAT rules for ESP (protocol 50) are unnecessary.

The remote office firewall is configured with rules to allow outbound VPN traffic originating from its WWAN WAN to the Internet. The main office firewall is configured with rules to allow inbound VPN traffic from the Internet to reach its eth WAN.

Note:

- The 4G modem performs NAT between its inside (private IPv4 subnet), and its outside address
- Large scale NAT (Carrier Grade NAT) is performed by the cellular provider for traffic from the cellular network to the Internet
- There is also an intermediate third-party NAT gateway device, located at the main office.

IPsec NAT-T is used to allow the VPN security VPN to negotiate between the two AlliedWare Plus firewalls, despite intermediate address translations.

VPN traffic initiated from the remote office firewall is destined to the known fixed public WAN IP address of the intermediate NAT gateway device.

The intermediate third party NAT gateway must be configured with firewall/NAT port forwarding rules for ISAKMP traffic (UDP port 500), and NAT-T traffic (UDP port 4500). This is required to allow the incoming VPN traffic initiated from the remote office arriving on its public WAN to be forwarded to the internal fixed (eth1 WAN) address of the main office firewall. The main office firewall virtual tunnel interface is configured with ‘destination dynamic’, since the remote initiator private address is assigned via DHCP and operates behind NAT.

Both main office and remote office firewall tunnel interfaces and pre-shared ISAKMP keys are configured to use local/remote hostnames to identify and match VPN traffic and pre-shared keys. This allows the ISAKMP and associated IPsec VPN security associations to form.

Example **Remote office** site configuration for VPN via 4G with NAT

```
!
zone private
 network local
  ip subnet 192.168.2.0/24
 network remote
  ip subnet 192.168.1.0/24
 network tunnel
  ip subnet 10.0.0.0/30
!
zone public
 network wan
  ip subnet 0.0.0.0/0 interface wwan0
 host router
  ip address dynamic interface wwan0
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
application natt
 protocol udp
 sport 4500
 dport 4500
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private to public
 rule 30 permit isakmp from public.wan.router to public
 rule 40 permit natt from public.wan.router to public
protect
!
nat
 rule 10 masq any from private to public
enable
!
crypto isakmp key SAMPLEKEY hostname office1
!
interface wwan0
 ip address dhcp
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel2
 tunnel source wwan0
 tunnel destination 1.1.1.1
 tunnel local name office2
 tunnel remote name office1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 10.0.0.2/30
!
ip route 192.168.1.0/24 tunnel2
!
```

Example **Main office** site configuration for VPN via 4G with NAT

```
!
zone private
 network local
  ip subnet 192.168.1.0/24
 network remote
  ip subnet 192.168.2.0/24
 network tunnel
  ip subnet 10.0.0.0/30
!
zone public
 network wan
  ip subnet 0.0.0.0/0 interface eth1
 host router
  ip address 192.168.100.1/24
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
application natt
 protocol udp
 sport 4500
 dport 4500
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private to public
 rule 30 permit isakmp from public to public.wan.router
 rule 40 permit natt from public to public.wan.router
protect
!
nat
 rule 10 masq any from private to public
enable
!
crypto isakmp key SAMPLEKEY hostname office2
!
interface eth1
 ip address 192.168.100.1/24
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel2
 tunnel source eth1
 tunnel destination dynamic
 tunnel local name office1
 tunnel remote name office2
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 10.0.0.1/30
!
ip route 0.0.0.0/0 192.168.100.254
ip route 192.168.2.0/24 tunnel2
!
```

Example 12: VPN redundancy between main and remote sites

In this example, both main and remote site routers have dual Internet connections via eth1 and eth2 to two different ISPs.

The main and remote site AlliedWare Plus firewalls each have two VPNs configured, a primary VPN and a backup VPN. Each VPN is terminated by a VTI. In AlliedWare Plus, by default, VPNs are ‘persistent’ and so will automatically attempt to re-establish connectivity should the VPN to the peer go down. Traffic traverses the primary IPsec VPN via eth1. When the Internet connection via eth1 fails, traffic traverses the backup VPN routing path via eth2.

To achieve VPN redundancy, the solution uses a combination of OSPF and static routing via the VPNs between the two offices.

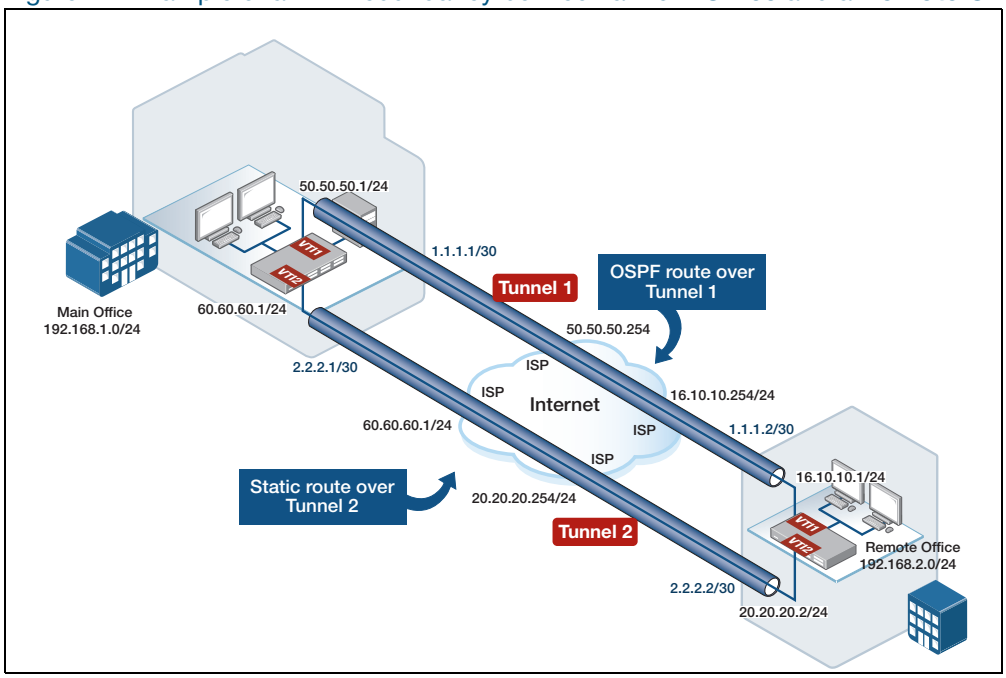
- OSPF routing is used via the VTI (tunnel10, sourced via eth1) terminating the primary IPsec VPN.
- A static route is configured via the VTI (tunnel20, sourced via eth2) terminating the backup IPsec VPN. The static route (via tunnel20) is configured with a high metric, so the route learned by OSPF will be selected as the preferred route for traffic between the private LANs.

If the primary VPN link fails (for example, when there is a failure of the primary Internet connection via eth1), then this results in the OSPF neighbor relationship via the primary VPN going down, and automatic removal of the route to the remote site LAN, learned by OSPF over the VPN. The static routing path via the backup IPsec VPN is then automatically selected, allowing traffic to flow between the office private LANs.

When the primary VPN is re-established, OSPF routes are then re-learned, allowing the traffic to flow via the primary VPN again.

In this example, the full device configurations are included for both AlliedWare Plus firewalls. This includes multi-zone firewall and associated NAT configuration, static and dynamic (OSPF) routing configuration, and VPN configuration.

Figure 11: Example of a VPN redundancy between a **Main Office** and a **Remote Office**



Example **Main office** site configuration for VPN redundancy

```
!
hostname main-office
!
zone private
network remote
 ip subnet 192.168.2.0/24
network local
 ip subnet 192.168.1.0/24 interface vlan1
network tunnel1
 ip subnet 1.1.1.0/30
network tunnel2
 ip subnet 2.2.2.0/30
network ospf_mcast
 ip subnet 224.0.0.5/32
 ip subnet 224.0.0.6/32
!
zone public
network all
 ip subnet 0.0.0.0/0
network intf
 ip subnet 50.50.50.0/24 interface eth1
 ip subnet 60.60.60.0/24 interface eth2
 host router
  ip address 50.50.50.1
  ip address 60.60.60.1
!
application esp
 protocol 50
!
application isakmp
 protocol udp
 sport 500
 dport 500
```

Example **Main office** site configuration for VPN redundancy (continued)

```
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private.local to public
 rule 30 permit esp from public.intf.router to public
 rule 40 permit isakmp from public.intf.router to public
 rule 50 permit esp from public to public.intf.router
 rule 60 permit isakmp from public to public.intf.router
 protect
!
nat
 rule 10 masq any from private.local to public
 enable
!
crypto isakmp key SAMPLEKEY1 address 16.10.10.1
crypto isakmp key SAMPLEKEY2 address 20.20.20.1
!
interface eth1
 ip address 50.50.50.1/24
!
interface eth2
 ip address 60.60.60.1/24
!
interface vlan1
 ip address 192.168.1.254/24
!
interface tunnel1
 tunnel source 50.50.50.1
 tunnel destination 16.10.10.1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 1.1.1.1/30
!
interface tunnel2
 tunnel source 60.60.60.1
 tunnel destination 20.20.20.1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 2.2.2.1/30
!
router ospf
 ospf router-id 1.1.1.1
 passive-interface vlan1
 network 1.1.1.0/30 area 0
 network 192.168.1.0/24 area 0
!
ip route 16.10.10.0/24 50.50.50.254
ip route 20.20.20.0/24 60.60.60.254
ip route 192.168.2.0/24 tunnel2 150
!
```

Example **Remote Office** configuration for VPN redundancy

```
!
hostname remote-office
!
aaa authentication enable default local
aaa authentication login default local
!
zone private
network remote
 ip subnet 192.168.1.0/24
network local
 ip subnet 192.168.2.0/24 interface vlan1
network tunnel1
 ip subnet 1.1.1.0/30
network tunnel2
 ip subnet 2.2.2.0/30
network ospf_mcast
 ip subnet 224.0.0.5/32
 ip subnet 224.0.0.6/32
!
zone public
network all
 ip subnet 0.0.0.0/0
network intf
 ip subnet 16.10.10.0/24 interface eth1
 ip subnet 20.20.20.0/24 interface eth2
 host router
 ip address 16.10.10.1
 ip address 20.20.20.1
!
application esp
 protocol 50
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private.local to public
 rule 30 permit esp from public.intf.router to public
 rule 40 permit isakmp from public.intf.router to public
 rule 50 permit esp from public to public.intf.router
 rule 60 permit isakmp from public to public.intf.router
 protect
!
nat
 rule 10 masq any from private.local to public
 enable
!
crypto isakmp key SAMPLEKEY1 address 50.50.50.1
crypto isakmp key SAMPLEKEY2 address 60.60.60.1
!
```

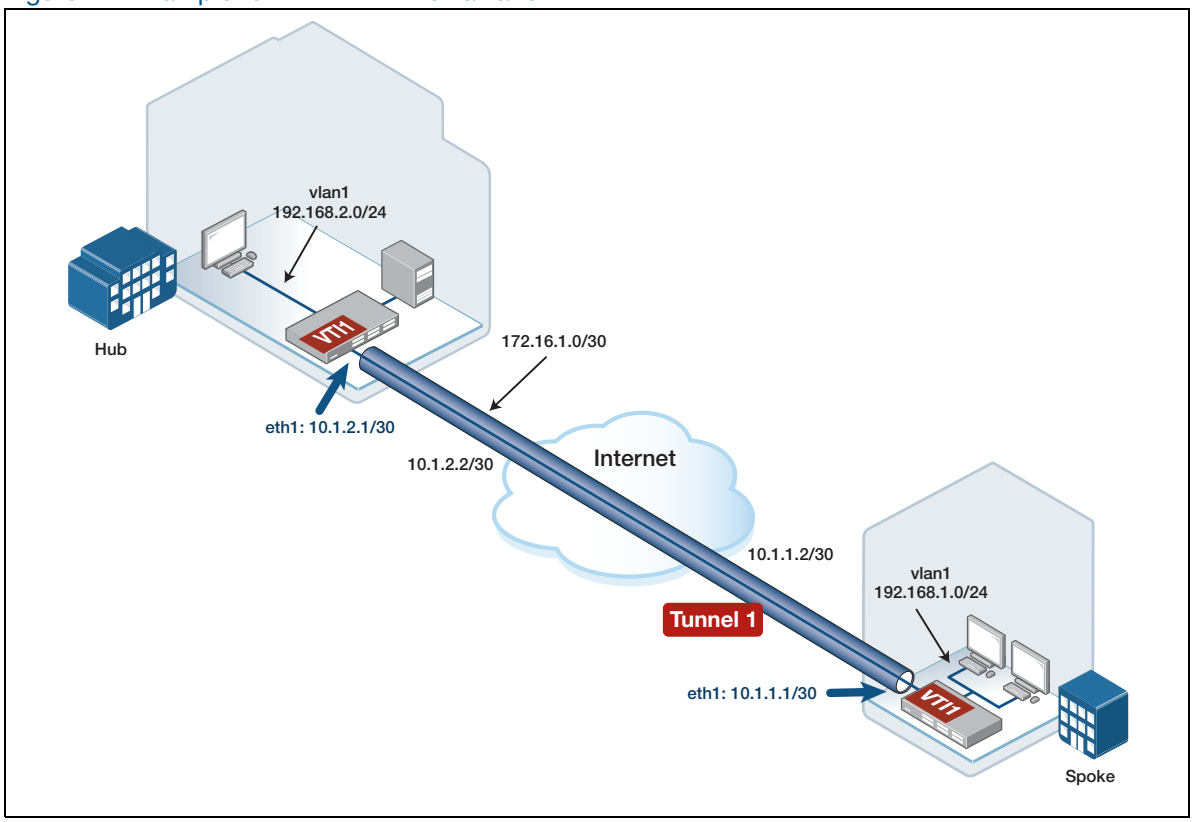
Example **Remote Office** configuration for VPN redundancy (continued)

```
interface eth1
 ip address 16.10.10.1/24
!
interface eth2
 ip address 20.20.20.1/24
!
interface vlan1
 ip address 192.168.2.254/24
!
interface tunnel1
 tunnel source 16.10.10.1
 tunnel destination 50.50.50.1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 1.1.1.2/30
!
interface tunnel2
 tunnel source 20.20.20.1
 tunnel destination 60.60.60.1
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ip address 2.2.2.2/30
!
router ospf
ospf router-id 1.1.1.2
passive-interface vlan1
network 1.1.1.0/30 area 0
network 192.168.2.0/24 area 0
!
ip route 50.50.50.0/24 16.10.10.254
ip route 60.60.60.0/24 20.20.20.254
ip route 192.168.1.0/24 tunnel2 150
!
```

Example 13: VPN with firewall, DPI, and Malware Protection

In this example, there is a site-to-site VPN between two AlliedWare Plus firewalls. Deep Packet Inspection (DPI) and Malware Protection is also enabled.

Figure 12: Example for VPN with Firewall and DPI



By default, the firewall performs DPI on ingress when DPI is enabled. So the encrypted VPN traffic from the peer will be identified by DPI as IPsec traffic on ingress, based on the outer VPN headers.

Tunnel inline-processing

From version 5.5.2-1.1 onwards, you can use **tunnel inline-processing** to improve the forwarding performance of incoming application traffic. Use this feature when traffic is encapsulated within an encrypted VPN and subsequently processed and identified via Deep Packet Inspection (DPI).

Tunnel inline-processing is useful because it means packets are decrypted before being analyzed and processed via the DPI engine. This is especially important for VPN traffic, where you actually want to identify application traffic transported within the IPSEC VPN, rather than the outer encrypted IPsec VPN headers.

Tunnel inline-processing avoids the need to configure **tunnel security-reprocessing**, which is the alternative, less efficient option. With tunnel security-reprocessing configured, the DPI engine processes incoming VPN traffic twice (before and after decryption), in order to identify incoming application traffic transported via an encrypted VPN.

Tunnel inline-processing:

- is applied on ingress on physical ports before DPI processing
- detects and processes ESP headers in NAT-T UDP (port 4500) encapsulated packets

Configuration

Use the following commands to configure tunnel inline-processing:

```
awplus# configure terminal
awplus(config)# interface tunnel0
awplus(config-if)# tunnel inline-processing
```

Use the following command to display tunnel inline-processing counters:

```
awplus# show tunnel inline-processing counters
```

VTI tunnel modes

The tunnel inline-reprocessing feature is supported on either IPsec IPv4 or IPsec IPv6 tunnel types, including the following tunnel combinations:

- IPv4 traffic over IPsec IPv4 tunnels
- IPv6 traffic over IPsec IPv4 tunnels
- IPv4 traffic over IPsec IPv6 tunnels
- IPv6 traffic over IPsec IPv6 tunnels

Note: The tunnel inline-processing feature is not supported on GRE tunnels with IPsec protection, nor L2TPv2/L2TPv3 tunnels with IPsec protection. You can continue to use the global **tunnel security-reprocessing** command and simultaneously configure tunnel inline-processing for individual IPsec mode VTI's.

Tunnel security reprocessing

If you need DPI to inspect and identify application traffic transported within the site-to-site VPN, then you need to enable global tunnel security reprocessing (**tunnel security-reprocessing** command).

When tunnel security reprocessing is enabled, the data-stream that was transported within the VPN is passed through the DPI engine a second time. This allows DPI to inspect traffic that arrives from the VPN (that is terminated on the device) a second time after decryption and decapsulation.

This allows DPI to inspect the contents of those decrypted packets, and identify the embedded application traffic correctly. This ensures any application-based rules, such as PBR rules and firewall rules, can be applied properly to bi-directional application traffic.

Without the tunnel security-reprocessing command configured, DPI is only able to identify the applications transported via the VPN based on the traffic ingressing from LAN to the VPN. DPI will not be able to inspect associated application traffic arriving from the VPN.

This can be a problem for some applications—particularly for applications which use a separate control channel and a dynamically negotiated data channel. For example, consider an application control channel initiated in the direction from local LAN to site remote LAN (via the VPN), with an associated dynamic data channel proposed in the opposite direction (from the remote site device arriving via the VPN). DPI will not be able to associate the outbound control channel data flow that it knows about, with the application data channel because it was proposed within a VPN packet, particularly if the traffic is encrypted.

Tunnel security reprocessing ensures application data can be matched on the bi-directional traffic flows. This ensures any subsequent application-based rules can be properly applied and better match all bi-directional application traffic flows when DPI is also in use.

Caution: Tunnel security reprocessing increases the load on your device and reduces throughput. This is because traffic is processed twice through the DPI engine. Therefore, it should only be enabled if your solution requires it.

The **tunnel security-reprocessing** command is supported from version 5.4.8-0.x and later. It is supported for both built-in and licensed Procera-based DPI Application Awareness.

Tunnel reprocessing is supported for all VTI tunnel modes, such as GRE, IPSEC, OpenVPN and DS-Lite. The principles described above also apply to all AlliedWare Plus firewall stream-based UTM security features: IPS, IP Reputation, Malware Protection and URL Filtering.

Example The following configuration uses tunnel security reprocessing with licensed feature Procera DPI. A firewall rule is configured to block the application RTSP at the Spoke device, which would otherwise flow via the VPN. Also, licensed UTM feature Malware Protection is enabled to scan for Malware threats for all traffic flows via the device. This includes bi-directional data streams transported within the VPN.

Spoke

Spoke firewall configuration

```
!
hostname SPOKE
!
zone private
 network local
  ip subnet 192.168.1.0/24
 network remote
  ip subnet 192.168.2.0/24
 network tun1
  ip subnet 172.16.1.0/30
!
zone public
 network internet
  ip subnet 0.0.0.0/0 interface eth1
  host wan_local
   ip address 10.1.1.1
  host wan_remote
   ip address 10.1.2.1
!
```

Spoke firewall configuration (continued)

```
application esp
 protocol 50
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
firewall
 rule 5 deny RTSP from private to private
 rule 10 permit any from private to private
 rule 20 permit any from private to public
 rule 30 permit any from public.internet.wan_local to public
 rule 40 permit esp from public.internet.wan_remote to public.internet.wan_local
 rule 50 permit isakmp from public.internet.wan_remote to
public.internet.wan_local
 protect
!
nat
 rule 10 masq any from private to public
 enable
!
malware-protection
 provider Kaspersky
 protect
!
dpi
 provider procera
 enable
!
crypto isakmp key <samplekey> address 10.1.2.1
!
tunnel security-reprocessing
!
interface eth1
 ip address 10.1.1.1/30
!
interface vlan1
 ip address 192.168.1.0/24
!
interface tunnel1
  tunnel source 10.1.1.1
  tunnel destination 10.1.2.1
  tunnel protection ipsec
  tunnel mode ipsec ipv4
  ip address 172.16.1.1/30
!
ip route 0.0.0.0/0 10.1.1.2
ip route 192.168.2.0/24 172.16.1.2
!
```

Hub

Hub firewall configuration

```
!
hostname HUB
!
zone private
 network local
  ip subnet 192.168.2.0/24
 network remote
  ip subnet 192.168.1.0/24
 network tun1
  ip subnet 172.16.1.0/30
!
zone public
 network internet
  ip subnet 0.0.0.0/0 interface eth1
  host wan_local
   ip address 10.1.2.1
  host wan_remote
   ip address 10.1.1.1
!
application esp
 protocol 50
!
application isakmp
 protocol udp
 sport 500
 dport 500
!
firewall
 rule 10 permit any from private to private
 rule 20 permit any from private to public
 rule 30 permit any from public.internet.wan_local to public
 rule 40 permit esp from public.internet.wan_remote to public.internet.wan_local
 rule 50 permit isakmp from public.internet.wan_remote to
public.internet.wan_local
 protect
!
nat
 rule 10 masq any from private to public
 enable
!
dpi
 provider procera
 enable
!
crypto isakmp key <samplekey> address 10.1.1.1
!
interface eth1
 ip address 10.1.2.1/30
!
interface vlan1
 ip address 192.168.2.0/24
!
interface tunnel1
  tunnel source 10.1.2.1
  tunnel destination 10.1.1.1
  tunnel protection ipsec
  tunnel mode ipsec ipv4
  ip address 172.16.1.2/30
!
ip route 0.0.0.0/0 10.1.2.2
ip route 192.168.1.0/24 172.16.1.1
!
```

Example 14: IPsec certificate-based authentication

Introduction

There are several reasons why you might use Internet Protocol Security (IPsec) certificate-based authentication:

Security: Certificate-based authentication provides a higher level of security than other authentication methods, such as pre-shared keys. This is because it eliminates the need to distribute and protect shared secrets, and it also makes it more difficult for an attacker to impersonate a legitimate peer.

Scalability: Using certificate-based authentication makes it easier to scale an IPsec deployment. This is because it eliminates the need to configure each peer individually, and it also allows for automated provisioning of new peers.

Non-repudiation: By using digital certificates, IPsec certificate-based authentication provides non-repudiation. This means that a peer can prove that it was the sender of a specific packet, and the recipient can prove that a specific packet was received from a specific peer.

Compliance: Some regulatory requirements, such as HIPAA, may require the use of certificate-based authentication for secure communications.

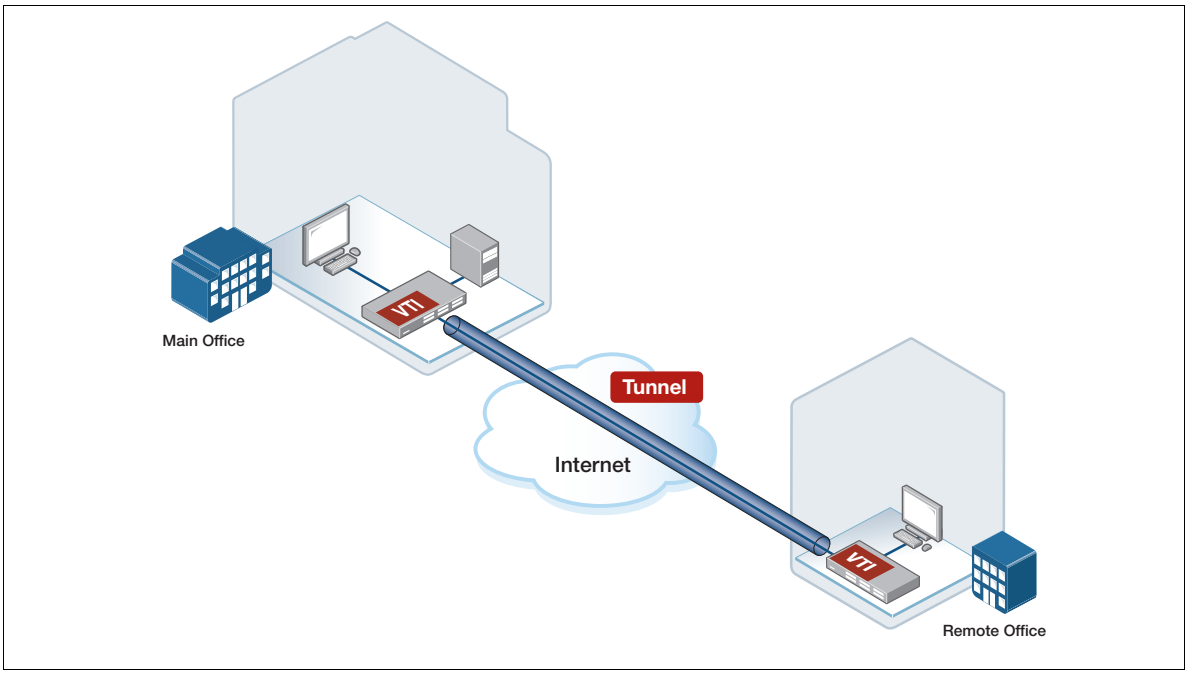
Stronger Authentication: Certificate-based authentication uses the public key infrastructure, which is considered a stronger form of authentication as compared to pre-shared keys. This is because the public key infrastructure uses the private key for signing the certificate and the public key for the verification.

Ease of management: IPsec certificate-based authentication enables central management of the certificates and the associated private keys. This makes it easier to revoke or replace certificates, and also eases the burden of managing the keys.

In summary, IPsec certificate-based authentication provides a more secure, scalable, and manageable solution for secure IP communications than other methods like pre-shared keys. It also helps to meet regulatory and compliance requirements and provides non-repudiation.

What's in this example?

This example uses AlliedWare Plus gateways to manage certificates in a variety of site-to-site VPN scenarios.



This example is divided into the following sections:

- "Important concepts" on page 57
- "Scenarios" on page 58, each with configuration and show output.

Important concepts

Recognizing a number of terms is useful when implementing IPsec on your systems. Some of these terms and their description are listed in the following table:

Term	Description
VPN	Virtual Private Networks provide secure communication over an untrusted network.
IPsec	Internet Protocol Security defines the protection of IP packets using encryption and authentication.
PKI	A Public Key Infrastructure is the mechanism that a device uses to manage and authenticate its trusted hierarchy of digital certificates. A PKI can range from a manually distributed scheme in a small company, to automated distribution with public Certification Authorities (CA) as is used in web browsers for SSL/TLS.
DES/3DES/AES	Symmetric key ciphers used for bulk data encryption. The Data Encryption Standard algorithm is no longer considered secure and was replaced by Triple DES and now the AES (Advanced Encryption Standard).
RSA/DSA	Asymmetric key algorithms used for authentication and a major component of a PKI.
Public/Private Key	Parts of an RSA/DSA key; the private component is kept secret and the public component is freely distributed (usually in the form of a certificate).
Certificate	A combination of an entities identity and public key that has been signed by a third party proving its authenticity.
CA	Certificate Authority is an entity with the ability to sign certificates.
Self-signed CA	A CA that signs their own certificate.
Intermediate CA	A CA whose certificate is signed by another CA.
CSR	Certificate Signing Requests are sent to CAs to obtain a signed certificate. A CSR should contain all information needed for the signed certificate.
PEM	Privacy Enhanced Mail is the most common format for certificates, CSRs, and public/private keys. PEM is used by AlliedWare Plus to manage certificates, including importing and exporting.

Scenarios

For information on creating and managing certificates, see "Certificate management" on page 68.

One-sided local self-signed CA certificate

In this scenario, we configure a site-to-site VPN between two AlliedWare Plus routers, R1 and R2, with R1 authenticated by a certificate and R2 authenticated by a pre-shared key (PSK):

Setup a local self-signed CA

Perform the following steps to generate a self-signed CA locally with the trustpoint CLI on an AlliedWare Plus device:

1. Create R1's server certificate signed by a self-signed CA, see "Server certificate signed by local self-signed CA" on page 68.
2. Export R1's copy of the self-signed CA's certificate, in PEM format.

```
R1# crypto pki export ATL-Corporate pem terminal
---------------------------------------------------------------
-----BEGIN CERTIFICATE-----
MIIDdjCCAl6gAwIBAgIJAMLO9mdPucoIMA0GCSqGSIb3DQEBCwUAMEsxHTAbBgNV
...
...
-----END CERTIFICATE-----
---------------------------------------------------------------
```

3. Configure R2 to trust R1’s self-signed CA, see "Trust CA certificate" on page 71, using the exported certificate from above.

Example configuration

For the following configurations, ensure that R1's tunnel0 **local name** and R2's tunnel0 **remote name** match the subject alternative name in R1's generated certificate.

R1

```
crypto pki trustpoint ATL-Corporate
 enrollment selfsigned
 subject-alt-name atl.corporate.com
 subject-name /CN=atl.corporate.com
!
crypto isakmp profile cert_psk
 local authentication certificate
 pki trustpoint ATL-Corporate
 ...
!
crypto isakmp key friend hostname atl.sales.com
!
crypto isakmp peer hostname atl.sales.com profile cert_psk
!
interface tunnel0
 tunnel local name atl.corporate.com
 tunnel remote name atl.sales.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

R2

```
crypto pki trustpoint ATL-Corporate
 enrollment terminal
!
crypto isakmp profile psk_cert
 remote authentication certificate
 pki trustpoint ATL-Corporate
 ...
!
crypto isakmp key friend hostname atl.corporate.com
!
crypto isakmp peer hostname atl.corporate.com profile psk_cert
!
interface tunnel0
 tunnel local name atl.sales.com
 tunnel remote name atl.corporate.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

Example show commands

R1

```
awplus# show crypto pki certificates
--------------------
Trustpoint "ATL-Corporate" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.corporate.com
   Issuer      : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV108ACD2CFB74D7E
   Valid From  : Nov 23 01:05:01 2022 GMT
   Valid To    : Nov 22 01:05:01 2027 GMT
   Fingerprint : D8B6A788 CA1F069D 65FF8D68 42625052 12DDD8D1

Self-signed root certificate
   Subject     : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV108ACD2CFB74D7E
   Issuer      : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV108ACD2CFB74D7E
   Valid From  : Nov 23 01:05:00 2022 GMT
   Valid To    : Nov 20 01:05:00 2032 GMT
   Fingerprint : 97051FCF 7ABBC61F 5C149757 4F7994E8 96F10594

awplus# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Corporate"
   Type            : Self-signed certificate authority
   Root Certificate: 97051FCF 7ABBC61F 5C149757 4F7994E8 96F10594
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : selfsigned
      Subject      : /CN=atl.corporate.com
      Subj Alt Name: atl.corporate.com
```

R2

```
R2# show crypto pki certificates
--------------------
Trustpoint "ATL-Corporate" Certificate Chain
--------------------
Imported root certificate
   Subject     : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV108ACD2CFB74D7E
   Issuer      : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV108ACD2CFB74D7E
   Valid From  : Nov 23 01:05:00 2022 GMT
   Valid To    : Nov 20 01:05:00 2032 GMT
   Fingerprint : 97051FCF 7ABBC61F 5C149757 4F7994E8 96F10594

R2# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Corporate"
   Type            : External certificate authority
   Root Certificate: 97051FCF 7ABBC61F 5C149757 4F7994E8 96F10594
   Local Server    : The server is not enrolled to this trustpoint.

   Authentication and Enrollment Parameters
      Enrollment   : terminal
```

One-sided external self-signed CA signed certificate.

In this scenario, we configure a site-to-site VPN between two AlliedWare Plus routers, R1 and R2, where R1 is authenticated via an external self-signed CA, and R2 is authenticated via PSK.

Generate certificates

This scenario involves the authentication of R1 via a server certificate signed by a self-signed CA.

- This example requires a self-signed CA, see "AlliedWare Plus server certificate signed by external self-signed CA" on page 69.

Example configuration

R1

```
crypto pki trustpoint ATL-Corporate
 enrollment terminal
 subject-alt-name atl.corporate.com
 subject-name /CN=atl.corporate.com
!
crypto isakmp profile cert_psk
 local authentication certificate
 pki trustpoint ATL-Corporate
 ...
!
crypto isakmp key friend hostname atl.sales.com
!
crypto isakmp peer hostname atl.sales.com profile cert_psk
!
interface tunnel0
 tunnel local name atl.corporate.com
 tunnel remote name atl.sales.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

R2

```
crypto pki trustpoint CA
 enrollment terminal
!
crypto isakmp profile psk_cert
 remote authentication certificate
 pki trustpoint CA
 ...
!
crypto isakmp key 8 hrk/vsst437i5m9EnxhSMuWQU7fPx5mQEq7Ta1Jz+Qc= hostname
atl.corporate.com
!
crypto isakmp peer hostname atl.corporate.com profile psk_cert
!
interface tunnel0
 tunnel local name atl.sales.com
 tunnel remote name atl.corporate.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

Example show commands

R1

```
R1# show crypto pki certificates
--------------------
Trustpoint "ATL-Corporate" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.corporate.com
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:29 2022 GMT
   Valid To    : Nov 24 01:05:29 2022 GMT
   Fingerprint : E1CCF52D 7A9231BA FD1F31BC B1FD5DF7 C056F3C6

Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:26 2022 GMT
   Valid To    : Nov 24 01:05:25 2022 GMT
   Fingerprint : 3E6977E0 9AA9CCDF AE8F457C 1E837AED 627F698F

awplus# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Corporate"
   Type            : External certificate authority
   Root Certificate: 3E6977E0 9AA9CCDF AE8F457C 1E837AED 627F698F
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : terminal
      Subject      : /CN=atl.corporate.com
      Subj Alt Name: atl.corporate.com
```

R2

```
R2# show crypto pki certificates
--------------------
Trustpoint "CA" Certificate Chain
--------------------
Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:26 2022 GMT
   Valid To    : Nov 24 01:05:25 2022 GMT
   Fingerprint : 3E6977E0 9AA9CCDF AE8F457C 1E837AED 627F698F

awplus# show crypto pki trustpoint
--------------------
Trustpoint "CA"
   Type            : External certificate authority
   Root Certificate: 3E6977E0 9AA9CCDF AE8F457C 1E837AED 627F698F
   Local Server    : The server is not enrolled to this trustpoint.

   Authentication and Enrollment Parameters
      Enrollment   : terminal
```

Two-sided external self-signed CA signed certificate

In this scenario, we configure a site-to-site VPN between two AlliedWare Plus routers, R1 and R2, where both R1 and R2 are authenticated via the same external self-signed CA.

Generate certificates

This scenario involves the authentication of R1 and R2 via a server, with both certificates signed by a self-signed CA.

- This example requires a self-signed certificate. For information on self-signed CAs, see "Certificate management" on page 68.

Example configuration

R1

```
crypto pki trustpoint ATL-Corporate
 enrollment terminal
 subject-alt-name atl.corporate.com
 subject-name /CN=atl.corporate.com
!
crypto isakmp profile isakmp0
 local authentication certificate
 remote authentication certificate
 pki trustpoint ATL-Corporate
 ...
!
crypto isakmp peer hostname atl.sales.com profile isakmp0
!
interface tunnel0
 tunnel local name atl.corporate.com
 tunnel remote name atl.sales.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

R2

```
crypto pki trustpoint ATL-Sales
 enrollment terminal
 subject-alt-name atl.sales.com
 subject-name /CN=atl.sales.com
!
crypto isakmp profile isakmp0
 local authentication certificate
 remote authentication certificate
 pki trustpoint ATL-Sales
 ...
!
crypto isakmp peer hostname atl.corporate.com profile isakmp0
!
interface tunnel0
 tunnel local name atl.sales.com
 tunnel remote name atl.corporate.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
 ...
```

Example show commands

R1

```
R1# show crypto pki certificates
--------------------
Trustpoint "ATL-Corporate" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.corporate.com
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:59 2022 GMT
   Valid To    : Nov 24 01:05:58 2022 GMT
   Fingerprint : 99C113CA E1DB43C0 D65161D6 D0877952 425D88AF

Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:55 2022 GMT
   Valid To    : Nov 24 01:05:55 2022 GMT
   Fingerprint : 5063A500 951400C2 5F793CF4 3201C0D2 312012B1

awplus# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Corporate"
   Type            : External certificate authority
   Root Certificate: 5063A500 951400C2 5F793CF4 3201C0D2 312012B1
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : terminal
      Subject      : /CN=atl.corporate.com
      Subj Alt Name: atl.corporate.com
```

R2

```
awplus# show crypto pki certificates
--------------------
Trustpoint "ATL-Sales" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.sales.com
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:06:02 2022 GMT
   Valid To    : Nov 24 01:06:02 2022 GMT
   Fingerprint : 3BA9BF7F 3B01F32B DF5DD520 3FBB817B 349646D5

Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:05:55 2022 GMT
   Valid To    : Nov 24 01:05:55 2022 GMT
   Fingerprint : 5063A500 951400C2 5F793CF4 3201C0D2 312012B1

awplus# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Sales"
   Type            : External certificate authority
   Root Certificate: 5063A500 951400C2 5F793CF4 3201C0D2 312012B1
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : terminal
      Subject      : /CN=atl.sales.com
      Subj Alt Name: atl.sales.com
```

Two-sided different external intermediate CA signed certificates

In this scenario, we configure a site-to-site VPN between two AlliedWare Plus routers, R1 and R2, where both R1 and R2 are authenticated via different external intermediate CAs.

Generate certificates

This scenario involves the authentication of R1 and R2 via server certificates signed by different intermediate CAs. R1's server certificate is signed by the intermediate CA INT-CA1. R2's server certificate is signed by the intermediate CA INT-CA2.

- This example requires an Intermediate CA.
- Create a server certificate for R1 externally signed by the intermediate INT-CA1.
- Create an intermediate CA, INT-CA2. INT-CA2 should be signed by the same self-signed CA
- Create a server certificate for R2 externally signed by the intermediate INT-CA2.

Example configuration

R1

```
crypto pki trustpoint ATL-Corporate
 enrollment terminal
 subject-alt-name atl.corporate.com
 subject-name /CN=atl.corporate.com
!
crypto isakmp profile cert_cert
 local authentication certificate
 remote authentication certificate
 pki trustpoint ATL-Corporate
...
!
crypto isakmp peer hostname atl.sales.com profile cert_cert
!
interface tunnel0
 tunnel local name atl.corporate.com
 tunnel remote name atl.sales.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
```

R2

```
crypto pki trustpoint ATL-Sales
 enrollment terminal
 subject-alt-name atl.sales.com
 subject-name /CN=atl.sales.com
!
crypto isakmp profile cert_cert
 local authentication certificate
 remote authentication certificate
 pki trustpoint ATL-Sales
...
!
crypto isakmp peer hostname atl.corporate.com profile cert_cert
!
interface tunnel0
 tunnel local name atl.sales.com
 tunnel remote name atl.corporate.com
 tunnel protection ipsec
 tunnel mode ipsec ipv4
```

Example show commands

R1

```
R1# show crypto pki certificates
--------------------
Trustpoint "ATL-Corporate" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.corporate.com
   Issuer      : /C=NZ/ST=INT-CA1-aes128-None/O=INT-CA1-aes128-None
                 /OU=INT-CA1-aes128-None/CN=INT-CA1-aes128-None
                 /emailAddress=INT-CA1-aes128-None@INT-CA1.co.nz
   Valid From  : Nov 23 00:06:57 2022 GMT
   Valid To    : Nov 24 01:06:56 2022 GMT
   Fingerprint : 31308059 C74478DD ABE451A9 1E79BDF5 6320B593

Intermediate CA certificate
   Subject     : /C=NZ/ST=INT-CA1-aes128-None/O=INT-CA1-aes128-None
                 /OU=INT-CA1-aes128-None/CN=INT-CA1-aes128-None
                 /emailAddress=INT-CA1-aes128-None@INT-CA1.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:06:53 2022 GMT
   Valid To    : Nov 24 01:06:52 2022 GMT
   Fingerprint : 4FAF5343 5C9F8795 C9157763 EA061EC2 CBC7BC81

Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:06:51 2022 GMT
   Valid To    : Nov 24 01:06:51 2022 GMT
   Fingerprint : EF0AB610 5FD209FF 5D748DE1 9AE5B2DD C9FCB5E7

R1# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Corporate"
   Type            : External certificate authority
   Root Certificate: EF0AB610 5FD209FF 5D748DE1 9AE5B2DD C9FCB5E7
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : terminal
      Subject      : /CN=atl.corporate.com
      Subj Alt Name: atl.corporate.com
```

R2

```
R2# show crypto pki certificates
--------------------
Trustpoint "ATL-Sales" Certificate Chain
--------------------
Server certificate
   Subject     : /CN=atl.sales.com
   Issuer      : /C=NZ/ST=INT-CA2-aes128-None/O=INT-CA2-aes128-None
                 /OU=INT-CA2-aes128-None/CN=INT-CA2-aes128-None
                 /emailAddress=INT-CA2-aes128-None@INT-CA2.co.nz
   Valid From  : Nov 23 00:07:02 2022 GMT
   Valid To    : Nov 24 01:07:01 2022 GMT
   Fingerprint : 657EABFD 77EBAD4E 724A1AFF FB092B38 7A4A4EF0

Intermediate CA certificate
   Subject     : /C=NZ/ST=INT-CA2-aes128-None/O=INT-CA2-aes128-None
                 /OU=INT-CA2-aes128-None/CN=INT-CA2-aes128-None
                 /emailAddress=INT-CA2-aes128-None@INT-CA2.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:06:58 2022 GMT
   Valid To    : Nov 24 01:06:58 2022 GMT
   Fingerprint : E9A25CFB FB4C3645 78B73161 42C67F2B 84475E56

Imported root certificate
   Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
                 /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
   Valid From  : Nov 23 00:06:51 2022 GMT
   Valid To    : Nov 24 01:06:51 2022 GMT
   Fingerprint : EF0AB610 5FD209FF 5D748DE1 9AE5B2DD C9FCB5E7

R2# show crypto pki trustpoint
--------------------
Trustpoint "ATL-Sales"
   Type            : External certificate authority
   Root Certificate: EF0AB610 5FD209FF 5D748DE1 9AE5B2DD C9FCB5E7
   Local Server    : The server is enrolled to this trustpoint.
   Server Key      : server-default

   Authentication and Enrollment Parameters
      Enrollment   : terminal
      Subject      : /CN=atl.sales.com
      Subj Alt Name: atl.sales.com
```

Certificate management

Keep in mind that the certificate examples serve only as a demonstration, and specifics such as the subject and expiry date will vary each time you generate a certificate.

Server certificate signed by local self-signed CA

This example outlines the creation of both a server certificate and a self-signed CA certificate on an AlliedWare Plus router, R1.

1. Create a self-signed trustpoint.

```
R1# configure terminal
R1(config)# crypto pki trustpoint ATL-Corporate
R1(ca-trustpoint)# enrollment selfsigned
R1(ca-trustpoint)# end
R1# crypto pki authenticate ATL-Corporate
Generating 2048-bit key for local CA...
Successfully authenticated trustpoint "ATL-Corporate".
```

2. Create a server certificate for authentication of R1, and sign it using R1's self-signed CA. In this step, enter R1-specific parameters such as the **subject** and **subject-alt-name**.

```
R1# configure terminal
R1(config)# crypto pki trustpoint ATL-Corporate
R1(ca-trustpoint)# subject /CN=atl.corporate.com
R1(ca-trustpoint)# subject-alt-name atl.corporate.com
R1(ca-trustpoint)# end
R1# crypto pki enroll ATL-Corporate
Generating 2048-bit key "server-default"...
Successfully enrolled the local server.
```

3. Export R1's self-signed certificate to copy to any device that should trust R1.

```
R1#crypto pki export ATL-Corporate pem
---------------------------------------------------------------
-----BEGIN CERTIFICATE-----
MIIDdDCCAlygAwIBAgIJANlqDnzc04psMA0GCSqGSIb3DQEBCwUAMEoxHTAbBgNV
...
-----END CERTIFICATE-----
---------------------------------------------------------------
```

AlliedWare Plus server certificate signed by external self-signed CA

In this example, we will create a server certificate for an AlliedWare Plus router named R1, which will be signed by an external self-signed CA. This example requires a self-signed CA, managed externally from the AlliedWare Plus router.

1. Create a trustpoint and set **R1** specific certificate details, such as **subject** and **subject-alt-name**.

```
R1# configure terminal
R1(config)# crypto pki trustpoint ATL-Corporate
Created trustpoint "ATL-Corporate".
R1(ca-trustpoint)# enrollment terminal
R1(ca-trustpoint)# subject /CN=atl.corporate.com
R1(ca-trustpoint)# subject-alt-name atl.corporate.com
```

2. Import the CA’s certificate to R1's trustpoint.

```
R1# crypto pki authenticate ATL-Corporate
Paste the certificate PEM file into the terminal.
Type "abort
```

- Paste in the self-signed CA's public certificate, the following uses the PEM format.

```
-----BEGIN CERTIFICATE-----
MIIDvzCCAqegAwIBAgIBATANBgkqhkiG9w0BAQsFADCBmTELMAkGA1UEBhMCTlox
...
...
...
-----END CERTIFICATE-----
```

- Accept the imported certificate.

```
Complete ("END CERTIFICATE" detected).
Subject     : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
              /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
              /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
Valid From  : Nov 18 21:06:05 2022 GMT
Valid To    : Nov 19 22:06:05 2022 GMT
Fingerprint : C068ACBB 6563DFC6 90E4BA80 CB799B94 04825A2C
This is a self-signed CA certificate.
The certificate has been validated successfully.
Accept this certificate?  y

Successfully authenticated trustpoint "ATL-Corporate".
```

3. Generate R1's CSR for the self-signed CA.

```
R1# crypto pki enroll ATL-Corporate
Generating 2048-bit key "server-default"...
Cut and paste this request to the certificate authority:
---------------------------------------------------------------
-----BEGIN CERTIFICATE REQUEST-----
MIICyjCCAbICAQAwHDEaMBgGA1UEAwwRYXRsLmNvcnBvcmF0ZS5jb20wggEiMA0G
...
...
...
-----END CERTIFICATE REQUEST-----
---------------------------------------------------------------
```

4. Wait for the CA to sign R1's CSR.

5. Import the server certificate generated by the self-signed CA from R1’s CSR.

```
crypto pki import ATL-Corporate pem terminal
Paste the certificate PEM file into the terminal.
Type "abort
<pre>
Paste in '''R1''''s server certificate
<pre>
-----BEGIN CERTIFICATE-----
MIIDizCCAnOgAwIBAgIBAjANBgkqhkiG9w0BAQsFADCBmTELMAkGA1UEBhMCTlox
...
...
...
-----END CERTIFICATE-----
" to cancel.
```

6. Accept R1's imported server certificate.

```
Complete ("END CERTIFICATE" detected).
Subject     : /CN=atl.corporate.com
Issuer      : /C=NZ/ST=CA-aes128-None/O=CA-aes128-None/OU=CA-aes128-None
              /CN=CA-aes128-None/emailAddress=CA-aes128-None@CA.co.nz
Valid From  : Nov 18 21:06:08 2022 GMT
Valid To    : Nov 19 22:06:08 2022 GMT
Fingerprint : 14B65C5D B5489EE8 62E6E451 28414021 62480F62
This is not a valid CA certificate. Attempting to import as a server certificate.
The certificate has been validated successfully.
Accept this certificate?  y

The certificate was successfully imported.
```

Trust CA certificate

In this example, we configure the AlliedWare Plus router, R1, to trust a CA and all certificates signed by the CA including past, present, and future. Please note that when creating a server certificate for R1 (as described in "AlliedWare Plus server certificate signed by external self-signed CA" on page 69), the CA will be trusted too.

1. Create a trustpoint to store the public CA certificate.

```
R1# configure terminal
R1(config)# crypto pki trustpoint ATL-Corporate
Created trustpoint "ATL-Corporate".
R1(ca-trustpoint)# enrollment terminal
R1(ca-trustpoint)# end
```

2. Import the CA certificate.

```
R1# crypto pki authenticate ATL-Corporate
Paste the certificate PEM file into the terminal.
Type "abort" to cancel
```

- Paste in the CA's certificate.

```
-----BEGIN CERTIFICATE-----
MIIDdjCCAl6gAwIBAgIJAMLO9mdPucoIMA0GCSqGSIb3DQEBCwUAMEsxHTAbBgNV
...
...
...
-----END CERTIFICATE-----
```

- Confirm the completed message and accept the certificate.

```
Complete ("END CERTIFICATE" detected).
Subject     : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV1000000000000000
Issuer      : /O=Allied Telesis, Inc./CN=AlliedWarePlusCAV1000000000000000
Valid From  : Nov 18 02:45:18 2022 GMT
Valid To    : Nov 15 02:45:18 2032 GMT
Fingerprint : 8EAB5A05 0E22AEC6 9BB8453A 14E01983 AEDA808F
This is a self-signed CA certificate.
The certificate has been validated successfully.
Accept this certificate?  y

Successfully authenticated trustpoint "ATL-Corporate".
```

For more information about certificate usage, see the Public Key Infrastructure (PKI) Feature Overview and Configuration Guide.

Diagnostics

Checking the state of ISAKMP and IPsec security associations

There are several useful commands to display the state of ISAKMP and IPsec security associations.

show isakmp sa

Use the command **show isakmp sa** to check the state of the ISAKMP security association formed between two IPsec peers:

```
awplus#show isakmp sa
-------------------------------------------------------------------------------
Peer                Cookies (initiator:responder)       Auth  Ver   Expires
                    Encryption    Integrity    Group    DPD   NATT  State
-------------------------------------------------------------------------------
10.0.0.20           f93c2717a1ece407:972bc0c77344d7a4   PSK   1     78340s
                    AES256        SHA256       2        yes   no    Established
10.0.0.22           ccb7f90b54945375:2642525bd20f3428   PSK   1     3334s
                    3DES          SHA1         2        yes   no    Established
10.0.0.25           bd0efef134c86656:d46d0b1b72b46444   PSK   1     819s
                    AES128        SHA1         2        yes   no    Established
```

show ipsec sa

Use the command **show ipsec sa** to show the state of the IPsec security association formed between two IPsec peers:

```
awplus#show ipsec sa

-------------------------------------------------------------------------------

Peer                  SPI (in:out)          Mode          Proto  Expires
                      Encryption            Integrity     PFS
-------------------------------------------------------------------------------
10.0.0.20             c2d8c150:7b24d3f5     tunnel        ESP    28786s
                      AES256                SHA256        -
10.0.0.22             c6c2ad0d:0d008e3d     tunnel        ESP    3582s
                      3DES                  SHA1          -
10.0.0.25             cb36f9dd:cd87a834     tunnel        ESP    28778s

                      AES128                SHA1          2
```

show isakmp key

Use the command **show isakmp key** to show the ISAKMP pre-shared key:

```
awplus#show isakmp key
Hostname/IP address            Key
10.2.0.10                      mytunnelkey
```

show isakmp profile

Use the command **show isakmp profile** to show ISAKMP profiles, including the default profile:

```
awplus#show isakmp profile
ISAKMP Profile: default
  Version:        IKEv2
  Authentication: PSK
  Expiry:         24h
  DPD Interval:   30s
  Transforms:
       Integrity   Encryption  DH Group
    1    SHA256      AES256      14
    2    SHA256      AES256      16
    3    SHA1        AES256      14
    4    SHA1        AES256      16
    5    SHA256      AES128      14
    6    SHA256      AES128      16
    7    SHA1        AES128      14
    8    SHA1        AES128      16
    9    SHA256      3DES        14
   10    SHA256      3DES        16
   11    SHA1        3DES        14
   12    SHA1        3DES        16
ISAKMP Profile: my_profile
  Version:        IKEv2
  Authentication: PSK
  Expiry:         24h
  DPD Interval:   30s
  Transforms:
       Integrity   Encryption  DH Group
  2     SHA1        3DES        5
```

Additional example: To show ISAKMP profile '**my_profile**' use the command:

```
awplus# show isakmp profile my_profile
```

```
awplus#show isakmp profile my_profile
ISAKMP Profile: my_profile
  Version:        IKEv2
  Authentication: PSK
  Expiry:         24h
  DPD Interval:   30s
  Transforms:
       Integrity   Encryption  DH Group
    2    SHA1        3DES        5
```

show isakmp counters

Use the command **show isakmp counters** to show ISAKMP counters:

```
awplus#show isakmp counters
Name                          Value
-------------------------------------
ikeInitRekey                  0
ikeRspRekey                   0
ikeChildSaRekey               0
ikeInInvalid                  0
ikeInInvalidSpi               0
ikeInInitReq                  0
ikeInInitRsp                  0
ikeOutInitReq                 0
ikeOutInitRsp                 0
ikeInAuthReq                  0
ikeInAuthRsp                  0
ikeOutAuthReq                 0
ikeOutAuthRsp                 0
ikeInCrChildReq               0
ikeInCrChildRsp               0
ikeOutCrChildReq              0
ikeOutCrChildRsp              0
ikeInInfoReq                  0
ikeInInfoRsp                  0
ikeOutInfoReq                 0
ikeOutInfoRsp                 0
```

show ipsec counters

Use the command **show ipsec counters** to show IPsec counters:

```
awplus#show ipsec counters
Name                          Value
-------------------------------------
InError                       0
InBufferError                 0
InHdrError                    0
InNoStates                    0
InStateProtoError             0
InStateModeError              0
InStateSeqError               0
InStateExpired                0
InStateMismatch               0
InStateInvalid                0
InTmplMismatch                0
InNoPols                      0
InPolBlock                    0
InPolError                    0
OutError                      0
OutBundleGenError             0
OutBundleCheckError           0
OutNoStates                   0
OutStateProtoError            0
OutStateModeError             0
OutStateSeqError              0
OutStateExpired               0
OutPolBlock                   0
OutPolDead                    0
OutPolError                   0
FwdHdrError                   0
```

show ipsec policy

Use the **show ipsec policy** command to show IPsec policies:

```
awplus#show ipsec policy
Traffic Selector (addresses protocol ports interface)
  Profile           Peer
0.0.0.0/0 0.0.0.0/0   tunnel1
  default           10.2.0.10
```

show ipsec profile

Use the **show ipsec profile** command to show all IPsec profiles, including the default profile:

```
awplus#show ipsec profile
 IPsec Profile: default
   Replay-window:  32
   Rekey:          Always
   Expiry:         8h
   PFS group:      disabled
   Transforms:
        Protocol  Integrity   Encryption
     1    ESP       SHA256      AES256
     2    ESP       GCM16       AES256
     3    ESP       GCM8        AES256
     4    ESP       SHA1        AES256
     5    ESP       SHA256      AES128
     6    ESP       GCM16       AES128
     7    ESP       GCM8        AES128
     8    ESP       SHA1        AES128
     9    ESP       SHA256      3DES
    10    ESP       SHA1        3DES

IPsec Profile: my_profile
  Replay-window:  32
  Expiry:         8h
  PFS group:      disabled
  Transforms:
       Protocol  Integrity   Encryption
    2    ESP       SHA1        3DES
```

show ipsec peer

Use the **show ipsec peer** command to show IPsec information on a per peer basis:

```
awplus#show ipsec peer 172.16.0.1
172.16.0.2
  IPsec
    Selectors (local:remote)
      Address:  0.0.0.0/0 : 0.0.0.0/0
      Protocol: any:any
      Port:     any:any
      Mark:     1:1
    Profile: default
    SAs:
      SPI (In:Out): ca865389:c9c7e3d3
        Selectors:  192.168.1.0/24 : 192.168.2.0/24
        Proto:      ESP
        Mode:       tunnel
        Encryption: AES256
        Integrity:  SHA256
        Expires:    28796s
  ISAKMP
    LocalID:  172.16.0.1
    RemoteID: 172.16.0.2
    SAs:
      Cookies (Initiator:Responder) 03071749781e5992:93f8457816d3d40d
        Ver:            2             Lifetime: 84569s    State: Established
        Authentication: PSK           Group:    14
        Encryption:     AES256        NATT:     no
        Integrity:      SHA256        DPD:      yes
```

Debug

The debug feature is a very powerful and flexible tool for troubleshooting issues. Comprehensive debugging is available, and multiple options can be used to enable debug for different aspects of IPsec and ISAKMP. All of the message types can be individually enabled or disabled with the **debug isakmp** command.

In this example all debug is enabled at the basic level, and CFG messages are enabled at the detailed level, then the tunnel is initiated with a ping from the remove device.

Example **debug** configuration for a tunnel initiation

```
awplus#show debugging isakmp
03:56:20 awplus IMISH[17992]: [manager@ttyS0]show debugging isakmp
ISAKMP Debugging status:
  CFG (Configuration management)                enabled
  CHD (Child SA/IPsec SA)                       enabled
  DMN (Main daemon signal handling)             enabled
  ENC (Packet encryption/decryption)            enabled
  IKE (IKE SA/ISAKMP SA)                        enabled
  JOB (Jobs queuing/processing)                 enabled
  KNL (IPsec/Networking kernel interface)       enabled
  MGR (IKE SA manager)                          enabled
  NET (IKE network communication)               enabled
awplus#debug isakmp cfg detail
03:56:23 awplus IMISH[17992]: [manager@ttyS0]debug isakmp cfg detail
```

Example **debug** configuration for a tunnel initiation

```
 awplus#show debugging isakmp
 03:56:28 awplus IMISH[17992]: [manager@ttyS0]show debugging isakmp
 ISAKMP Debugging status:
   CFG (Configuration management)                enabled  (with detail)
   CHD (Child SA/IPsec SA)                       enabled
   DMN (Main daemon signal handling)             enabled
   ENC (Packet encryption/decryption)            enabled
   IKE (IKE SA/ISAKMP SA)                        enabled
   JOB (Jobs queuing/processing)                 enabled
   KNL (IPsec/Networking kernel interface)       enabled
   MGR (IKE SA manager)                          enabled
   NET (IKE network communication)               enabled
 awplus#03:56:42 awplus IPSEC: 15[CFG] looking for an ike config for
128.0.0.1...128.0.0.2
 03:56:42 awplus IPSEC: 15[CFG]   candidate: 128.0.0.1...128.0.0.2, prio 3100
 03:56:42 awplus IPSEC: 15[CFG] found matching ike config: 128.0.0.1...128.0.0.2
with prio 3100
 03:56:42 awplus IPSEC: 15[IKE] 128.0.0.2 is initiating an IKE_SA
 03:56:42 awplus IPSEC: 15[CFG] selecting proposal:
 03:56:42 awplus IPSEC: 15[CFG]   proposal matches
 03:56:42 awplus IPSEC: 15[CFG] received proposals: IKE:AES_CBC_256/
HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_4096
 03:56:42 awplus IPSEC: 15[CFG] configured proposals: IKE:AES_CBC_256/
HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_4096
 03:56:42 awplus IPSEC: 15[CFG] selected proposal: IKE:AES_CBC_256/
HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_4096
 03:56:44 awplus IPSEC: 10[CFG] looking for peer configs matching
128.0.0.1[128.0.0.1]...128.0.0.2[128.0.0.2]
 03:56:44 awplus IPSEC: 10[CFG]  candidate "tunnel1", match: 20/20/3100 (me/other/
ike)
 03:56:44 awplus IPSEC: 10[CFG] selected peer config 'tunnel1'
 03:56:44 awplus IPSEC: 10[IKE] IKE_SA tunnel1[16] established between
128.0.0.1[128.0.0.1]...128.0.0.2[128.0.0.2]
 03:56:44 awplus IPSEC: 10[CFG] looking for a child config for 192.168.0.1/32[icmp]
0.0.0.0/0 === 192.168.0.2/32[icmp] 0.0.0.0/0
 03:56:44 awplus IPSEC: 10[CFG] proposing traffic selectors for us:
 03:56:44 awplus IPSEC: 10[CFG]  0.0.0.0/0
 03:56:44 awplus IPSEC: 10[CFG] proposing traffic selectors for other:
 03:56:44 awplus IPSEC: 10[CFG]  0.0.0.0/0
 03:56:44 awplus IPSEC: 10[CFG]   candidate "tunnel1" with prio 7+7
 03:56:44 awplus IPSEC: 10[CFG] found matching child config "tunnel1" with prio 14
 03:56:44 awplus IPSEC: 10[CFG] selecting proposal:
 03:56:44 awplus IPSEC: 10[CFG]   proposal matches
 03:56:44 awplus IPSEC: 10[CFG] received proposals: ESP:AES_CBC_256/
HMAC_SHA2_256_128/EXT_SEQ/NO_EXT_SEQ
 03:56:44 awplus IPSEC: 10[CFG] configured proposals: ESP:AES_CBC_256/
HMAC_SHA2_256_128/MODP_4096/EXT_SEQ/NO_EXT_SEQ
 03:56:44 awplus IPSEC: 10[CFG] selected proposal: ESP:AES_CBC_256/
HMAC_SHA2_256_128/EXT_SEQ
 03:56:44 awplus IPSEC: 10[CFG] selecting traffic selectors for us:
 03:56:44 awplus IPSEC: 10[CFG]  config: 0.0.0.0/0, received: 192.168.0.1/32[icmp]
=> match: 192.168.0.1/32[icmp]
 03:56:44 awplus IPSEC: 10[CFG]  config: 0.0.0.0/0, received: 0.0.0.0/0 => match:
0.0.0.0/0
 03:56:44 awplus IPSEC: 10[CFG] selecting traffic selectors for other:
 03:56:44 awplus IPSEC: 10[CFG]  config: 0.0.0.0/0, received: 192.168.0.2/32[icmp]
=> match: 192.168.0.2/32[icmp]
 03:56:44 awplus IPSEC: 10[CFG]  config: 0.0.0.0/0, received: 0.0.0.0/0 => match:
0.0.0.0/0
 03:56:44 awplus IPSEC: 10[IKE] CHILD_SA tunnel1{21} established with SPIs
c1ec4386_i c1f9cd8f_o and TS 0.0.0.0/0 === 0.0.0.0/0
```

In this example a lot of detailed configuration debug is printed with the IPsec CFG messages. To get complete debug, use the command **debug isakmp detail**. This can be quite verbose, so you can disable all ISAKMP using the command **undebug isakmp**.

show isakmp peer

The output of the **show isakmp peer** command is quite useful for any IPsec configurations. The following is an example of the command entered into a **Main Office** site device:

Example **show isakmp peer** command

```
awplus#show isakmp peer
Peer                        Profile (* incomplete)                Key
-------------------------------------------------------------------
Remote_Site_1               phase1                                PSK
```

This command shows the ISAKMP profile and key status for any configured ISAKMP peers.

To display the interface and counter information for a specific virtual tunnel interface, use the following command:

```
awplus# show interface <tunnel-instance>
```

```
awplus#show interface tunnel2
Interface tunnel2
  Link is UP, administrative state is UP
  Hardware is Tunnel
  IPv4 address 192.168.1.1/24 point-to-point 192.168.1.255
  index 21 metric 1 mtu 1438
  <UP,POINT-TO-POINT,RUNNING,MULTICAST>
  SNMP link-status traps: Disabled
  Tunnel source 10.1.0.10, destination 10.2.0.10
  Tunnel name local 10.1.0.10, remote 10.2.0.10
  Tunnel traffic selectors (ID, local, remote)
    1    192.168.2.0/24             192.168.3.0/24
    2    0.0.0.0/0                  192.168.10.0/24
  Tunnel protocol/transport ipsec ipv4, key disabled, sequencing disabled
  Checksumming of packets disabled, path MTU discovery disabled
  Tunnel protection via IPsec (profile "default")
    input packets 11, bytes 924, dropped 0, multicast packets 0
    output packets 0, bytes 0, multicast packets 0 broadcast packets 0
  Time since last state change: 0 days 03:23:10
```