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** or later.

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

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

These documents are available from the above links on our website at alliedtelesis.com.Feature support may change in later software versions. For the latest information, see the above documents.

Contents

IPsec Introduction

What does IPsec do?

IPsec provides the following security services for traffic at 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 site-to-site VPN. A site-to-site VPN is used to connect 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 Virtual Tunnel Interface (VTI). The AR-Series Firewalls 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 and 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 types of tunnels that 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 the AR-Series Firewalls to inter-operate easily with a broad range of other vendors VPN equipment. No specific configuration is required to enable the AR-Series Firewall 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:

Table 2: Default IPsec profiles

ATTRIBUTE	PROTOCOL	ENCRYPTION (ALL CBC)	INTEGRITY (ALL HMAC)
Transform 1	ESP	AES256	SHA256
Transform 2	ESP	AES256	SHA1
Transform 3	ESP	AES128	SHA256
Transform 4	ESP	AES128	SHA1
Transform 5	ESP	3DES	SHA256
Transform 6	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, 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 cases are:

When peering to a device that may not support up to date secure cryptographic algorithms that are outside the range included in the AR-Series Firewalls default profiles. 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-Helman 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, the AR-Series Firewalls 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 AR-Series Firewall 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-Helman 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
- 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.

Traffic selectors

By default AlliedWare Plus uses a route based VPN, where the VPN is terminated via a Virtual Tunnel Interface (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.

Step-by-step Configuration

How to configure 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:

Table 3: How to configure basic IPsec protection	
Step 1. Configure the pre-shared key for ISAKMP	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `crypto isakmp key <key> address <peer-address>`	Enter the pre-shared key and peer IP address. The key is associated with a peer address.
Step 2. Set up the tunnel to apply IPsec protection	
`awplus(config)#` `interface tunnel <0-255>`	Enter Interface mode and specify a tunnel name. For example tunnel1.
`awplus(config-if)#` `ip address <IP-address>`	Enter the IP address for the tunnel interface.
`awplus(config-if)#` `tunnel source <interface-name>`	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)#` `tunnel destination <IP address>`	Enter the IP address for the peer tunnel destination.
`awplus(config-if)#` `tunnel mode <mode>`	Enter the mode, where mode can be one of: ■ IPsec IPv4 ■ IPsec IPv6 ■ L2TP v3 ■ L2TP v3 IPv6 ■ GRE ■ GRE IPv6
Step 3. Apply IPsec protection to traffic in the tunnel	
`awplus(config-if)#` `tunnel protection ipsec`	Enter this command to apply IPsec protection to traffic in the tunnel.
`awplus(config-if)#` `exit`	Exit to Configuration mode.
Step 4. Configure routes to the IP subnets at the receiving end of the tunnel	
`awplus(config)#` `ip route <far-end-subnet> <tunnel-name>`	Enter the far end subnet IP address and the tunnel name.

How to use Custom Profiles

The configuration steps to use custom profiles are:

- The profiles are defined and named
- Global parameters for the profiles are configured
- A list of transforms is added to each profile
- An ISAKMP profile is associated with one or more peers
- An IPsec profile is applied to a tunnel

ISAKMP profiles

Follow these steps to configure your custom profiles for ISAKMP:

Table 4: How to configure ISAKMP profiles	
Step 1. Define and name profiles	
`awplus#` `configure terminal`	Enter Global Configuration mode
`awplus(config)#` `crypto isakmp profile <profile-name>`	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.
Step 2. Set up global parameters (optional)	
`awplus(config-isakmp-profile)#` `lifetime <lifetime>`	Enter the lifetime in seconds. This is optional and the default is 86400 seconds (24 hours).
`awplus(config-isakmp-profile)#` `version {1 mode {aggressive|main|2}`	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)#` `dpd-interval <interval>`	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>`	Enter the wait time in seconds. The default is 150 seconds. DPD timeout defines the timeout interval after which all connections to a peer are deleted in case of inactivity. This only applies to IKEv1. In IKEv2 the default retransmission timeout applies as every exchange is used to detect dead peers.

Step 3. Add transforms	
`awplus(config-isakmp-profile)#` `transform <1-255> integrity [sha1|sha256|sha512] encryption [3des|aes128|aes192|aes256] group [2|5|14|15|16|18`	Specify the following: ■ transform priority (1 is the highest) ■ integrity (Secure Hash Standard) ■ encryption (Advanced Encryption Standard or 3DES) ■ Diffie-helman group.
Step 4. Associate with a peer	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `crypto isakmp peer {dynamic|address {<ipv4-addr>|<ipv6-addr>}} profile <profile_name>`	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) ■ profile-name

IPsec profiles

Follow these steps to configure your custom profiles for IPsec:

Table 5: How to configuration IPsec profiles	
Step 1. Define and name profiles	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `crypto ipsec profile <profile-name>`	Enter the custom IPsec 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.
Step 2. Set up global parameters (optional)	
`awplus(config-ipsec-profile)#` `lifetime seconds <lifetime>`	Enter the lifetime in seconds. The default is 28800 seconds (8 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-ipsec-profile)#` `pfs <2|5|14|15|16|18>`	Enter the pfs (Perfect Forward Security). The numbers represent the diffie-helman group. PFS is disabled by default.
Step 3. Add transforms	
`awplus(config-ipsec-profile)#` `transform <1-255> protocol esp integrity [sha1|sha256|sha512] encryption [3des|aes128|aes192|aes256]`	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).

Step 4. **Associate with a tunnel**	
`awplus(config)#` `interface tunnel <0-255>`	Enter Interface mode and specify a tunnel name. For example tunnel1.
`awplus(config-if)#` `tunnel protection ipsec {profile <profile-name>}`	Enter your custom profile name. By default IPsec protection for packets encapsulated by tunnel is disabled.

How to use traffic selectors

The commands for selecting the traffic to be associated with different IPsec SAs are entered in interface mode for the tunnel being protected. There are separate commands to match the source address and the 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 these commands to set up your traffic selectors:

Table 6: How to set up traffic selectors	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `interface tunnel <0-255>`	Enter Interface mode and specify a tunnel name. For example tunnel1.
`awplus(config-if)#` `tunnel local selector {ID-number} <address-range>`	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 remote 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.

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 name that is embedded in the packets, that initiate the connection. The commands to do this are entered in Interface mode for the tunnel being protected.

A peer that needs to identify itself configures a **local** name:

Table 7: Set up the local peer	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `interface tunnel <0-255>`	Enter interface mode and specify a tunnel interface index identifier (from 0-255). By default no tunnel interfaces exist. For example tunnel1.
`awplus(config-if)#` `tunnel local name <local-name>`	Enter the local tunnel name that is sent in IPsec setup packets.

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

Table 8: Set up the remote peer	
`awplus#` `configure terminal`	Enter Global Configuration mode.
`awplus(config)#` `interface tunnel <0-255>`	Enter interface mode and specify a tunnel interface index identifier (from 0-255). By default no tunnel interfaces exist.
`awplus(config-if)#` `tunnel remote name <name-expected-to-be_received-in-ipsec-connections>`	Enter the remote tunnel name that is expected to be received in IPsec setup packets.

Configuration Examples

Example 1: An IPsec tunnel between two AR-Series Firewalls

This example shows the step-by-step instructions to configure an IPsec tunnel between two AR-Series Firewalls. It assumes that IP has been configured correctly 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 9: 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 AR-Series Firewalls

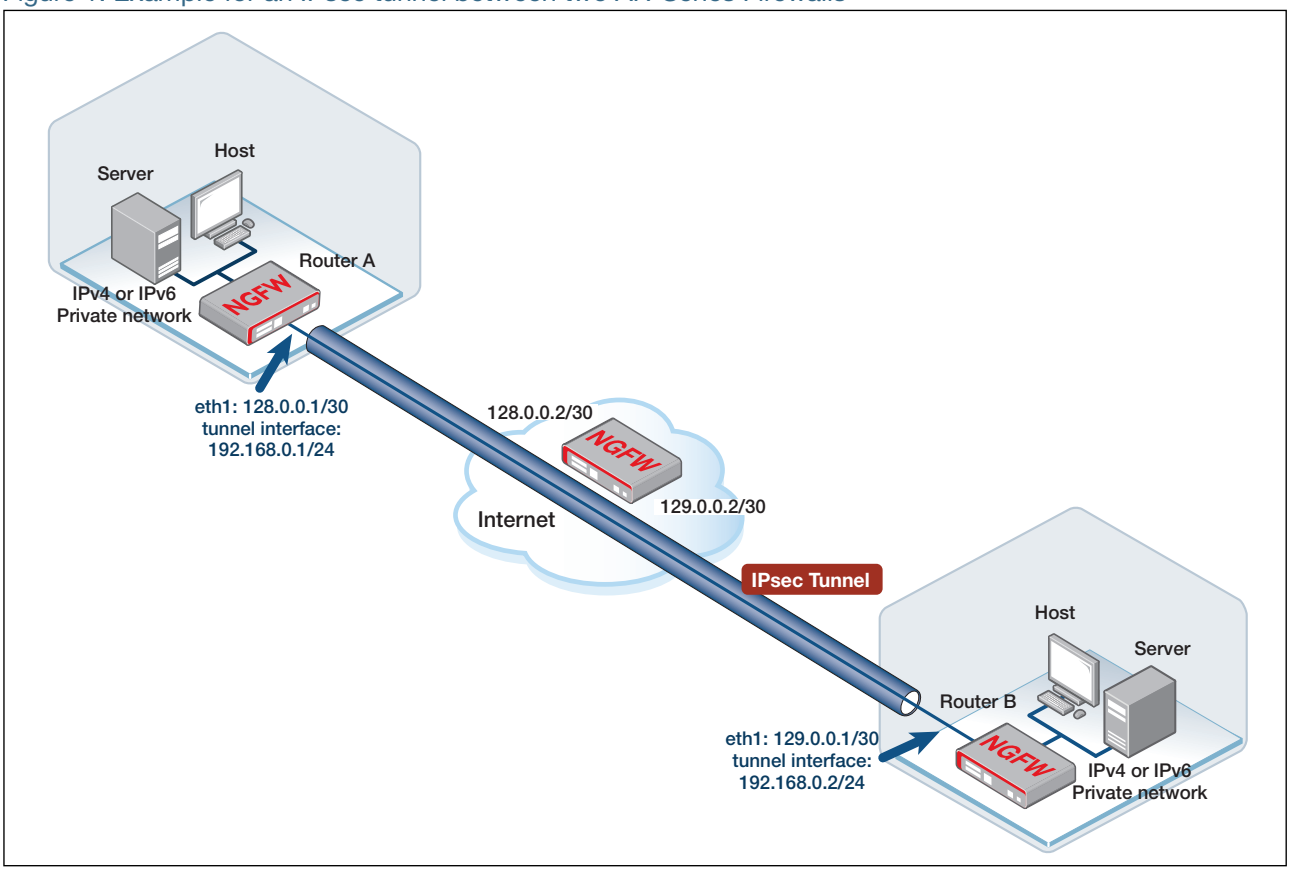


Table 10: How to configure an IPsec tunnel between two AR-Series Firewalls

Step 1. Configure Device A	
`awplus#` `configure terminal`	Enter the Global Configuration mode.
`awplus(config)#` `interface eth1`	Enter the Interface Configuration mode.
`awplus(config-if)#` `ip address 128.0.0.1/30`	To assign an IP address for interface eth1.
`awplus(config-if)#` `exit`	Exit the Interface Configuration mode and enter the Global Configuration mode.
`awplus(config)#` `interface tunnel1`	Create virtual tunnel called tunnel1.
`awplus(config-if)#` `ip address 192.168.0.1/24`	Assign an IP address to tunnel1.
`awplus(config-if)#` `tunnel source eth1`	Designate the interface or IP address that will be used as the source IP of the tunnel.
`awplus(config-if)#` `tunnel destination 129.0.0.1`	Designate the tunnel destination address, which is the IP address of interface eth1 on Device B.
`awplus(config-if)#` `tunnel mode IPsec ipv4`	Specify the tunnel mode.
`awplus(config-if)#` `tunnel protection IPsec`	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.
Step 2. Configure Device B	
`awplus#` `configure terminal`	Enter the Global Configuration mode.
`awplus(config)#` `interface eth1`	Enter the Interface Configuration mode.
`awplus(config-if)#` `ip address 129.0.0.1/30`	To assign an IP address for interface eth1.
`awplus(config-if)#` `exit`	Exit the Interface Configuration mode and enter the Global Configuration mode.
`awplus(config)#` `interface tunnel1`	Create virtual tunnel called tunnel1.
`awplus(config-if)#` `ip address 192.168.0.2/24`	Assign an IP address to tunnel1.

Table 10: How to configure an IPsec tunnel between two AR-Series Firewalls

`awplus(config-if)#` `tunnel source eth1`	Designate the interface whose IP address will be used as the source IP of the tunnel.
`awplus(config-if)#` `tunnel destination 128.0.0.1`	Designate the tunnel destination address, which is the IP address of interface eth1 on Device A.
`awplus(config-if)#` `tunnel mode IPsec ipv4`	Specify the tunnel mode.
`awplus(config-if)#` `tunnel protection IPsec`	To securely route packets through the tunnel, you need to use the **tunnel protection IPsec** command to encrypt and authenticate its packets.
Step 3. Configure authentication key on Device A	
`awplus#` `configure terminal`	Enter the Global Configuration mode.
`awplus(config)#` `crypto isakmp key tunnelkey address 129.0.0.1`	Enter the tunnel key tunnelkey.
Step 4. Configure authentication key on Device B	
`awplus#` `configure terminal`	Enter the Global Configuration mode.
`awplus(config)#` `crypto isakmp key tunnelkey address 128.0.0.1`	Enter the tunnel key tunnelkey.
Step 5. Verify the configuration	
`awplus#` `ping 192.168.0.2`	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. **Note:** Note 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
!
```

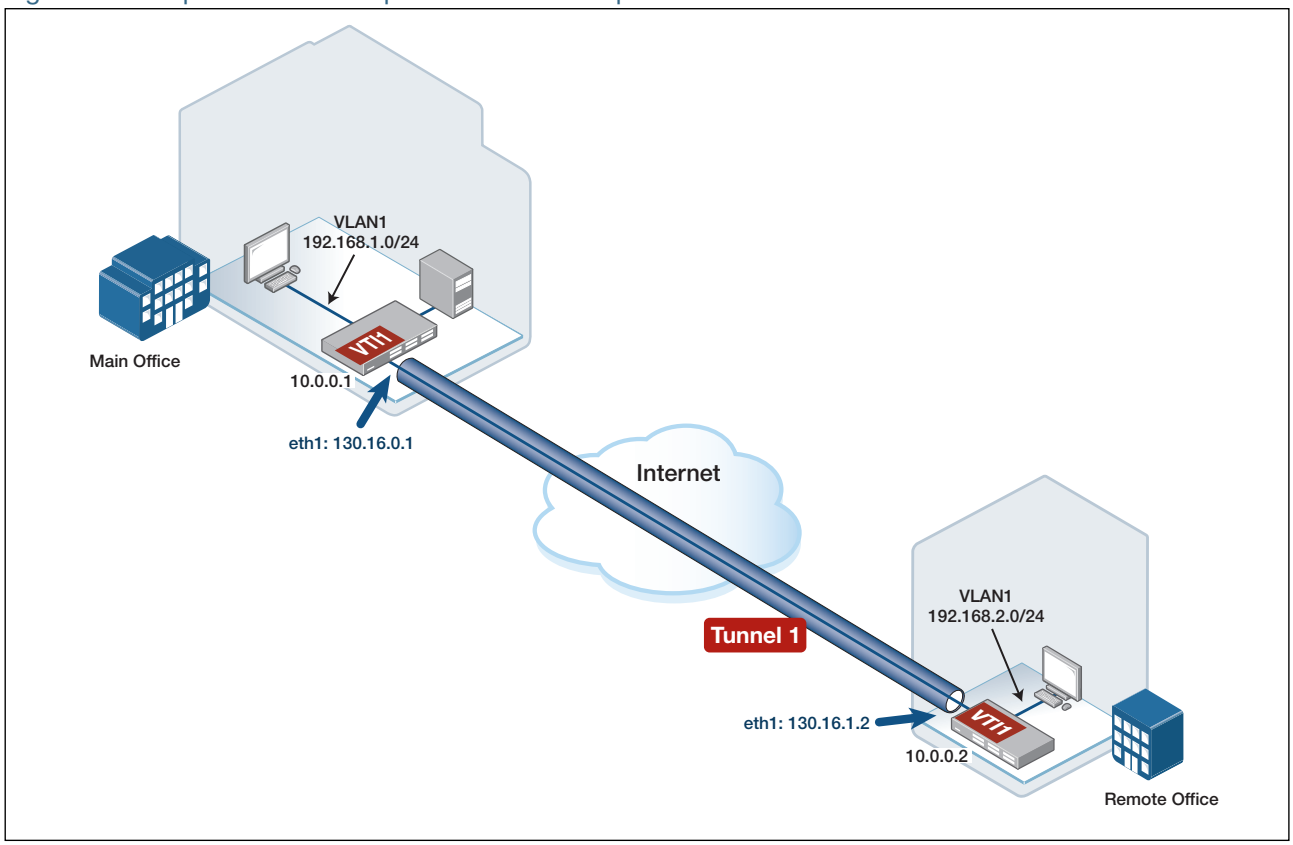
Example 3: A Custom profile with a PFS option

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 phase1
 transform 1 integrity SHA256 encryption AES256 group 5
 version 1 mode main
!
crypto ipsec profile phase2
 transform 1 protocol esp integrity SHA256 encryption AES256
 pfs 5
!
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
!
```

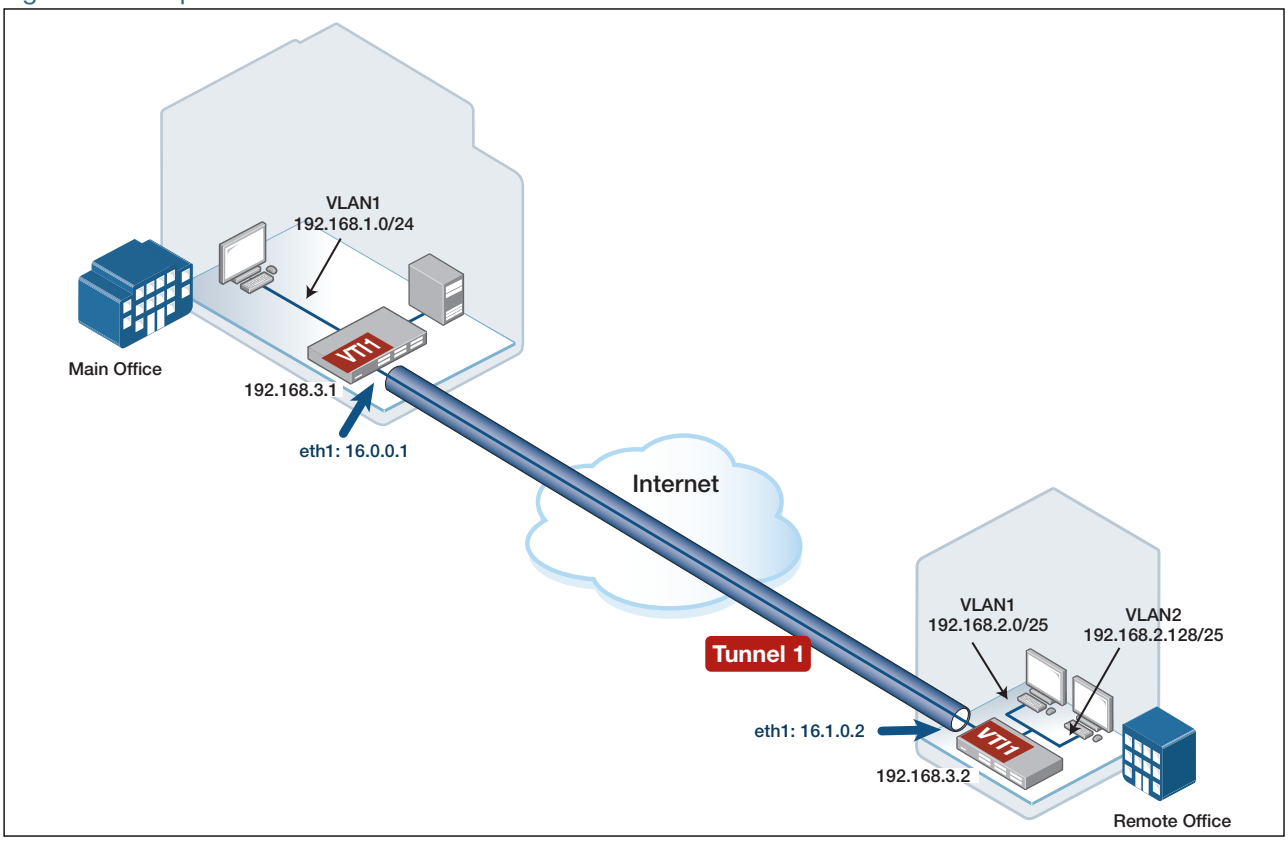
Example 4: Traffic selectors

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

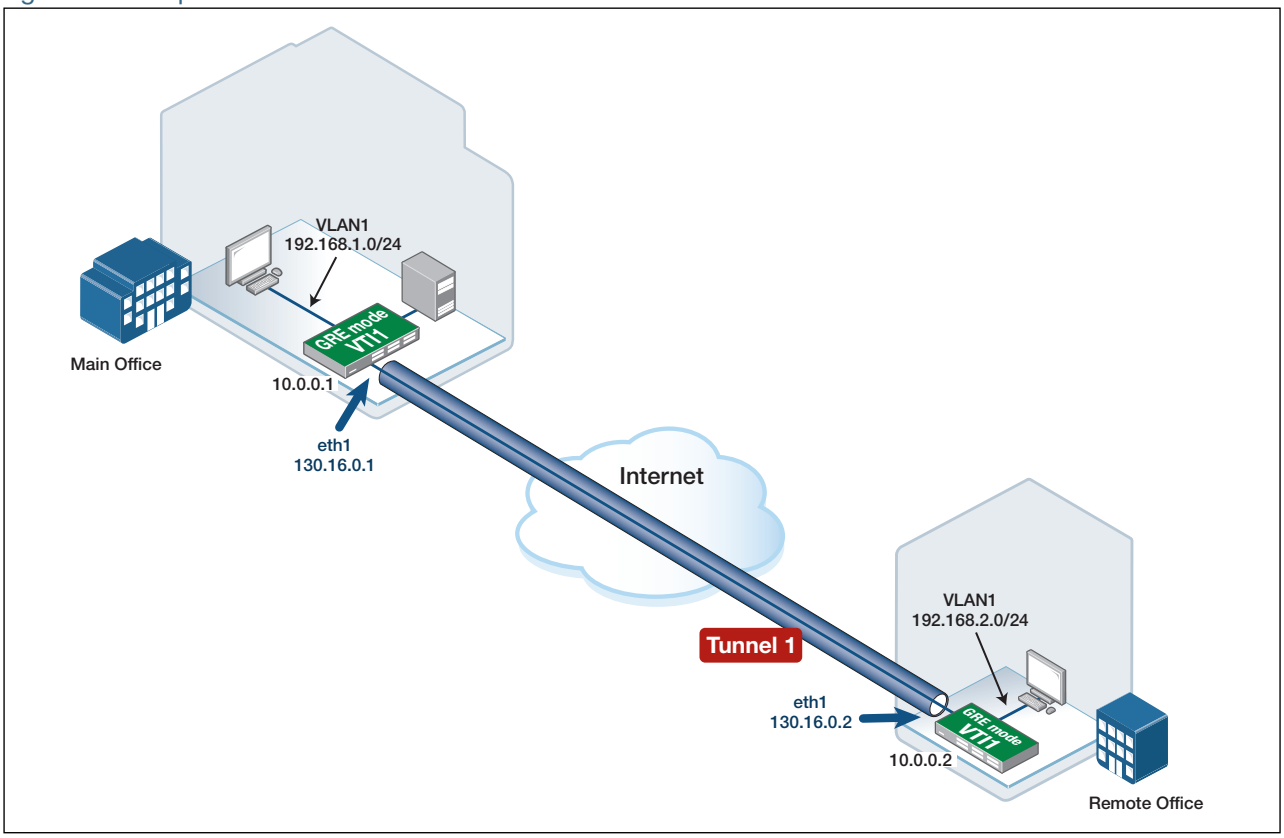
Example **Main Office** configuration for Traffic Selectors

```
!
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 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

The AR-Series Firewalls 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



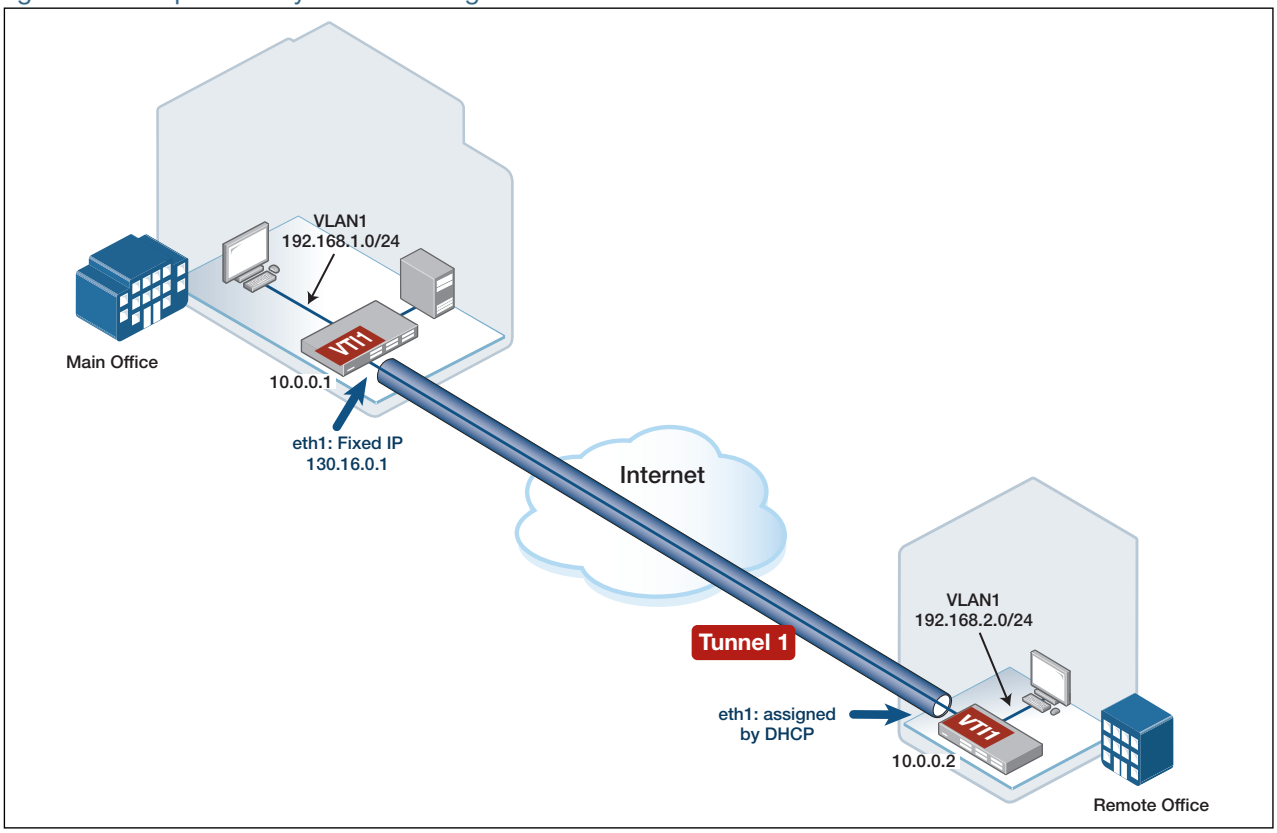
Example **Main Office** configuration for IPsec over GRE

```
!
crypto ipsec profile phase1
 transform 1 integrity SHA256 encryption AES256 group 14
 version 2
!
crypto ipsec profile phase2
 transform 1 protocol esp integrity SHA256 encryption AES256
!
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
 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**) 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 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 IPs, 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.

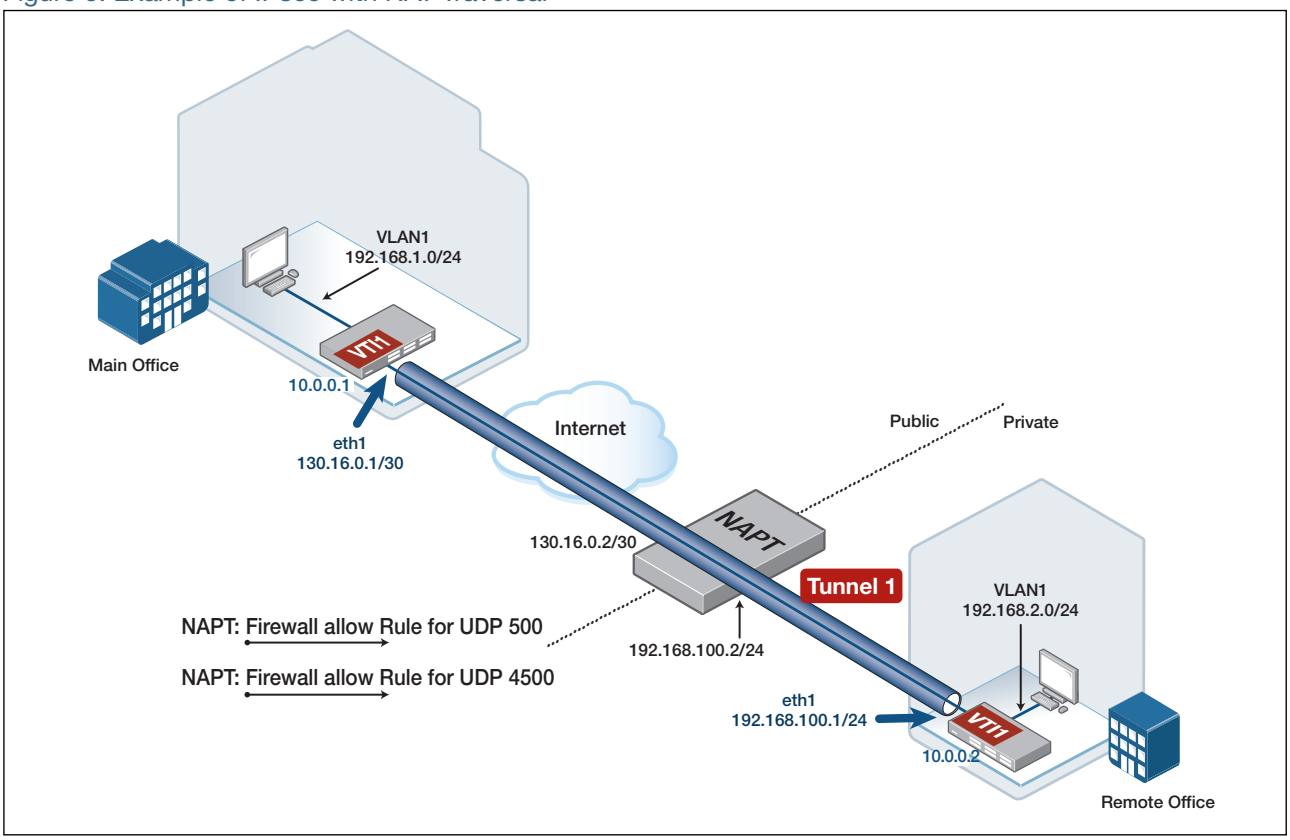
If an intermediate NAPT device is detected, NAT-T will change the UDP port numbers used by ISAKMP from UDP 500, to become UDP 4500, for all subsequent ISAKMP communications, and the ISAKMP SA is then formed.

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
!
```

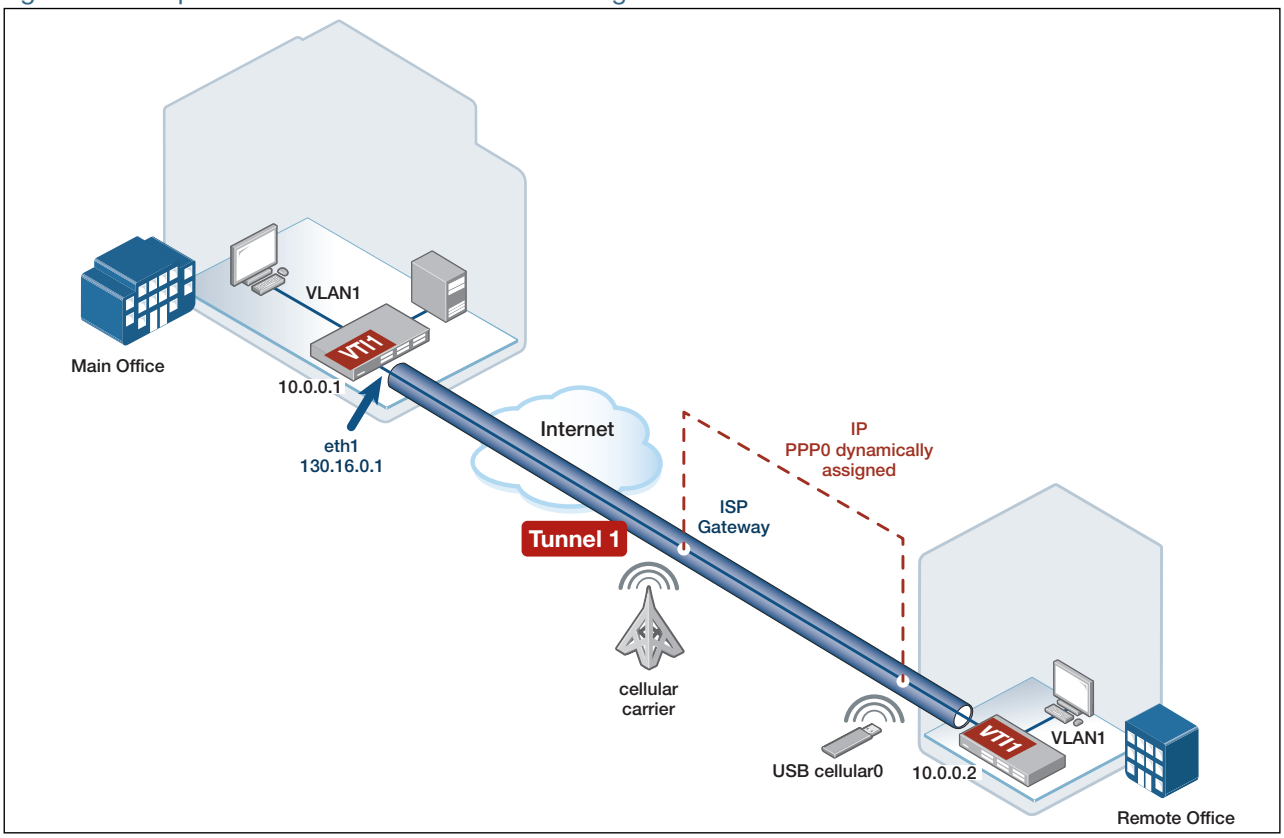
Example **Remote Office** configuration with NAT-T

```
!
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: A VPN with one end connecting over a Cellular interface

In this example, 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.

Figure 7: Example of a VPN with one end connecting over a cellular interface



The main Site AR-Series Firewall 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 https://en.wikipedia.org/wiki/Access_Point_Name.

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
!
```

Example 9: IPsec pairing to main site legacy device with firewall and dynamically assigned IP address

This example shows how to configure an AR-Series 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 AR-Series 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**.

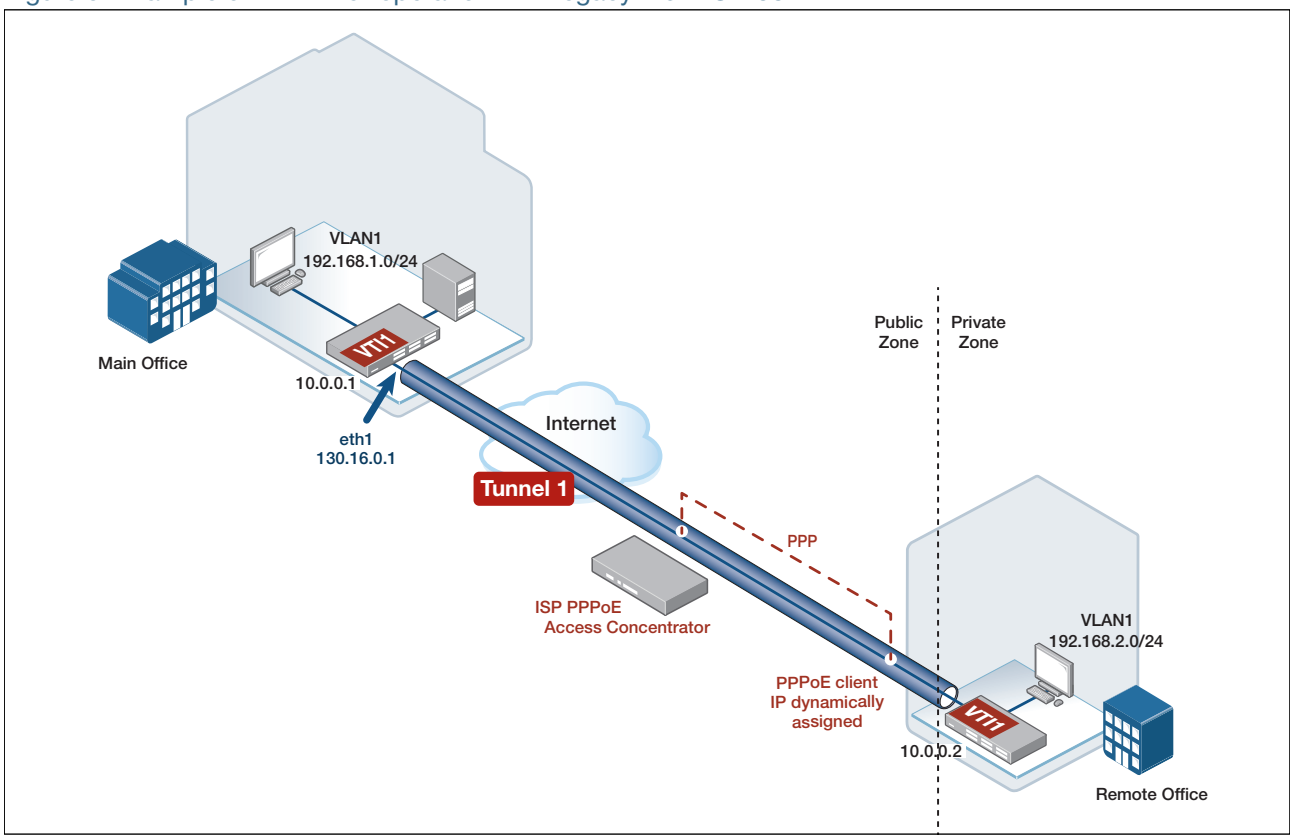
The PPPoE client WAN interface IP address is dynamically assigned. The Main Office router has fixed IP address on its WAN interface.

The AR-Series Firewall PPPoE WAN interface is located in the firewall Public zone. The Main and Remote Office LAN networks, and also VPN traffic terminated at the Virtual Tunnel Interface (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 8: 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 10: A 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 AR-Series Firewalls each have two VPNs configured, a primary VPN and a backup VPN. Each VPN is terminated by a virtual tunnel interface. 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 virtual tunnel interface (tunnel10, sourced via eth1) terminating the primary IPsec VPN.

A static route is configured via the virtual tunnel interface (tunnel20, sourced via eth2) terminating the backup IPsec VPN.

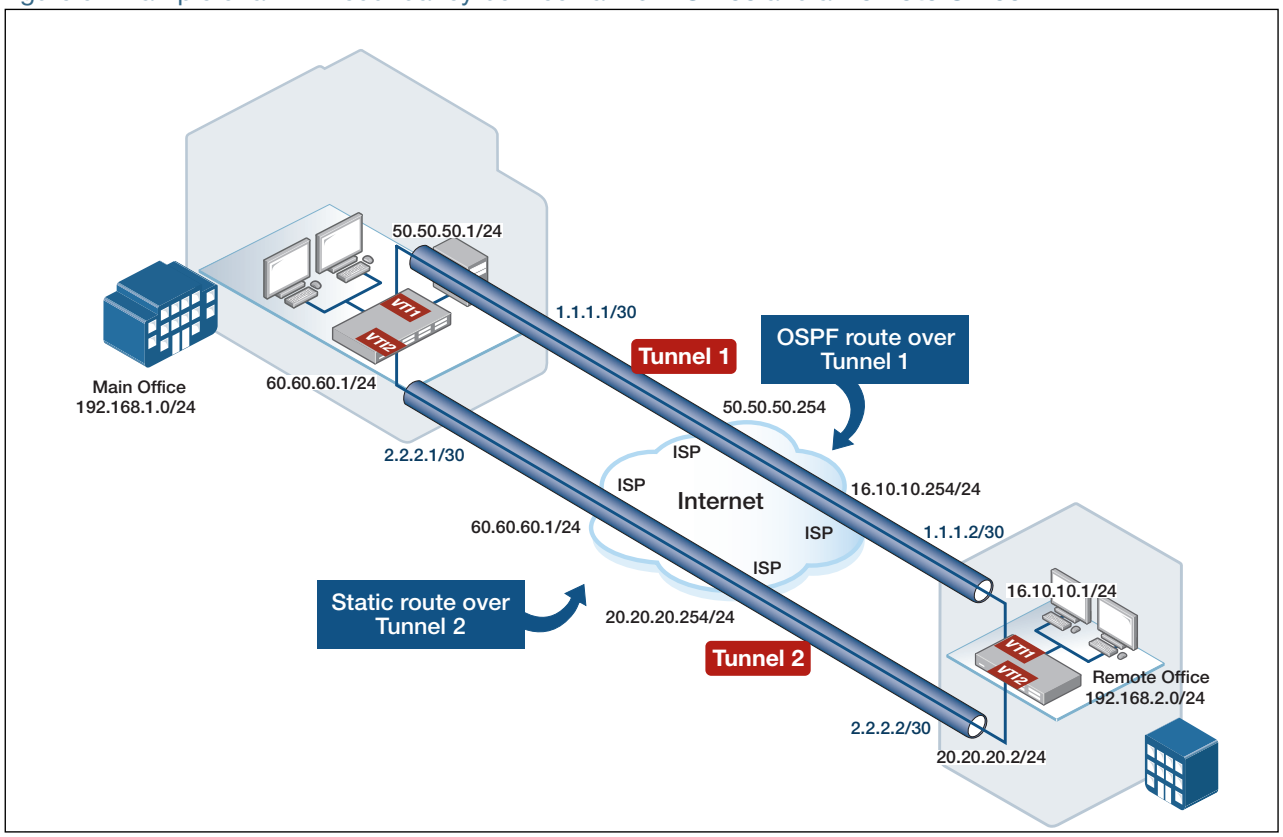
The static route (via tunnel20) is configured with a high metric, so the route learned by OSPF will be selected as the preferred route for traffic between the private LANs.

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 AR-Series Firewalls. This includes multi-zone firewall and associated NAT configuration, static and dynamic (OSPF) routing configuration, and VPN configuration.

Figure 9: 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
!
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
!
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
!
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
!
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 11: Diagnostics

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.

North America Headquarters | 19800 North Creek Parkway | Suite 100 | Bothell | WA 98011 | USA | T: +1 800 424 4284 | F: +1 425 481 3895
Asia-Pacific Headquarters | 11 Tai Seng Link | Singapore | 534182 | T: +65 6383 3832 | F: +65 6383 3830
EMEA & CSA Operations | Incheonweg 7 | 1437 EK Rozenburg | The Netherlands | T: +31 20 7950020 | F: +31 20 7950021

alliedtelesis.com