Application Note

IPsec Tunnel

Advantech Czech s.r.o., Sokolska 71, 562 04 Usti nad Orlici, Czech Republic
Document No. APP-0006-EN, revised on April 27, 2026.

Used symbols

Important

Important — Indicates a risk to personal safety or potential damage to the router. Follow these instructions precisely to prevent injury or equipment damage.

Warning

Warning — Highlights conditions that may cause malfunction, loss of data, or unexpected behavior in specific situations. Read carefully before proceeding.

Info

Info — Provides helpful tips, context, or references that improve understanding but are not strictly required to complete the task.

Contents

List of Figures

List of Tables

1. IPsec and its Protocols

IPsec (Internet Protocol Security) is a security extension of the IP protocol based on authentication and encryption of every IP datagram. Within the OSI architecture, it operates at the network layer, which means that IPsec provides security for any type of network traffic, regardless of the application.

IPsec addresses these two major security concerns:

- **Authentication** – Verifies the origin and integrity of received data, confirming that a packet was sent by the claimed sender and has not been tampered with (Phase I, IKE phase, Main mode). When using a Pre-Shared Key (PSK), this phase concludes with the key exchange.
- **Encryption** – Both sides negotiate the encryption method in advance. In transport mode, the packet payload is encrypted while the original IP header is preserved; in tunnel mode, the entire original packet is encrypted and a new IP header is added (Phase II, IPsec phase, Quick mode). This phase concludes with the establishment of the tunnel.

Info

IPsec consists of two core protocols – *Authentication Header (AH)* and *Encapsulating Security Payload (ESP)*. The AH protocol is not supported in the router's web interface configuration. Using both protocols simultaneously is often unsupported by intermediate gateways on the Internet.

Part of IPsec is also the *IKE (Internet Key Exchange)* protocol, which handles key management. IKE establishes logical channels called *Security Associations (SA)*. These channels are always unidirectional; therefore, two separate channels (SAs) are required for full-duplex communication. IKE also supports automatic generation and renewal of encryption keys.

1.1 Encapsulating Security Payload (ESP)

The Encapsulating Security Payload (ESP) protocol ensures the confidentiality of transmitted data by encrypting packets, and optionally provides data origin authentication, data integrity verification, and replay protection. As with the Authentication Header (AH) protocol, an additional header is attached to the IP packet. This header contains the security parameters, followed by the encrypted payload. However, the outer IP header is not protected and its integrity is not guaranteed.

When both encryption and authentication are required, the receiving side first verifies the authentication of the packet and, only if that step succeeds, proceeds with decryption. This approach reduces processing overhead and limits exposure to denial-of-service attacks.

1.1.1 Usage of Encapsulating Security Payload Protocol

The ESP protocol can be used in two modes – *transport mode* or *tunnel mode*. In transport mode, the ESP header is inserted after the original IP header. The encrypted payload of the original datagram is followed by the ESP trailer and optional authentication data. Transport mode has lower processing overhead than tunnel mode, but provides less comprehensive data protection.

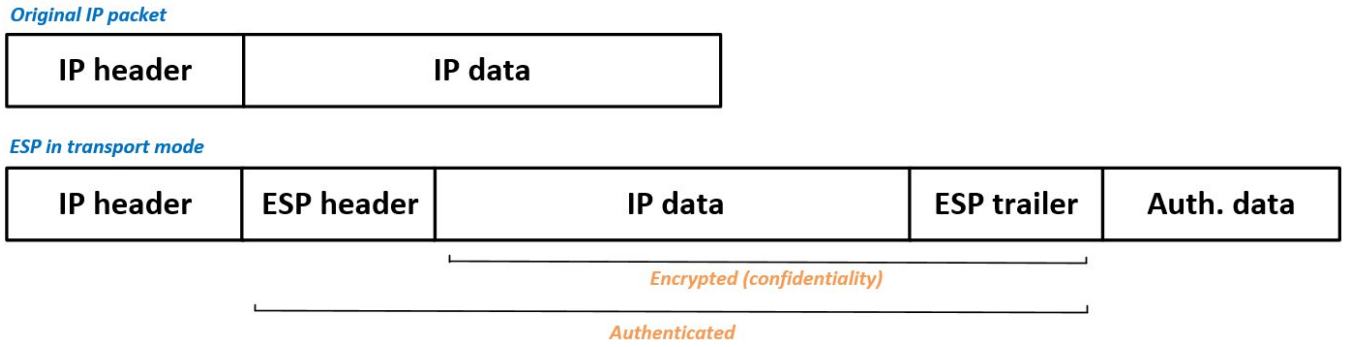

Figure 1: ESP – transport mode

In tunnel mode, a new IP header is prepended to the ESP header, which is followed by the entire original datagram as the encrypted payload, and terminated by the ESP trailer and optional authentication data. This mode fully protects the original datagram when both encryption and authentication are used. The main advantage of tunnel mode is complete protection of the encapsulated IP datagram, and it also enables the use of private IP addresses.

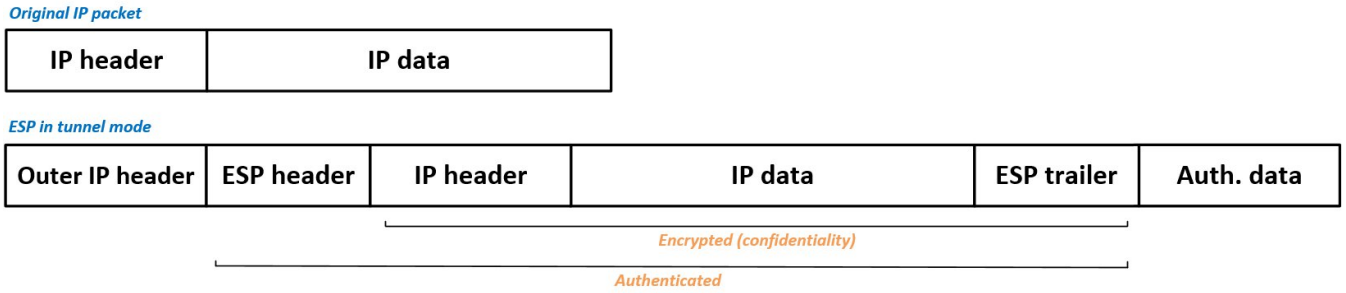

Figure 2: ESP – tunnel mode

2. IPsec Tunnel Configuration

The IPsec tunnel function allows you to create a secure connection between two separate LAN networks. This router family allows you to create up to four IPsec tunnels.

To open the IPsec tunnel configuration page, click *IPsec* in the *Configuration* section of the main menu. The menu item will expand, and you will see separate configuration pages: *1st Tunnel*, *2nd Tunnel*, *3rd Tunnel*, and *4th Tunnel*.

Both **policy-based** and **route-based** VPN approaches are supported. For a comparison of the two modes, see Section 2.1. For route-based configuration scenarios, see Section 2.2.

IPv4 and IPv6 tunnels are supported (**dual stack**). You can transport IPv6 traffic through an IPv4 tunnel and vice versa. For different IPsec authentication scenarios, see Chapter 2.3.

Warning

- To encrypt data between the local and remote subnets, specify the appropriate values in the subnet fields on both routers. To encrypt only the data stream between the routers, leave the local and remote subnet fields blank.
- If you specify protocol and port information in the *Local Protocol/Port* field, the router will encapsulate only the packets matching those settings.
- For an optimal and secure setup, we recommend following the instructions on the Security Recommendations page of the *strongSwan* website.

Info

- The dual stack IPsec tunnels are not supported by routers of v2 product line.
- The *FRR* Router App is an internet routing protocol suite for Advantech routers. It includes protocol daemons for BGP, IS-IS, LDP, OSPF, PIM, and RIP.

2.1 Policy-based vs. Route-based VPN

The router supports two VPN modes, selectable via the *Type* field on the IPsec configuration page. The key differences are summarized in the table below.

Feature	Policy-based	Route-based
Traffic selection	Subnet pairs defined in *Local Subnet* and *Remote Subnet* fields	Routing table entries
Virtual interface	None	`ipsecX` interface is created
Traffic inspection	Not possible on tunnel traffic	Possible using `tcpdump -i ipsecX`
Dynamic routing	Not supported	Supported (e.g., FRR/BGP, FRR/OSPF)
Multiple clients	Limited	Fully supported
Cisco FlexVPN	Not supported	Supported
Configuration complexity	Lower	Higher

Table 1: Policy-based vs. route-based IPsec comparison

In **policy-based** mode, the router encrypts traffic based on configured security policies defined by the subnet pairs in *Local Subnet* and *Remote Subnet*. No virtual interface is created — the kernel's policy engine

handles encapsulation transparently. This is the simpler approach and is suitable for most standard site-to-site VPN deployments. In terms of connectivity, it is equivalent to the route-based *Enabled Installing Routes* scenario described in Section 2.2.

In **route-based** mode, a virtual `ipsecX` interface is created for each tunnel. Traffic is routed into the tunnel using standard routing rules, which enables dynamic routing protocols and more flexible topologies. The available route-based scenarios are described in Section 2.2.

Info

When using policy-based mode, if neither *Local Subnet* nor *Remote Subnet* is configured, only router-to-router traffic is encrypted — no LAN-to-LAN traffic will pass through the tunnel.

2.2 Configuration Scenarios

The following scenarios describe the most common VPN topologies supported by Advantech routers. The examples use route-based mode, but — with the exception of scenarios 2 and 3 — they are equally applicable to policy-based mode.

1. **Enabled Installing Routes**
   • Remote and local subnets are used as traffic selectors (routes).
   • This results in the same outcome as a policy-based VPN.
   • A benefit of this approach is the ability to inspect unencrypted traffic on the `ipsecX` interface using a tool like `tcpdump -i ipsecX`.
   • Set *Install Routes* to *yes*.
2. **Static Routes** (route-based only)
   • Routes are installed statically by an application as soon as the IPsec tunnel is established.
   • An application like FRR/STATICD can be used for this purpose.
   • Set *Install Routes* to *no*.
3. **Dynamic Routing** (route-based only)
   • Routes are installed dynamically by a routing protocol application, such as FRR/BGP or FRR/OSPF.
   • Set *Install Routes* to *no*.
4. **Multiple Clients**
   • This allows for a VPN network with multiple clients. One router acts as the server and assigns IP addresses to all clients.
   • The server has *Remote Virtual Network* and *Remote Virtual Mask* configured, while clients use the *Local Virtual Address* setting.
   • Set *Install Routes* to *yes*.

2.3 IPsec Authentication Scenarios

Four basic authentication options are supported:

1. **Pre-shared Key**
   - Set *Authenticate Mode* to *pre-shared key*.
   - Enter the shared key into the *Pre-shared Key* field.
2. **Public Key**
   - Set *Authenticate Mode* to *X.509 certificate*.
   - Enter the public key into the *Local Certificate / PubKey* field.
   - A CA certificate is not required.

3. **Peer Certificate**
    - Set *Authenticate Mode* to *X.509 certificate*.
    - Enter the remote key into the *Remote Certificate / PubKey* field. Users with this certificate will be allowed.
    - A CA certificate is not required.
4. **CA Certificate**
    - Set *Authenticate Mode* to *X.509 certificate*.
    - Enter the CA certificate(s) into the *CA Certificate* field. Any certificate signed by the specified CA will be accepted.
    - The remote certificate itself is not required.

Notes:

- The Peer and CA Certificate modes can be used simultaneously; authentication can be performed by either method.
- The *Local ID* is significant. When using certificate authentication, the IKE identity must be contained in the certificate, either as its subject or as a `subjectAltName`.

2.4 Configuration Items Description

The IPsec configuration GUI is shown in Figure 3, and all items are described in the tables below.

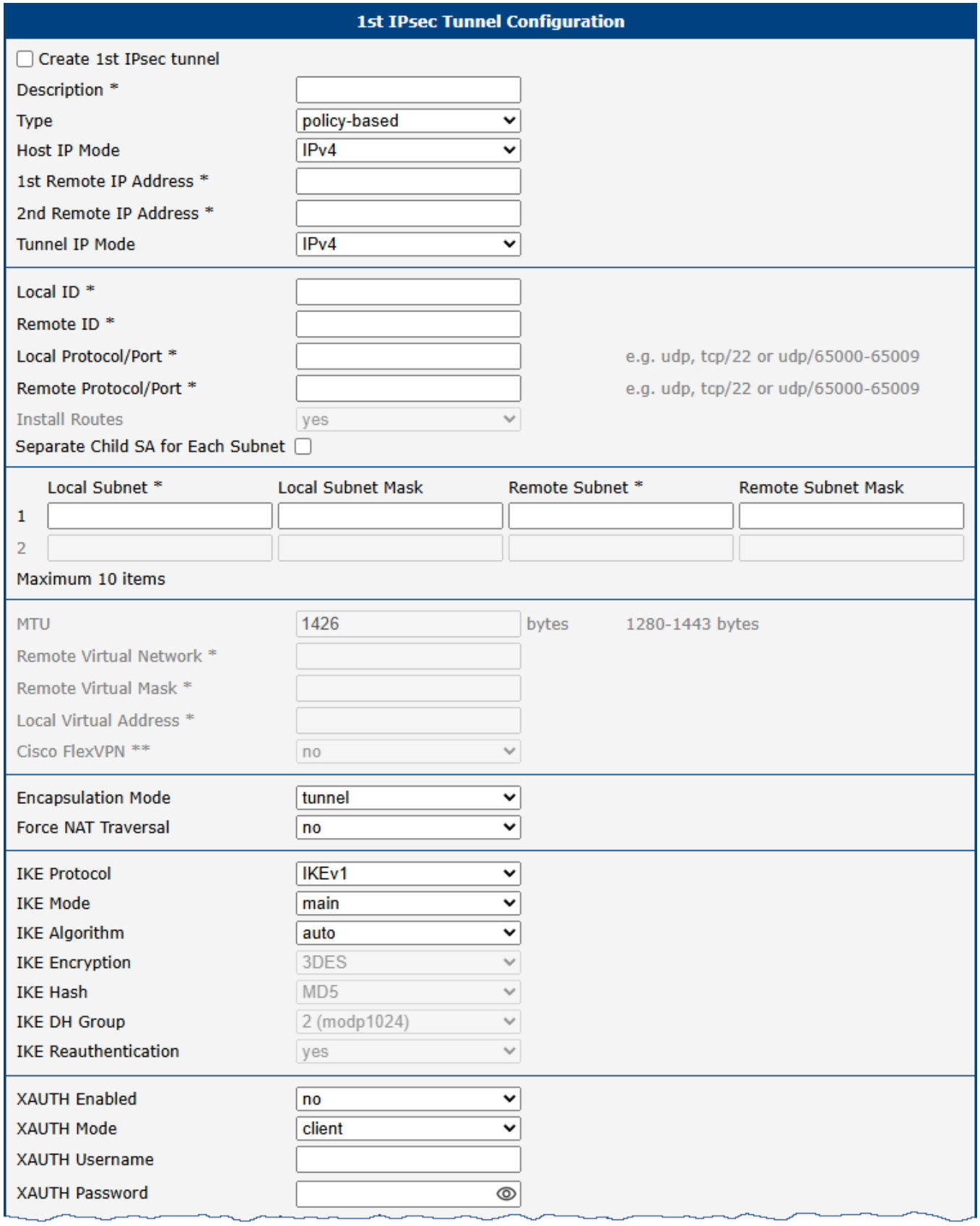

Figure 3: IPsec tunnels configuration page – part 1

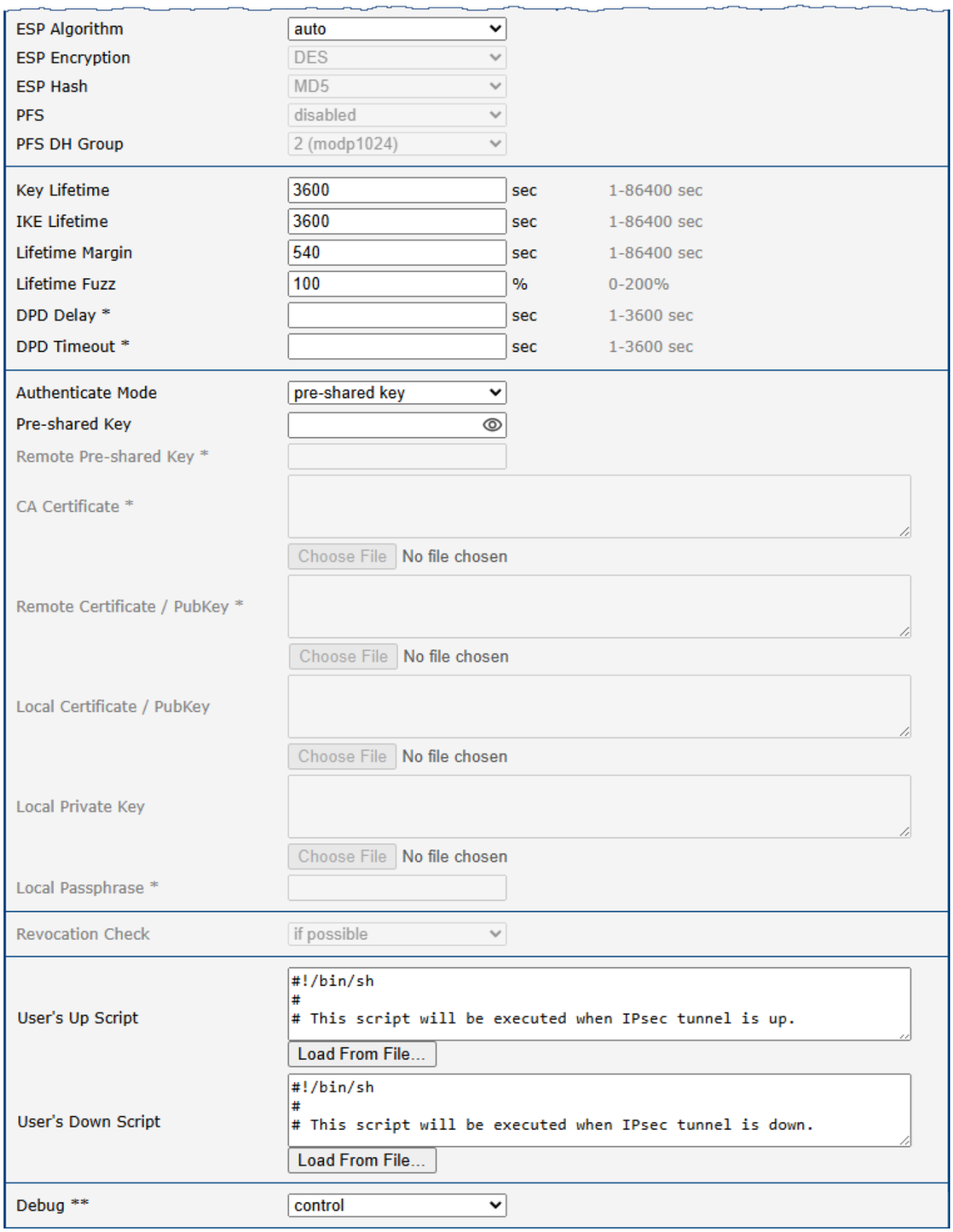

Figure 4: IPsec tunnels configuration page – part 2

Item	Description
Description	A user-defined name or description for the tunnel.
Type	• **policy-based** – Standard VPN approach based on security policies. • **route-based** – VPN approach based on routing rules. Data throughput may be slightly lower compared to policy-based VPN.
Host IP Mode	• **IPv4** – The router communicates with the remote peer using IPv4. • **IPv6** – The router communicates with the remote peer using IPv6.
1st Remote IP Address	The primary IPv4, IPv6 address, or domain name of the remote peer, corresponding to the selected *Host IP Mode*.
2nd Remote IP Address	The secondary (failover) IPv4 or IPv6 address, or domain name, of the remote peer. If configured, failover works as follows: at startup, the router initiates a connection to the *1st Remote IP Address*. If that connection fails, the router attempts to connect to the *2nd Remote IP Address*. Once the secondary connection is established, the router continues to use it until it fails — it does not automatically switch back to the primary address while the secondary connection is active.
Tunnel IP Mode	• **IPv4** – IPv4 traffic is transported inside the tunnel. • **IPv6** – IPv6 traffic is transported inside the tunnel.
Local ID	The identifier (ID) for the local side of the tunnel, typically composed of a hostname and a domain name (e.g., `router@mycompany.com`).
Remote ID	The identifier (ID) for the remote side of the tunnel.
Local Protocol/Port	Narrows the traffic selector by specifying the protocol and port for the local network. The format is *protocol/port* (e.g., `17/1701` for UDP port 1701).
Remote Protocol/Port	Narrows the traffic selector by specifying the protocol and port for the remote network.
Install Routes	For route-based mode only. If set to **yes**, the router automatically uses the traffic selectors to create and install routes.
Separate Child SA for Each Subnet	If enabled, a unique Child Security Association (SA) is created for each pair of local and remote subnets. This can improve interoperability with certain vendors and allow for more granular traffic policies. If disabled, a single Child SA covers all defined traffic selectors.
Local Subnet	The IPv4 or IPv6 address of the local network, based on the selected *Tunnel IP Mode*.
Local Subnet Mask	The IPv4 subnet mask or IPv6 prefix length (0–128) for the local network.
Remote Subnet	The IPv4 or IPv6 address of the network behind the remote peer.
Remote Subnet Mask	The IPv4 subnet mask or IPv6 prefix length for the remote network.
MTU	The Maximum Transmission Unit for the tunnel in route-based mode. The default value is 1426 bytes.
Remote Virtual Network	Specifies the virtual remote network for a server (responder).
Remote Virtual Mask	Specifies the virtual remote network mask for a server.
Local Virtual Address	Specifies the virtual local network address for a client. Use 0.0.0.0 to have an address assigned by the server.
Cisco FlexVPN	Enable to support Cisco FlexVPN functionality (route-based type only).
Encapsulation Mode	Specifies the IPsec encapsulation method: • **tunnel** – The entire IP datagram is encapsulated. • **transport** – Only the IP header is encapsulated (not supported for route-based VPN).
Force NAT Traversal	Enforces NAT traversal by enabling UDP encapsulation of ESP packets.

Table 2: IPsec tunnel configuration items description

Item	Description
IKE Protocol	Specifies the version of the Internet Key Exchange (IKE) protocol: **IKEv1/IKEv2** (auto-negotiate), or explicitly **IKEv1** or **IKEv2**. When set to IKEv1/IKEv2, the router first attempts to negotiate using IKEv2. If the peer does not support IKEv2, the router automatically falls back to IKEv1. IKEv2 is strongly recommended whenever possible, as it provides improved security and enhanced functionality.
IKE Mode	Specifies the mode for establishing a connection: *main* or *aggressive*. **It is strongly recommended not to use *aggressive* mode due to lower security.**
IKE Algorithm	Specifies how algorithms are selected: • **auto** – Encryption and hash algorithms are selected automatically. • **manual** – Algorithms are defined by the user.
IKE Encryption	Available encryption algorithms: **3DES**, **AES128**, **AES192**, **AES256**, **AES128GCM128**, **AES192GCM128**, **AES256GCM128**.
IKE Hash	Available hash algorithms: **MD5**, **SHA1**, **SHA256**, **SHA384**, **SHA512**.
IKE DH Group	Selects the Diffie-Hellman (DH) group, which determines the strength of the key used in the key exchange process. The choice of group is a trade-off between security and performance, as stronger groups require more computation. For detailed guidance on selecting an appropriate group, please refer to the official *Algorithm Proposals (Cipher Suites)*.
IKE Reauthentication	Enable or disable IKE reauthentication (for IKEv2 only).
XAUTH Enabled	Enable eXtended Authentication (for IKEv1 only).
XAUTH Mode	Select the XAUTH mode: *client* or *server*.
XAUTH Username	The username for XAUTH.
XAUTH Password	The password for XAUTH.
ESP Algorithm	Specifies how algorithms are selected: • **auto** – Encryption and hash algorithms are selected automatically. • **manual** – Algorithms are defined by the user.
ESP Encryption	Available encryption algorithms: **DES**, **3DES**, **AES128**, **AES192**, **AES256**, **AES128GCM128**, **AES192GCM128**, **AES256GCM128**, **CAMELLIA192**, **CAMELLIA256**, **CHACHA20POLY1305**.
ESP Hash	Available hash algorithms: **MD5**, **SHA1**, **SHA256**, **SHA384**, **SHA512**.
PFS	Enables or disables Perfect Forward Secrecy, which ensures that session keys are not compromised if one of the long-term private keys is compromised.
PFS DH Group	Specifies the Diffie-Hellman group for PFS (see *IKE DH Group*).
Key Lifetime	Specifies the maximum interval after which a new set of encryption keys is automatically negotiated. The typical value is a few hours. The connection remains uninterrupted.
IKE Lifetime	Specifies the time period after which the router must re-authenticate the connection. The typical value ranges from a few hours to several days and must be greater than *Key Lifetime*. The entire connection is re-established, and a brief interruption may occur.
Lifetime Margin	Specifies how long before *Key Lifetime* or *IKE Lifetime* expires the router should initiate rekeying or reauthentication, to ensure the process completes within the lifetime interval. This value should be less than half of *Key Lifetime* and *IKE Lifetime*.

Table 2: IPsec tunnel configuration items description (continued)

Item	Description
Lifetime Fuzz	Specifies a percentage used to calculate a random time offset added to *Lifetime Margin*. This introduces random variation in each rekeying and reauthentication cycle to prevent multiple devices from synchronizing their requests.
DPD Timeout	The period the router waits for a DPD response before considering the peer to be down.
Authenticate Mode	Specifies the authentication method: • **Pre-shared key** – Use a shared secret for both sides. • **X.509 Certificate** – Use X.509 certificates for authentication.
Pre-shared Key	The shared secret for both sides of the tunnel (for IKEv2, this is the local key). This field appears only when pre-shared key mode is selected.
Remote Pre-shared Key	The shared secret for the remote side (for IKEv2). Appears only when pre-shared key mode is selected.
CA Certificate	The CA certificate or chain used for X.509 authentication to validate the remote peer's certificate.
Remote Certificate / PubKey	The remote peer's X.509 certificate or public key for signature-based authentication.
Local Certificate / PubKey	The local router's X.509 certificate or public key.
Local Private Key	The private key corresponding to the local certificate.
Local Passphrase	The passphrase used during private key generation.
Revocation Check	Certificate revocation policy: • **if possible** – Fails only if a certificate is known to be revoked. • **if URI defined** – Fails if a CRL/OCSP URI is available, but revocation checking fails. • **always** – Fails if no revocation information is available (certificate is not known to be unrevoked).
User's Up Script[1]	A custom script executed when the IPsec tunnel is established.
User's Down Script[1]	A custom script executed when the IPsec tunnel is closed.
Debug	Controls the level of logging verbosity: • **silent** – No logging. • **audit** – Logs only successful connections and disconnections. • **control** – Default level, logs normal messages and errors. • **control-more** – More verbose control messages. • **raw** – Logs raw protocol messages. • **private** – Most verbose level, including private keys. See the *Logger Configuration* page on the *strongSwan* website for details.

Table 2: IPsec tunnel configuration items description (continued)

We recommend retaining the default settings. Increasing key lifetimes reduces operational costs but also decreases security. Conversely, shorter lifetimes increase security but may affect performance. Changes are applied after clicking the *Apply* button.

Important Considerations

Warning

- If local and remote subnets are not configured, only router-to-router traffic is encrypted.
- If protocol/port fields are configured, only traffic matching those settings is encapsulated.

[1] Parameters passed to the script: for policy-based, the connection name (e.g., `ipsec1-1`); for route-based, the connection name and interface name (e.g., `ipsec1-1` and `ipsec0`).

2.5 Certificate Generation

The following procedure describes how to generate certificates and keys without a passphrase:

```
******************** certification authority ************************
openssl rand -out private/.rand 1024
openssl genrsa -des3 -out private/ca.key 2048
openssl req -new -key private/ca.key -out tmp/myrootca.req
openssl x509 -req -days 7305 -sha1 -extensions v3_ca -signkey
private/ca.key -in tmp/myrootca.req -out ca.crt
******************** server cert ***********************************
openssl genrsa -out private/server.key 2048
openssl req -new -key private/server.key -out tmp/server.req
openssl x509 -req -days 7305 -sha1 -extensions v3_req -CA ca.crt -CAkey
private/ca.key -in tmp/server.req -CAserial ca.srl -CAcreateserial
-out server.crt
******************** client cert ***********************************
openssl genrsa -out private/client.key 2048
openssl req -new -key private/client.key -out tmp/client.req
openssl x509 -req -days 7305 -sha1 -extensions v3_req -CA ca.crt -CAkey
private/ca.key -in tmp/client.req -CAserial ca.srl -CAcreateserial
-out client.crt
```

The following example generates server and client certificates protected by the passphrase `router` (the certification authority remains unchanged):

```
******************** server cert ***********************************
openssl genrsa -des3 -passout pass:router -out private/server.pem 2048
openssl req -new -key private/server.pem -out tmp/server.req
openssl x509 -req -days 7305 -sha1 -extensions v3_req -CA ca.crt -CAkey
private/ca.key -in tmp/server.req -CAserial ca.srl -CAcreateserial
-out server.crt
******************** client cert ***********************************
openssl genrsa -des3 -passout pass:router -out private/client.pem 2048
openssl req -new -key private/client.pem -out tmp/client.req
openssl x509 -req -days 7305 -sha1 -extensions v3_req -CA ca.crt -CAkey
private/ca.key -in tmp/client.req -CAserial ca.srl -CAcreateserial
-out client.crt
```

The IPsec configuration supports the following identifier (ID) types for the *Remote ID* and *Local ID* parameters:

- IP address (for example, `192.168.1.1`)
- DN (for example, `C=CZ,O=CompanyName,OU=TP,CN=A`)
- FQDN (for example, `@director.companyname.cz`) – **the symbol precedes the FQDN.**
- User FQDN (for example, `director@companyname.cz`)

Info

Certificates and private keys must be in PEM format. Use only certificates that contain the standard `-----BEGIN CERTIFICATE-----` and `-----END CERTIFICATE-----` tags.

The interval after which the router renegotiates new keys is randomized as follows:

$$\text{Lifetime} - (\text{Rekey Margin} + \text{random value in range } \langle 0, \text{Rekey Margin} \times \text{Rekey Fuzz} / 100 \rangle)$$

With default settings, the key renegotiation occurs within the following time range:

- Minimum time: $1\,\text{h} - (9\,\text{min} + 9\,\text{min}) = 42\,\text{min}$
- Maximum time: $1\,\text{h} - (9\,\text{min} + 0\,\text{min}) = 51\,\text{min}$

2.6 IPsec Status — Tunnel Established

To verify the status of established IPsec tunnels, navigate to *Status* → *IPsec* in the router's web interface. If the tunnel has been established correctly, the page will show **ESTABLISHED** and indicate **1 up** active connection (highlighted in the figure below). If this text is not present (e.g., `0 up`), the tunnel was not established successfully.

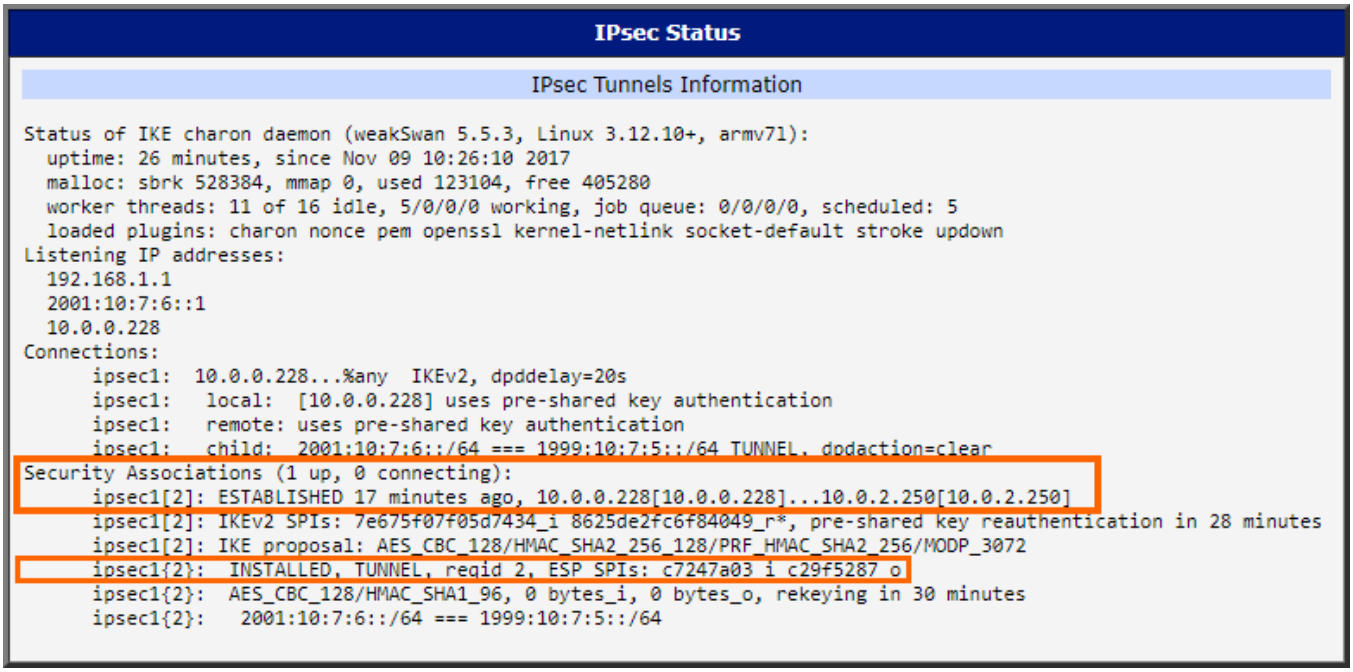

Figure 5: IPsec status page

3. Examples of Use

3.1 IPv6 IPsec Tunnel over IPv4 Internet

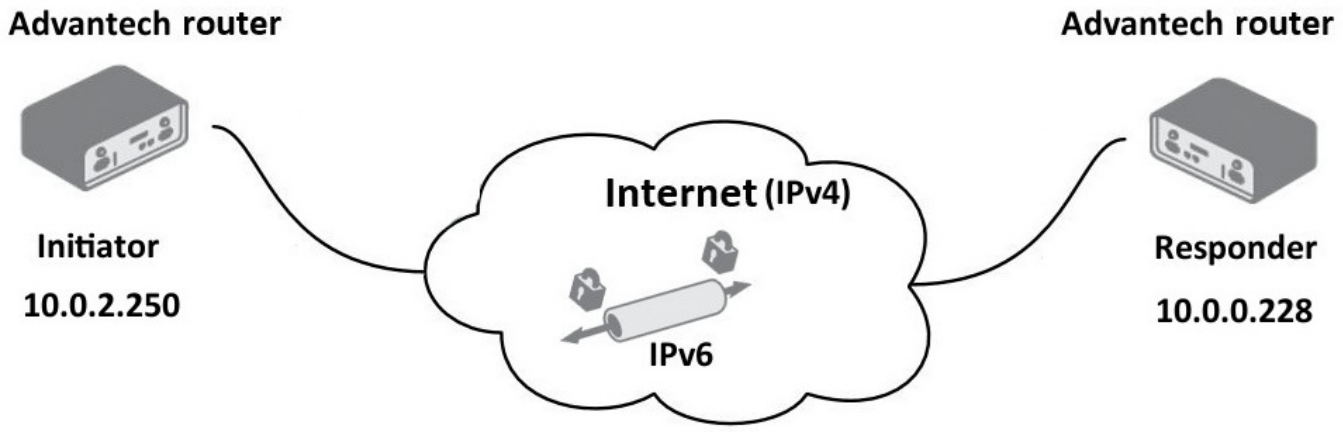

Figure 6: IPv6 IPsec tunnel over IPv4 Internet — two Advantech routers

This example demonstrates the establishment of an IPsec tunnel for an IPv6 network using two Advantech v3 routers (IPv6 is supported on the v3 platform). One router acts as the IPsec initiator and the other as the IPsec responder. Both routers connect to the Internet via IPv4, while the communication inside the established IPsec tunnel is IPv6, allowing the IPv6 networks on both sides to communicate with each other. The configuration pages and IPsec status pages are shown in the figures below.

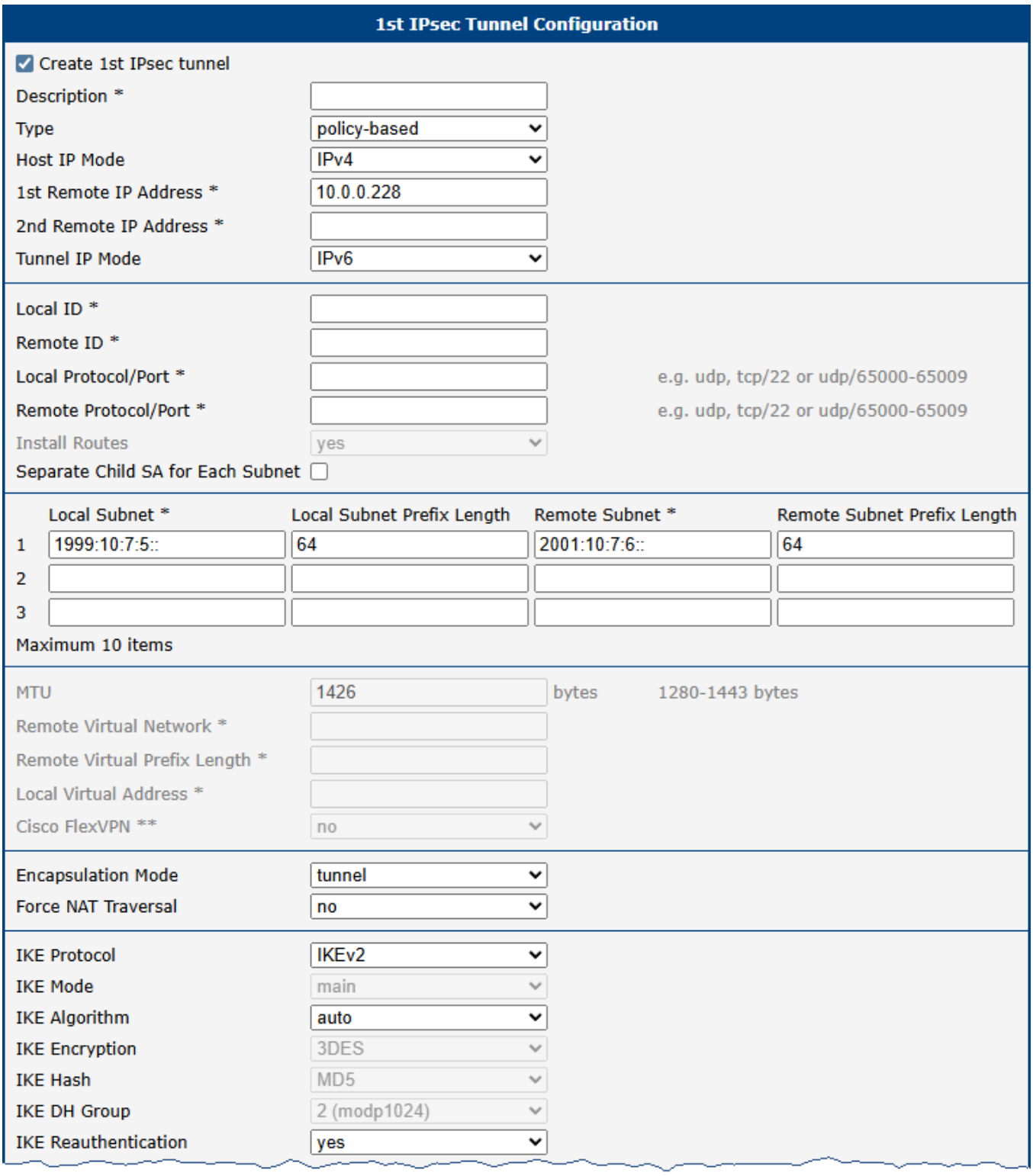

Figure 7: Initiator configuration of the IPv6 over IPv4 IPsec tunnel

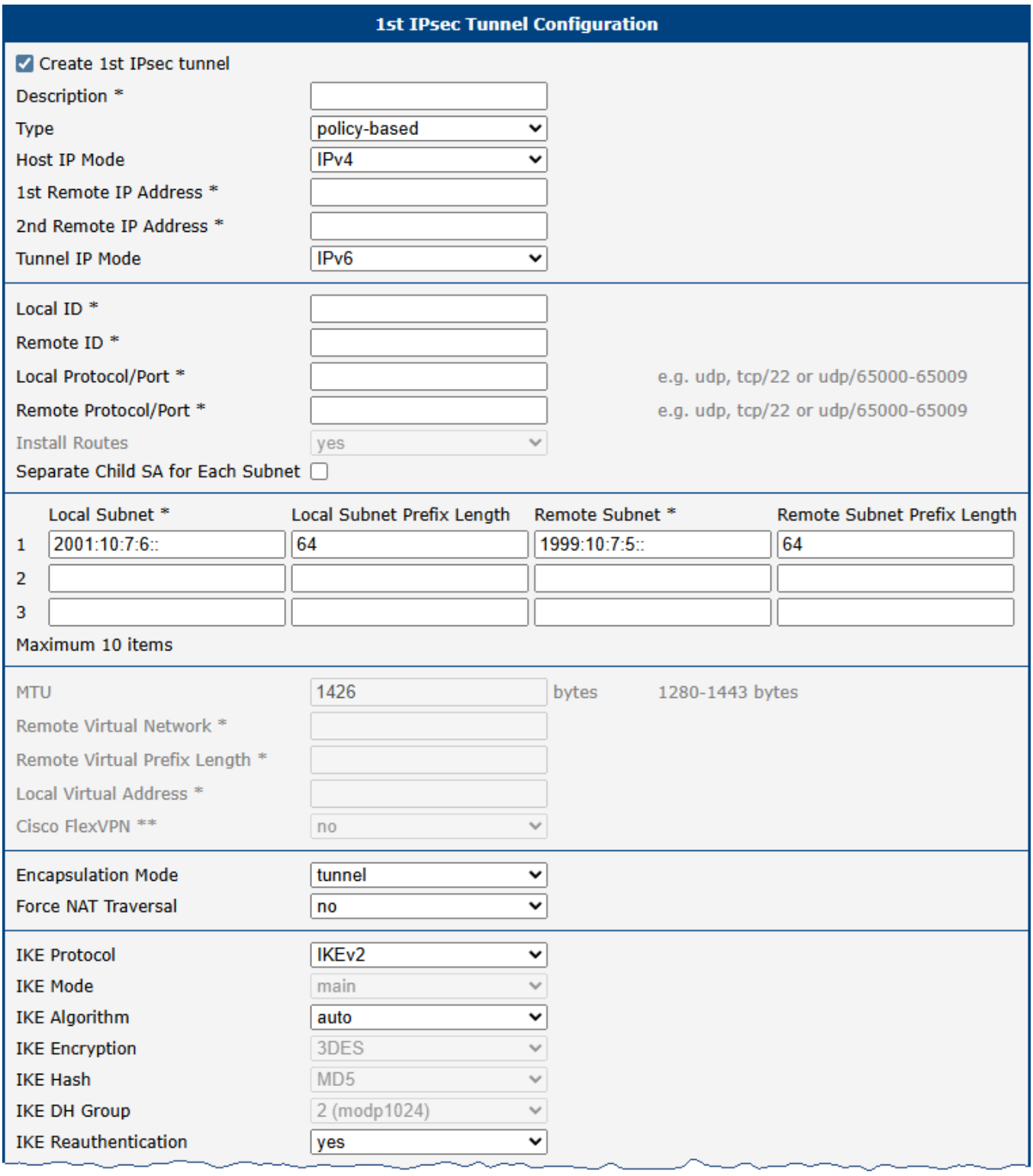

Figure 8: Responder configuration of the IPv6 over IPv4 IPsec tunnel

IPsec Status

IPsec Tunnels Information

```
Status of IKE charon daemon (weakSwan 5.5.3, Linux 3.12.10+, armv7l):
  uptime: 20 minutes, since Jan 01 00:08:11 2000
  malloc: sbrk 405504, mmap 0, used 122856, free 282648
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 4
  loaded plugins: charon nonce pem openssl kernel-netlink socket-default stroke updown
Listening IP addresses:
  192.168.10.1
  1999:10:7:5::1
  10.0.2.250
Connections:
      ipsec1:  10.0.2.250...10.0.0.228  IKEv2, dpddelay=20s
      ipsec1:   local:  [10.0.2.250] uses pre-shared key authentication
      ipsec1:   remote: [10.0.0.228] uses pre-shared key authentication
      ipsec1:   child:  1999:10:7:5::/64 === 2001:10:7:6::/64 TUNNEL, dpdaction=restart
Security Associations (1 up, 0 connecting):
      ipsec1[1]: ESTABLISHED 20 minutes ago, 10.0.2.250[10.0.2.250]...10.0.0.228[10.0.0.228]
      ipsec1[1]: IKEv2 SPIs: 7e675f07f05d7434_i* 8625de2fc6f84049_r, pre-shared key reauthentication in 16 minutes
      ipsec1[1]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
      ipsec1{1}:  INSTALLED, TUNNEL, reqid 1, ESP SPIs: c29f5287_i c7247a03_o
      ipsec1{1}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 22 minutes
      ipsec1{1}:   1999:10:7:5::/64 === 2001:10:7:6::/64
```

Figure 9: IPsec status of the initiator

IPsec Status

IPsec Tunnels Information

```
Status of IKE charon daemon (weakSwan 5.5.3, Linux 3.12.10+, armv7l):
  uptime: 26 minutes, since Nov 09 10:26:10 2017
  malloc: sbrk 528384, mmap 0, used 123104, free 405280
  worker threads: 11 of 16 idle, 5/0/0/0 working, job queue: 0/0/0/0, scheduled: 5
  loaded plugins: charon nonce pem openssl kernel-netlink socket-default stroke updown
Listening IP addresses:
  192.168.1.1
  2001:10:7:6::1
  10.0.0.228
Connections:
      ipsec1:  10.0.0.228...%any  IKEv2, dpddelay=20s
      ipsec1:   local:  [10.0.0.228] uses pre-shared key authentication
      ipsec1:   remote: uses pre-shared key authentication
      ipsec1:   child:  2001:10:7:6::/64 === 1999:10:7:5::/64 TUNNEL, dpdaction=clear
Security Associations (1 up, 0 connecting):
      ipsec1[2]: ESTABLISHED 17 minutes ago, 10.0.0.228[10.0.0.228]...10.0.2.250[10.0.2.250]
      ipsec1[2]: IKEv2 SPIs: 7e675f07f05d7434_i 8625de2fc6f84049_r*, pre-shared key reauthentication in 28 minutes
      ipsec1[2]: IKE proposal: AES_CBC_128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
      ipsec1{2}:  INSTALLED, TUNNEL, reqid 2, ESP SPIs: c7247a03_i c29f5287_o
      ipsec1{2}:  AES_CBC_128/HMAC_SHA1_96, 0 bytes_i, 0 bytes_o, rekeying in 30 minutes
      ipsec1{2}:   2001:10:7:6::/64 === 1999:10:7:5::/64
```

Figure 10: IPsec status of the responder

3.2 Advantech Router and Cisco Basic IPsec Tunnel Configurations

The examples in this section use **policy-based** mode. For a description of both VPN modes and their differences, see Section 2.1.

Warning

There is a known bug in Cisco ASA 5500-X Series Firewalls: IKEv2 between ASA and strongSwan (IKEv2 aes256/sha256) does not work. For more information, see `https://quickview.cloudapps.cisco.com/quickview/bug/CSCvb21927`.

3.2.1 IKEv1 Pre-Shared Key Tunnel

Advantech Router as IPsec Initiator

The IP address of the SIM card in the Advantech router can be either static or dynamic, because the IPsec tunnel is initiated by the router. In this case, the remote peer (Linux or Cisco router) acts as the IPsec responder and must therefore be reachable at a static IP address or a fixed domain name.

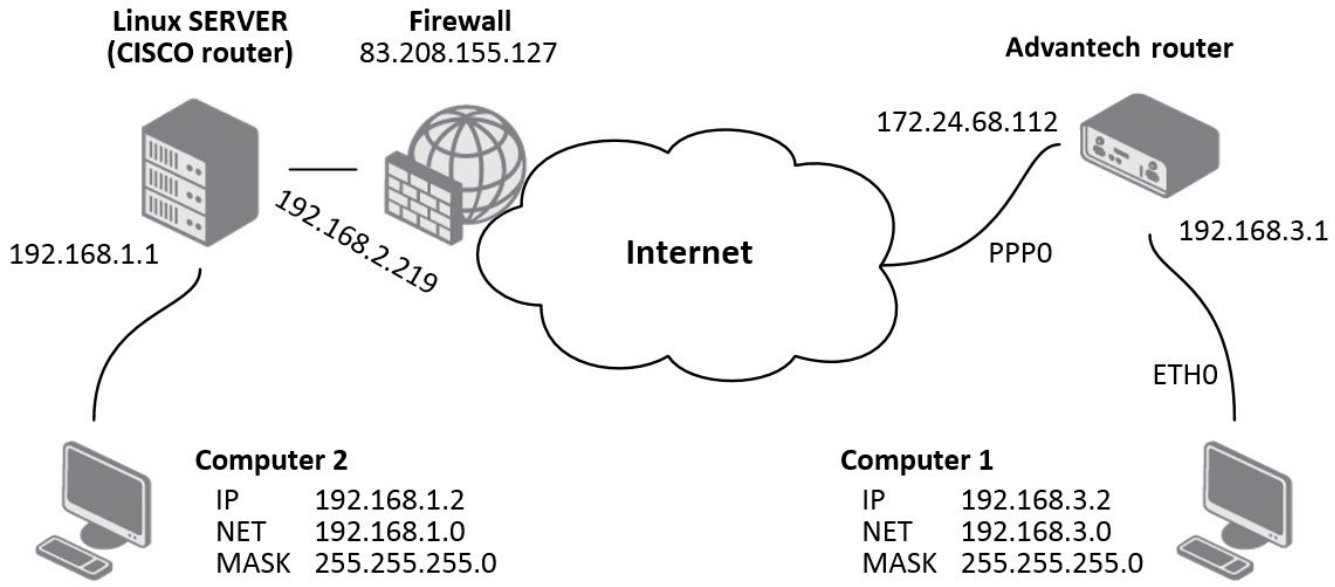

Figure 11: IPsec tunnel — initiator on the router

If both tunnel endpoints are mutually reachable (no NAT between them), it is sufficient to configure only the following items: *Description*, *Remote IP Address*, *First Remote Subnet*, *First Remote Subnet Mask*, *First Local Subnet*, and *First Local Subnet Mask*. If one end of the tunnel is behind NAT, *Force NAT Traversal* must be set to *yes*.

If *Force NAT Traversal* is enabled, the *Remote ID* must also be configured. The Remote ID must be set to an FQDN (Fully Qualified Domain Name) — the fully specified domain name of the remote host. Authentication using certificates is also supported, in which case the *Remote ID* does not need to be entered.

The following table shows an example of IPsec tunnel settings corresponding to the figure above:

Item	Value
Remote IP Address	83.208.155.127
Remote ID	ciscoasa@default.domain
First Remote Subnet	192.168.1.0
First Remote Subnet Mask	255.255.255.0
First Local Subnet	192.168.3.0
First Local Subnet Mask	255.255.255.0
Force NAT Traversal	yes
Pre-shared Key	test

Table 3: IPsec tunnel settings (initiator)

All other parameters can be left at their default values. If the *Remote IP Address* is left empty on one side of the IPsec tunnel, that side will wait for an incoming connection and will not attempt to initiate one.

Items not listed in the example settings and marked with an asterisk (*) are optional. They are used for more precise identification of the tunnel.

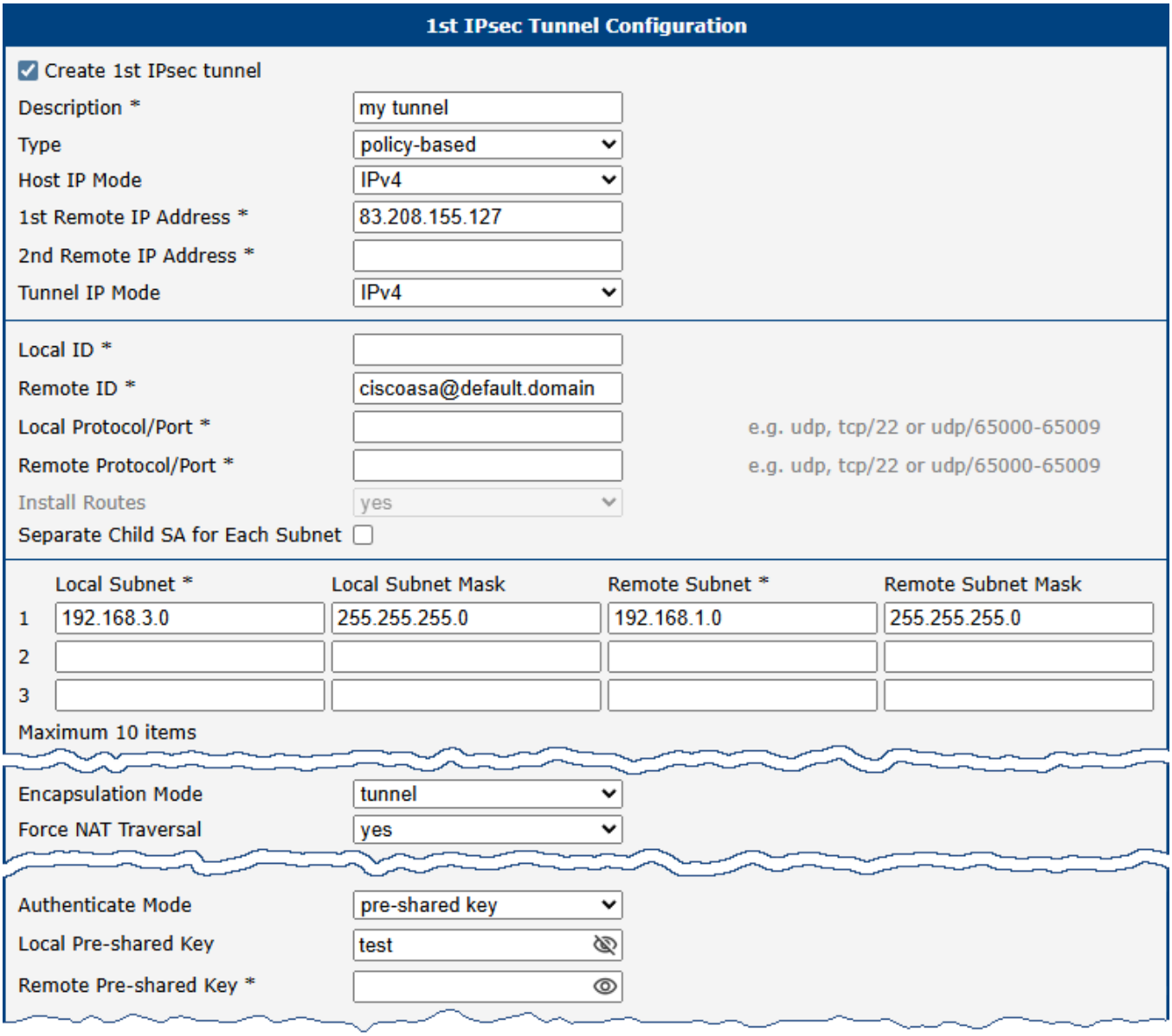

Figure 12: IPsec tunnel — example configuration of initiator on the router

Information about the active IPsec tunnel can be found under *Status* → *IPsec* in the router's web interface.

Advantech Router as IPsec Responder

The Advantech router must have a static IP address, or a dynamic IP address mapped to a DynDNS domain name. In this case, the remote peer (Linux or Cisco router) acts as the initiator and establishes the IPsec tunnel.

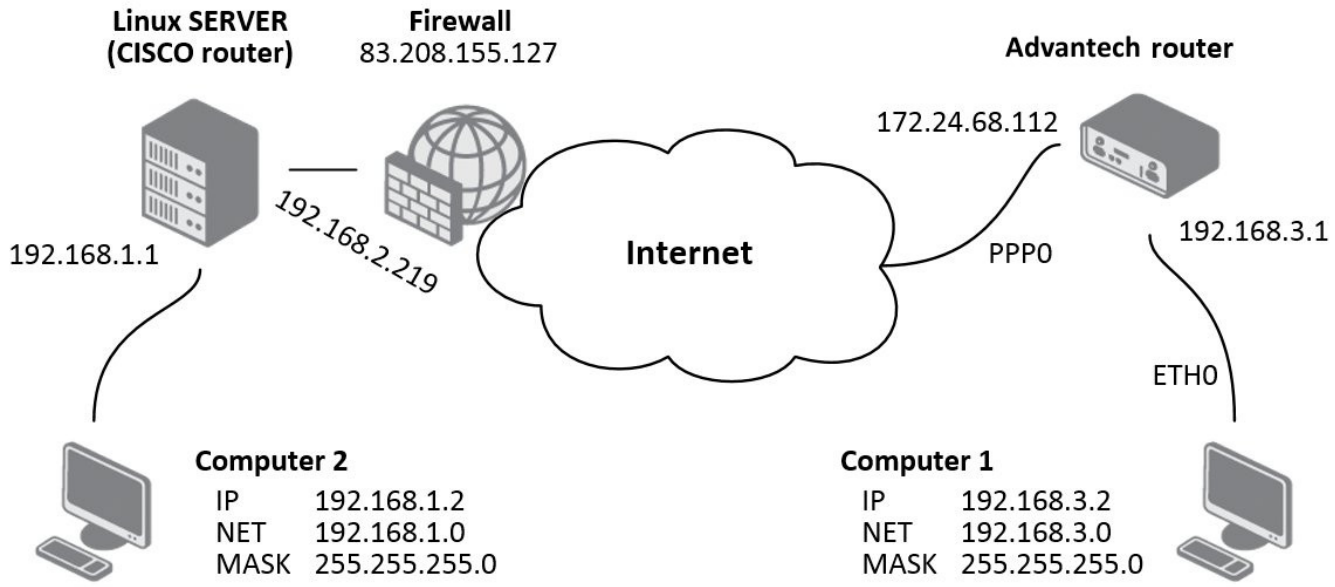

Figure 13: IPsec tunnel — responder on the router

If both tunnel endpoints are mutually reachable (no NAT between them), it is sufficient to configure only the following items: *Description*, *First Remote Subnet*, and *First Remote Subnet Mask*. If one end of the tunnel is behind NAT, *Force NAT Traversal* must be set to *yes*.

If *Force NAT Traversal* is enabled, the *Remote ID* must also be configured. The Remote ID must be set to an FQDN (Fully Qualified Domain Name) — the fully specified domain name of the remote host. Authentication using certificates is also supported, in which case the *Remote ID* does not need to be entered.

The following table shows an example of IPsec tunnel settings corresponding to the figure above:

Item	Value
Remote ID	ciscoasa@default.domain
First Remote Subnet	192.168.2.219
First Remote Subnet Mask	255.255.255.255
Force NAT Traversal	yes
Pre-shared Key	test

Table 4: IPsec tunnel settings (responder)

All other parameters can be left at their default values. If the *Remote IP Address* is left empty on one side of the IPsec tunnel, that side will wait for an incoming connection and will not attempt to initiate one.

Items not listed in the example settings and marked with an asterisk (*) are optional. They are used for more precise identification of the tunnel.

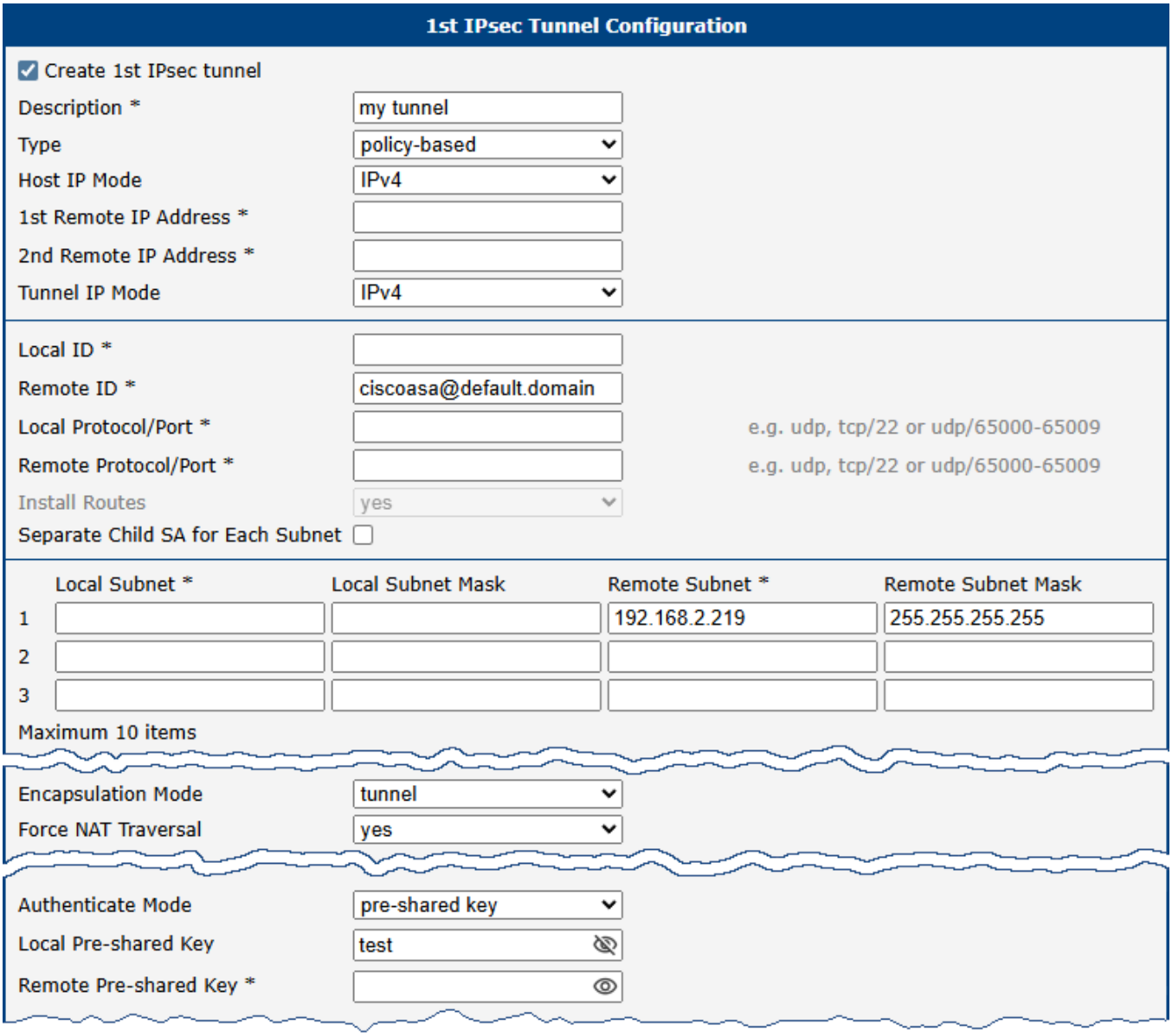

Figure 14: IPsec tunnel — example configuration of responder on the router

Information about the active IPsec tunnel can be found under *Status* → *IPsec* in the router's web interface.

Linux Server IPsec Configuration

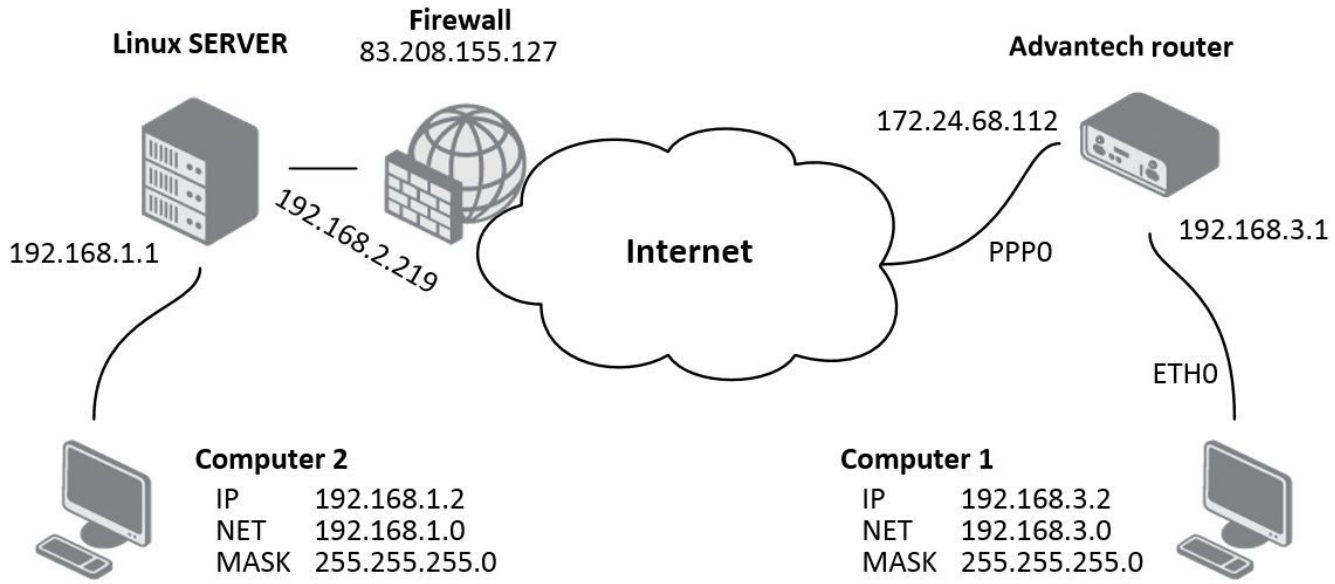

Figure 15: IPsec tunnel — Linux server

On the Linux server, the `ipsec.conf` and `ipsec.secrets` files must be configured. The `ipsec.conf` file can be configured as shown in the following example:

```
conn advantechrouter
   authby=secret
   type=tunnel
   left=83.208.155.127
   leftsubnet=192.168.1.0/24
   right=172.24.68.112
   rightsubnet=192.168.3.0/24
   ikelifetime=3600s
   keylife=3600s
   pfs=no
   auto=add
```

The `ipsec.secrets` file should be configured as follows:
`83.208.155.127 172.24.68.112: PSK "test"`

Cisco Router as Initiator — IPsec Configuration

Warning

Cisco routers support the IPsec protocol from IOS version 7.1 onwards.

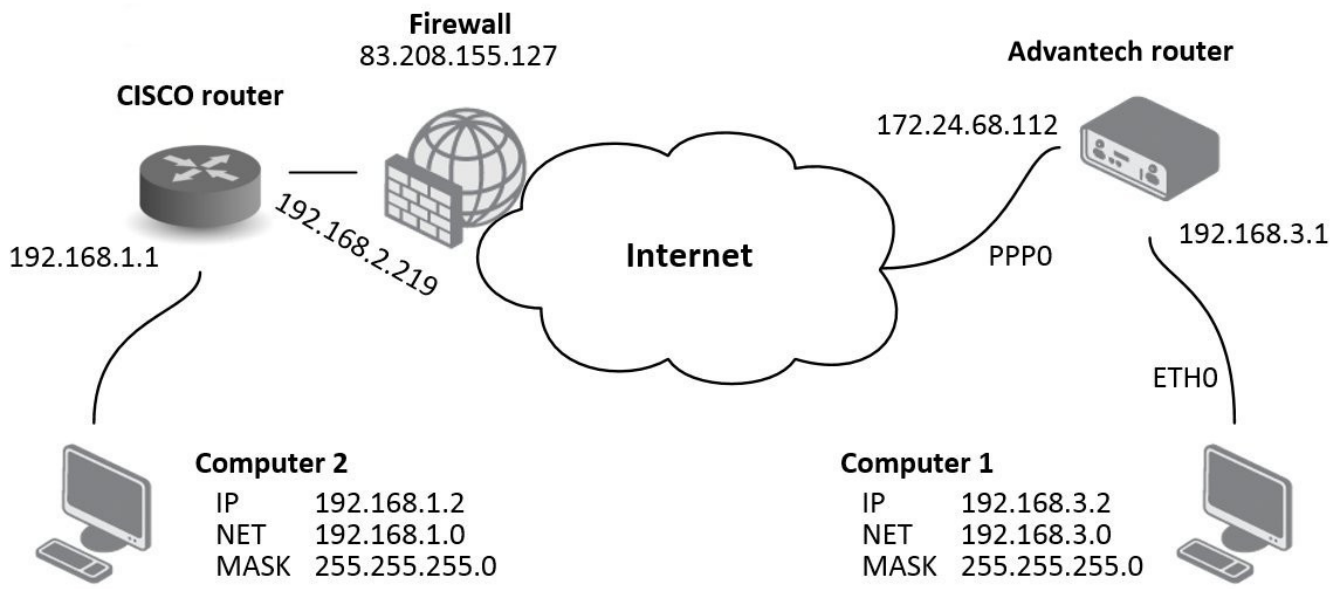

Figure 16: IPsec tunnel — Cisco router as initiator

```
access-list outside_2_cryptomap extended permit ip 192.168.1.0 255.255.255.0 192.168.3.0
↪  255.255.255.0
crypto ipsec transform-set ESP-DES-MD5 esp-des esp-md5-hmac
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto ipsec transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
crypto ipsec transform-set ESP-AES-128-SHA esp-aes esp-sha-hmac
crypto ipsec transform-set ESP-AES-256-MD5 esp-aes-256 esp-md5-hmac
crypto ipsec transform-set ESP-AES-256-SHA esp-aes-256 esp-sha-hmac
crypto ipsec transform-set ESP-AES-128-MD5 esp-aes esp-md5-hmac
crypto ipsec transform-set ESP-AES-192-MD5 esp-aes-192 esp-md5-hmac
crypto ipsec transform-set ESP-AES-192-SHA esp-aes-192 esp-sha-hmac
crypto ipsec transform-set UR1 esp-3des esp-none
crypto ipsec transform-set UR2 esp-des esp-none
crypto ipsec transform-set ESP-3DES-MD5 esp-3des esp-md5-hmac
crypto map outside_map 1 match address outside_2_cryptomap
crypto map outside_map 1 set connection-type answer-only
crypto map outside_map 1 set peer 172.24.68.112
crypto map outside_map 1 set transform-set ESP-3DES-MD5
crypto map outside_map interface outside
crypto isakmp identity hostname
crypto isakmp enable outside
crypto isakmp policy 10
authentication pre-share
encryption 3des
hash md5
group 2
lifetime 3600
crypto isakmp nat-traversal 20
```

```
group-policy DfltGrpPolicy attributes
banner none
wins-server none
dns-server none
dhcp-network-scope none
vpn-access-hours none
vpn-simultaneous-logins 3
vpn-idle-timeout none
vpn-session-timeout none
vpn-filter none
vpn-tunnel-protocol IPSec l2tp-ipsec webvpn
password-storage disable
ip-comp disable
re-xauth disable
group-lock none
pfs disable
ipsec-udp enable
ipsec-udp-port 10000
split-tunnel-policy tunnelall
split-tunnel-network-list none
default-domain none

tunnel-group DefaultL2LGroup ipsec-attributes
pre-shared-key *
isakmp keepalive threshold 20 retry 10
tunnel-group 172.24.68.112 type ipsec-l2l
tunnel-group 172.24.68.112 ipsec-attributes
pre-shared-key *
tunnel-group-map enable rules
tunnel-group-map default-group DefaultL2LGroup
prompt hostname context
```

Cisco Router as Responder — IPsec Configuration

Warning

Cisco routers support the IPsec protocol from IOS version 7.1 onwards.

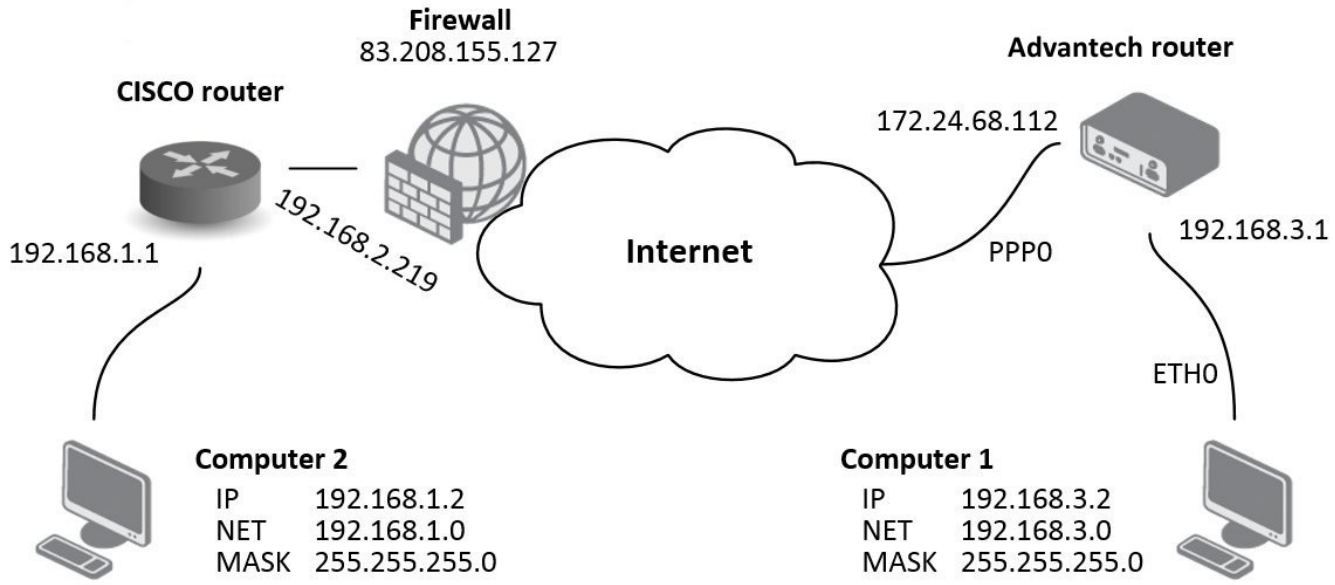

Figure 17: IPsec tunnel — Cisco router as responder

```
access-list outside_2_cryptomap extended permit ip 192.168.1.0 255.255.255.0 192.168.3.0
↪  255.255.255.0
crypto ipsec transform-set ESP-DES-MD5 esp-des esp-md5-hmac
crypto ipsec transform-set ESP-DES-SHA esp-des esp-sha-hmac
crypto ipsec transform-set ESP-3DES-SHA esp-3des esp-sha-hmac
crypto ipsec transform-set ESP-AES-128-SHA esp-aes esp-sha-hmac
crypto ipsec transform-set ESP-AES-256-MD5 esp-aes-256 esp-md5-hmac
crypto ipsec transform-set ESP-AES-256-SHA esp-aes-256 esp-sha-hmac
crypto ipsec transform-set ESP-AES-128-MD5 esp-aes esp-md5-hmac
crypto ipsec transform-set ESP-AES-192-MD5 esp-aes-192 esp-md5-hmac
crypto ipsec transform-set ESP-AES-192-SHA esp-aes-192 esp-sha-hmac
crypto ipsec transform-set UR1 esp-3des esp-none
crypto ipsec transform-set UR2 esp-des esp-none
crypto ipsec transform-set ESP-3DES-MD5 esp-3des esp-md5-hmac
crypto map outside_map 1 match address outside_2_cryptomap
crypto map outside_map 1 set connection-type originate-only
crypto map outside_map 1 set peer 172.24.68.112
crypto map outside_map 1 set transform-set ESP-3DES-MD5
crypto map outside_map interface outside
crypto isakmp identity hostname
crypto isakmp enable outside
```

```
crypto isakmp policy 10
  authentication pre-share
  encryption 3des
  hash md5
  group 2
  lifetime 3600
crypto isakmp nat-traversal 20

group-policy DfltGrpPolicy attributes
  banner none
  wins-server none
  dns-server none
  dhcp-network-scope none
  vpn-access-hours none
  vpn-simultaneous-logins 3
  vpn-idle-timeout none
  vpn-session-timeout none
  vpn-filter none
  vpn-tunnel-protocol IPSec l2tp-ipsec webvpn
  password-storage disable
  ip-comp disable
  re-xauth disable
  group-lock none
  pfs disable
  ipsec-udp enable
  ipsec-udp-port 10000
  split-tunnel-policy tunnelall
  split-tunnel-network-list none
  default-domain none

tunnel-group DefaultL2LGroup ipsec-attributes
  pre-shared-key *
isakmp keepalive threshold 20 retry 10
tunnel-group 172.24.68.112 type ipsec-l2l
tunnel-group 172.24.68.112 ipsec-attributes
  pre-shared-key *
tunnel-group-map enable rules
tunnel-group-map default-group DefaultL2LGroup
prompt hostname context
```

3.2.2 Certificate Generation

This section describes how to generate certificates and keys on a Linux- or Windows-based machine.

1. Certification Authority — `ca.key`, `ca.csr`, `ca.crt`

- Create the working directory and initialize the certificate database:
  `mkdir certs; cd certs; touch index.txt`
- Copy the `openssl.conf` configuration file into the working directory.
- Generate the CA private key:
  `openssl genrsa -des3 -out ca.key 2048`
- Generate the certificate signing request (CSR):
  `openssl req -verbose -new -key ca.key -out ca.csr -sha256`
- Self-sign the CA certificate (see Appendix *A* for an example `openssl.conf`):
  `openssl ca -create_serial -extensions v3_ca -config ./openssl.conf -out ca.crt -keyfile ca.key -verbose -selfsign -md sha256 -enddate 301231235959Z -infiles ca.csr`
- Verify the CA certificate:
  `openssl x509 -noout -text -in ca.crt`

2. Server Certificate — `server_cisco.key`, `server_cisco.csr`, `server_cisco.crt`

- Generate the server private key:
  `openssl genrsa -des3 -out server_cisco.key 2048`
- Generate the certificate signing request (see Appendix *B* for an example `server_req.conf`):
  `openssl req -verbose -new -key server_cisco.key -out server_cisco.csr -config server_req.conf`
- Sign the server certificate using the CA:
  `openssl ca -config ./server_req.conf -extensions v3_req -enddate 301231235959Z -out server_cisco.crt -keyfile ca.key -infiles server_cisco.csr`
- Verify the server certificate:
  `openssl x509 -noout -text -in server_cisco.crt`

3. Client Certificate — `client_router.key`, `client_router.csr`, `client_router.crt`

- Generate the client private key:
  `openssl genrsa -des3 -out client_router.key 2048`
- Generate the certificate signing request (see Appendix *C* for an example `client_req.conf`):
  `openssl req -verbose -new -key client_router.key -out client_router.csr -config client_req.conf`
- Sign the client certificate using the CA:
  `openssl ca -config ./client_req.conf -extensions v3_req -enddate 301231235959Z -out client_router.crt -keyfile ca.key -infiles client_router.csr`
- Verify the client certificate:
  `openssl x509 -noout -text -in client_router.crt`

4. Verify That the Certificates and Keys Match

The modulus hashes of the certificate and its corresponding private key must be identical:

- `openssl x509 -noout -modulus -in [client_router/server_cisco].crt | openssl md5`
- `openssl rsa -noout -modulus -in [client_router/server_cisco].key | openssl md5`

3.2.3 How to Import Certificates to Cisco

This chapter is an example showing how to import ca, server key and server certificates to a Cisco device.

1. configure terminal
2. crypto pki trustpoint server.ciso

```
no revocation-check
enrollment terminal pem
exit
```

3. crypto pki import server.cisco pem terminal password <password>

```
paste ca certificate in PEM format
paste encrypted private server key in PEM format
paste server certificate in PEM format
exit
```

4. crypto pki certificate map ike_v2_certmap 10

```
subject-name co client
```

5. show crypto pki trustpoint server.cisco status

```
Trustpoint server.cisco:
Issuing CA certificate configured:
  Subject Name:
   e=advantech@advantech.com,cn=www.advantech.com,ou=Advantech CZ,o=Advantech,
                                                                  st=Czechia,c=CZ
  Fingerprint MD5: 20514117 B5B696F5 00375153 A9DC864C
  Fingerprint SHA1: 532AA251 EB16DAEC 89BB97C4 DDE0D3E3 F7A07270
Router General Purpose certificate configured:
  Subject Name:
   cn=server@cisco,ou=Advantech CZ,o=Advantech,st=Czechia,c=CZ
  Fingerprint MD5: 1712292C A41F36FE 56F12682 1A503577
  Fingerprint SHA1: 01C99D4C 4064AFF6 123421A1 5A9F23BB 8DEA2D60
State:
  Keys generated ..... Yes (General Purpose, non-exportable)
  Issuing CA authenticated ..... Yes
  Certificate request(s) ..... Yes
```

Note: If cisco is configured by copy/paste raw config via terminal then private keys are not imported (only ca and cert is imported). In this case you can use these cmd to import private key:

1. crypto key import rsa <name> terminal <password>
2. crypto pki trustpoint <name>
   rsakeypair <name>

3.2.4 IKEv1 Certificate-Based Tunnel

This section describes how to set up an IKEv1 certificate-based tunnel between a Cisco device and an Advantech router. The Cisco device acts as the IPsec responder and the Advantech router as the initiator. See Section 3.2.2 for certificate generation and Section 3.2.3 for certificate import instructions.

Cisco Device Configuration

1. `configure terminal`
2. `crypto pki certificate map ikev1_map 10`

   ```
   subject-name co client
   ```

3. `crypto isakmp policy 10`

   ```
   encr aes 256
   hash sha256
   group 14
   ```

4. `crypto isakmp identity dn` (identity is the DN of the `server.cisco` certificate)
5. `crypto isakmp profile ikev1`

   ```
   ca trust-point server.cisco
   match certificate ikev1_map
   local-address <IP address>
   ```

6. `crypto map ike_v1_map 10 ipsec-isakmp`

   ```
   set peer <IP address>
   set transform-set aeset        ! ESP algorithms and mode are the same as for IKEv2
   set isakmp-profile ikev1
   match address ike_v2_acl       ! traffic selector is the same as for IKEv2
   ```

7. `interface GigabitEthernet0`

   ```
   ip address <IP address> <mask>
   duplex auto
   speed auto
   no keepalive
   crypto map ike_v1_map
   ```

8. `exit`
9. Verify the IKE security associations: `show crypto isakmp sa detail`

```
Codes: C - IKE configuration mode, D - Dead Peer Detection
       K - Keepalives, N - NAT-traversal
       T - cTCP encapsulation, X - IKE Extended Authentication
       psk - Preshared key, rsig - RSA signature
       renc - RSA encryption

IPv4 Crypto ISAKMP SA

C-id  Local         Remote        I-VRF  Status Encr Hash   Auth DH Lifetime Cap.

2966  <IP address>  <IP address>         ACTIVE aes  sha256 rsig 14 00:53:05
Engine-id:Conn-id = SW:966
```

Advantech Router Configuration

```
IPSEC_ENABLED=1
IPSEC_DESCRIPTION=
IPSEC_HOST_IPMODE=4
IPSEC_REMOTE_IPADDR=<IP address>
IPSEC_TUNNEL_IPMODE=4
IPSEC_REMOTE_ID=C=CZ,ST=Czechia,O=Advantech,OU=AdvantechCZ,CN=server@cisco
IPSEC_REMOTE_NETWORK=<IP address>
IPSEC_REMOTE_NETMASK=<mask>
IPSEC_REMOTE_NETWORK2=
IPSEC_REMOTE_NETMASK2=
IPSEC_REMOTE_PROTOPORT=
IPSEC_LOCAL_ID=<IP address>
IPSEC_LOCAL_NETWORK=<IP address>
IPSEC_LOCAL_NETMASK=<mask>
IPSEC_LOCAL_NETWORK2=
IPSEC_LOCAL_NETMASK2=
IPSEC_LOCAL_PROTOPORT=
IPSEC_IKE_PROTOCOL=ikev1
IPSEC_IKE_ALG=manual
IPSEC_IKE_ENC=aes256
IPSEC_IKE_HASH=sha2_256
IPSEC_IKE_DH=modp2048
IPSEC_IKE_REAUTH=0
IPSEC_XAUTH_ENABLED=0
IPSEC_XAUTH_MODE=client
IPSEC_XAUTH_USER=
IPSEC_XAUTH_PASS=
IPSEC_ESP_ALG=manual
IPSEC_ESP_ENC=aes256
IPSEC_ESP_HASH=sha2_256
IPSEC_PFS=0
IPSEC_PFS_DH=
IPSEC_KEY_LIFE=3600
IPSEC_IKE_LIFE=3600
IPSEC_REKEY_MARGIN=540
IPSEC_REKEY_FUZZ=100
IPSEC_DPD_DELAY=20
IPSEC_DPD_TIMEOUT=60
IPSEC_ENCAP=tunnel
IPSEC_FORCE_ENCAPS=0
IPSEC_AGGRESSIVE=0
IPSEC_AUTHBY=rsa
IPSEC_PSK=
IPSEC_CA_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1.....
IPSEC_REMOTE_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0.....
IPSEC_LOCAL_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t.....
IPSEC_LOCAL_KEY=LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktL.....
IPSEC_LOCAL_PASS=password
IPSEC_DEBUG=1
```

3.2.5 IKEv2 Certificate-based Tunnel

This chapter describes how to set up an IKEv2 certificate-based tunnel between the Cisco device and the Advantech router. The Cisco device acts as the server and the Advantech router as the client. See chaper 3.2.2 for demonstration of cerfificate generation and chapter 3.2.3 to see how to import it.

Setup of Cisco

1. configure terminal
2. crypto ikev2 authorization policy ike_v2_policy

```
crypto ikev2 proposal ike_v2_proposal
encryption aes-cbc-256
integrity sha256
group 14
```

3. crypto ikev2 policy ike_v2_policy

```
  proposal ike_v2_proposal
  crypto ikev2 profile ike_v2_profile
  match certificate ike_v2_certmap
  identity local [ fqdn server.cisco | email server@cisco | address XX.XX.XX.XX ]
  authentication remote rsa-sig
  authentication local rsa-sig
  pki trustpoint server.cisco
```

4. crypto ipsec transform-set aeset esp-aes 256 esp-sha256-hmac

```
  mode tunnel
```

5. crypto map ike_v2_map 10 ipsec-isakmp

```
  set peer <IP address>
  set transform-set aeset
  set ikev2-profile ike_v2_profile
  match address ike_v2_acl
```

6. ip access-list extended ike_v2_acl

```
  permit ip <local subnet> 0.0.0.255 <remote subnet> 0.0.0.255
```

7. interface GigabitEthernet0

```
    ip address <IP address> <mask>
    duplex auto
    speed auto
    no keepalive
    crypto map ike_v2_map
```

8. exit

9. show crypto ikev2 session

```
  IPv4 Crypto IKEv2 Session

  Session-id:28, Status:UP-ACTIVE, IKE count:1, CHILD count:1

  Tunnel-id Local              Remote             fvrf/ivrf          Status
  1         <IP address>/4500     <IP address>/4500    none/none     READY
        Encr: AES-CBC, keysize: 256, Hash: SHA256, DH Grp:14, Auth sign: RSA,
                                                              Auth verify: RSA
        Life/Active Time: 86400/1149 sec
  Child sa: local selector  192.168.6.0/0 - 192.168.6.255/65535
            remote selector 192.168.1.0/0 - 192.168.1.255/65535
            ESP spi in/out: 0xE5E902B1/0xC8A42CE4
```

10. show crypto ipsec sa

```
    interface: GigabitEthernet0
      Crypto map tag: ike_v2_map, local addr <IP address>

     protected vrf: (none)
     local  ident (addr/mask/prot/port): (192.168.6.0/255.255.255.0/0/0)
     remote ident (addr/mask/prot/port): (192.168.1.0/255.255.255.0/0/0)
     current_peer <IP address> port 4500
       PERMIT, flags={origin_is_acl,}
      #pkts encaps: 0, #pkts encrypt: 0, #pkts digest: 0
      #pkts decaps: 0, #pkts decrypt: 0, #pkts verify: 0
      #pkts compressed: 0, #pkts decompressed: 0
      #pkts not compressed: 0, #pkts compr. failed: 0
      #pkts not decompressed: 0, #pkts decompress failed: 0
      #send errors 0, #recv errors 0

       local crypto endpt.: <IP address>, remote crypto endpt.: <IP address>
       path mtu 1500, ip mtu 1500, ip mtu idb GigabitEthernet0
       current outbound spi: 0xC8A42CE4(3366202596)
       PFS (Y/N): N, DH group: none
```

```
      inbound esp sas:
        spi: 0xE5E902B1(3857253041)
          transform: esp-256-aes esp-sha256-hmac ,
          in use settings ={Tunnel, }
          conn id: 55, flow_id: Onboard VPN:55, sibling_flags 80000040,
                                                    crypto map: ike_v2_map
          sa timing: remaining key lifetime (k/sec): (4608000/2356)
          IV size: 16 bytes
          replay detection support: Y
          Status: ACTIVE(ACTIVE)

       inbound ah sas:
       inbound pcp sas:

       outbound esp sas:
        spi: 0xC8A42CE4(3366202596)
          transform: esp-256-aes esp-sha256-hmac ,
          in use settings ={Tunnel, }
          conn id: 56, flow_id: Onboard VPN:56, sibling_flags 80000040,
                                                    crypto map: ike_v2_map
          sa timing: remaining key lifetime (k/sec): (4608000/2356)
          IV size: 16 bytes
          replay detection support: Y
          Status: ACTIVE(ACTIVE)
```

11. show runnig-config

```
mit ip <local subnet> 0.0.0.255
 crypto pki trustpoint server.cisco

 revocation-check none
 rsakeypair server.cisco
!
!
!
crypto pki certificate map ike_v2_certmap 10
 subject-name co client
!
crypto pki certificate chain server.cisco
 certificate 29BEF8C0BE9377F585E4C9E7E569B4B1FEA8544A
  3082035A 30820242 A0030201 02020900 89CE1443 6667652F 300D0609 2A864886
  .....
  7A8B2AE7 2EF6FBB7 F9BE79B3 6DBD32C1 3F63EA9F 28460A23 122785C2 0504
  quit
certificate ca 29BEF8C0BE9377F585E4C9E7E569B4B1FEA8543C
  3082035D 30820245 A0030201 02020900 C32DDAD5 EF9ADEDE 300D0609 2A864886
  .....
  9AD70CB3 05431A4F DDA40424 657A29FF 5F1174FD 21171128 A541B781 CEAB845A C6
  quit
ip cef
!
!
!
!
!
```

```
crypto ikev2 authorization policy ike_v2_policy
!
crypto ikev2 proposal ike_v2_proposal
 encryption aes-cbc-256
 integrity sha256
 group 14
!
crypto ikev2 policy ike_v2_policy
 proposal ike_v2_proposal
!
!
crypto ikev2 profile ike_v2_profile
 match certificate ike_v2_certmap
 identity local fqdn server.cisco
 authentication remote rsa-sig
 authentication local rsa-sig
 pki trustpoint server.cisco
!
!
!
crypto ipsec transform-set aeset esp-aes 256 esp-sha256-hmac
 mode tunnel
!
!
crypto ipsec transform-set aeset esp-aes 256 esp-sha256-hmac
 mode tunnel
!
crypto map ike_v2_map 10 ipsec-isakmp
 set peer <IP address>
 set transform-set aeset
 set ikev2-profile ike_v2_profile
 match address ike_v2_acl
!
!
!
!
interface GigabitEthernet0
 ip address <IP address> <mask>
 ip access-group 101 in
 duplex auto
 speed auto
 no keepalive
 crypto map ike_v2_map
!
interface Vlan1
 ip address <cisco subnet> <mask>
!
!
!
ip access-list extended ike_v2_acl
 permit ip <cisco's subnet> <mask> <router's subnet> <mask>
!
access-list 101 permit ip any any
access-list 101 permit icmp any any
```

Setup of Advantech Router

```
IPSEC_ENABLED=1
IPSEC_DESCRIPTION=
IPSEC_HOST_IPMODE=4
IPSEC_REMOTE_IPADDR=<IP address>
IPSEC_TUNNEL_IPMODE=4
IPSEC_REMOTE_ID=server.cisco
IPSEC_REMOTE_NETWORK=<IP address>
IPSEC_REMOTE_NETMASK=<mask>
IPSEC_REMOTE_NETWORK2=
IPSEC_REMOTE_NETMASK2=
IPSEC_REMOTE_PROTOPORT=
IPSEC_LOCAL_ID=client.router
IPSEC_LOCAL_NETWORK=<IP address>
IPSEC_LOCAL_NETMASK=<mask>
IPSEC_LOCAL_NETWORK2=
IPSEC_LOCAL_NETMASK2=
IPSEC_LOCAL_PROTOPORT=
IPSEC_IKE_PROTOCOL=ikev2
IPSEC_IKE_ALG=manual
IPSEC_IKE_ENC=aes256
IPSEC_IKE_HASH=sha2_256
IPSEC_IKE_DH=modp2048
IPSEC_IKE_REAUTH=1
IPSEC_XAUTH_ENABLED=0
IPSEC_XAUTH_MODE=
IPSEC_XAUTH_USER=
IPSEC_XAUTH_PASS=
IPSEC_ESP_ALG=manual
IPSEC_ESP_ENC=aes256
IPSEC_ESP_HASH=sha2_256
IPSEC_PFS=0
IPSEC_PFS_DH=
IPSEC_KEY_LIFE=3600
IPSEC_IKE_LIFE=3600
IPSEC_REKEY_MARGIN=540
IPSEC_REKEY_FUZZ=100
IPSEC_DPD_DELAY=20
IPSEC_DPD_TIMEOUT=60
IPSEC_ENCAP=tunnel
IPSEC_FORCE_ENCAPS=0
IPSEC_AGGRESSIVE=0
IPSEC_AUTHBY=rsa
IPSEC_PSK=
IPSEC_CA_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1.....
IPSEC_REMOTE_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0.....
IPSEC_LOCAL_CERT=LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0t.....
IPSEC_LOCAL_KEY=LS0tLS1CRUdJTiBSU0EgUFJJVkFURSBLRVktL.....
IPSEC_LOCAL_PASS=password
IPSEC_DEBUG=1
```

3.2.6 IKEv2 with Asymmetric Pre-Shared Key

This section describes how to set up an IKEv2 tunnel with an asymmetric pre-shared key between a Cisco device and an Advantech router. In this configuration, each side authenticates using a different pre-shared key: the Cisco device uses `cisco` as its local key and expects `router` from the remote peer, while the Advantech router uses `router` as its local key and expects `cisco` from the remote peer.

Cisco Device Configuration

```
!
aaa new-model
!
aaa authorization network FLEXVPN-AAA-AUTHORIZATION local
!
crypto ikev2 authorization policy ike_v2_policy
!
crypto ikev2 authorization policy IKE-AUTH-POLICY
 pool VPN-SPLIT-TUNNEL-ADDRESSES
 route set interface
!
crypto ikev2 proposal ike_v2_proposal
 encryption aes-gcm-256
 prf sha256
 group 21
!
crypto ikev2 policy ike_v2_policy
 proposal ike_v2_proposal
!
!
crypto ikev2 profile ike_v2_profile
 match identity remote any
 identity local fqdn server.cisco
 authentication remote pre-share key router
 authentication local pre-share key cisco
 aaa authorization group psk list FLEXVPN-AAA-AUTHORIZATION IKE-AUTH-POLICY
 virtual-template 20
!
crypto ipsec transform-set aes-gcm esp-gcm 256
 mode transport
!
crypto ipsec profile FlexVPN
 set security-policy limit 100
 set transform-set aes-gcm
 set pfs group21
 set ikev2-profile ike_v2_profile
 responder-only
!
interface Loopback2
 ip address 172.16.100.1 255.255.255.255
!
interface GigabitEthernet0/0/0
 ip address 10.40.29.128 255.255.252.0
 ip nat outside
 ip access-group 101 in
```

```
 negotiation auto
 spanning-tree portfast disable
!
interface GigabitEthernet0/0/1.202
 encapsulation dot1Q 202
 ip address 192.168.202.254 255.255.255.0
!
interface GigabitEthernet0
 vrf forwarding Mgmt-intf
 no ip address
 negotiation auto
!
interface Virtual-Template20 type tunnel
 ip unnumbered Loopback2
 no ip redirects
 tunnel source 10.40.29.128
 tunnel mode ipsec ipv4
 tunnel protection ipsec profile FlexVPN
!
ip local pool VPN-SPLIT-TUNNEL-ADDRESSES 172.16.100.2 172.16.100.200
ip route 0.0.0.0 0.0.0.0 10.40.30.1
ip route 0.0.0.0 0.0.0.0 GigabitEthernet0/0/0
ip route 172.16.100.0 255.255.255.0 Null0
!
ip access-list extended FlexVPN_ACL
 permit ip 192.168.202.0 0.0.0.255 192.168.133.0 0.0.0.255
ip access-list extended NAT-ACL
 deny   ip any 192.168.1.0 0.0.0.255
access-list 20 permit 192.168.202.0 0.0.0.255
access-list 101 permit ip any any
access-list 101 permit esp any any
access-list 101 permit gre any any
access-list 101 permit icmp any any
!
!
```

Advantech Router Configuration

```
IPSEC_ENABLED=1
IPSEC_DESCRIPTION=”FlexVPN with asym. PSK”
IPSEC_TYPE=route
IPSEC_HOST_IPMODE=4
IPSEC_REMOTE_IPADDR=10.40.29.128
IPSEC_REMOTE_IPADDR2=
IPSEC_TUNNEL_IPMODE=4
IPSEC_REMOTE_ID=server.cisco
IPSEC_LOCAL_ID=client@router
IPSEC_INSTALL_ROUTES=0
IPSEC_REMOTE_NETWORK=0.0.0.0
IPSEC_REMOTE_NETMASK=0.0.0.0
IPSEC_REMOTE_NETWORK2=
IPSEC_REMOTE_NETMASK2=
IPSEC_REMOTE_PROTOPORT=
IPSEC_LOCAL_NETWORK=0.0.0.0
IPSEC_LOCAL_NETMASK=0.0.0.0
IPSEC_LOCAL_NETWORK2=
IPSEC_LOCAL_NETMASK2=
IPSEC_LOCAL_PROTOPORT=
IPSEC_MTU=1426
IPSEC_REMOTE_VIRTUAL_NETWORK=
IPSEC_REMOTE_VIRTUAL_MASK=
IPSEC_LOCAL_VIRTUAL_IP=0.0.0.0
IPSEC_CISCO_FLEXVPN=1
IPSEC_IKE_PROTOCOL=ikev2
IPSEC_IKE_ALG=manual
IPSEC_IKE_ENC=aes256gcm128
IPSEC_IKE_HASH=sha2_256
IPSEC_IKE_DH=ecp521
IPSEC_IKE_REAUTH=1
IPSEC_XAUTH_ENABLED=0
IPSEC_XAUTH_MODE=
IPSEC_XAUTH_USER=
IPSEC_XAUTH_PASS=
IPSEC_ESP_ALG=manual
IPSEC_ESP_ENC=aes256gcm128
IPSEC_ESP_HASH=
IPSEC_PFS=1
IPSEC_PFS_DH=ecp521
IPSEC_KEY_LIFE=3600
IPSEC_IKE_LIFE=3600
IPSEC_REKEY_MARGIN=540
IPSEC_REKEY_FUZZ=100
IPSEC_DPD_DELAY=10
IPSEC_DPD_TIMEOUT=20
IPSEC_ENCAP=tunnel
IPSEC_FORCE_ENCAPS=0
IPSEC_AGGRESSIVE=0
IPSEC_AUTHBY=secret
IPSEC_PSK=router
IPSEC_REMOTE_PSK=cisco
IPSEC_CA_CERT=
IPSEC_REMOTE_CERT=
IPSEC_LOCAL_CERT=
IPSEC_LOCAL_KEY=
IPSEC_LOCAL_PASS=
IPSEC_REVOCATION=
IPSEC_DEBUG=1
```

3.3 Windows Computer IPsec Tunnel with Advantech Router

This example uses **policy-based** mode. For a description of both VPN modes and their differences, see Section 2.1.

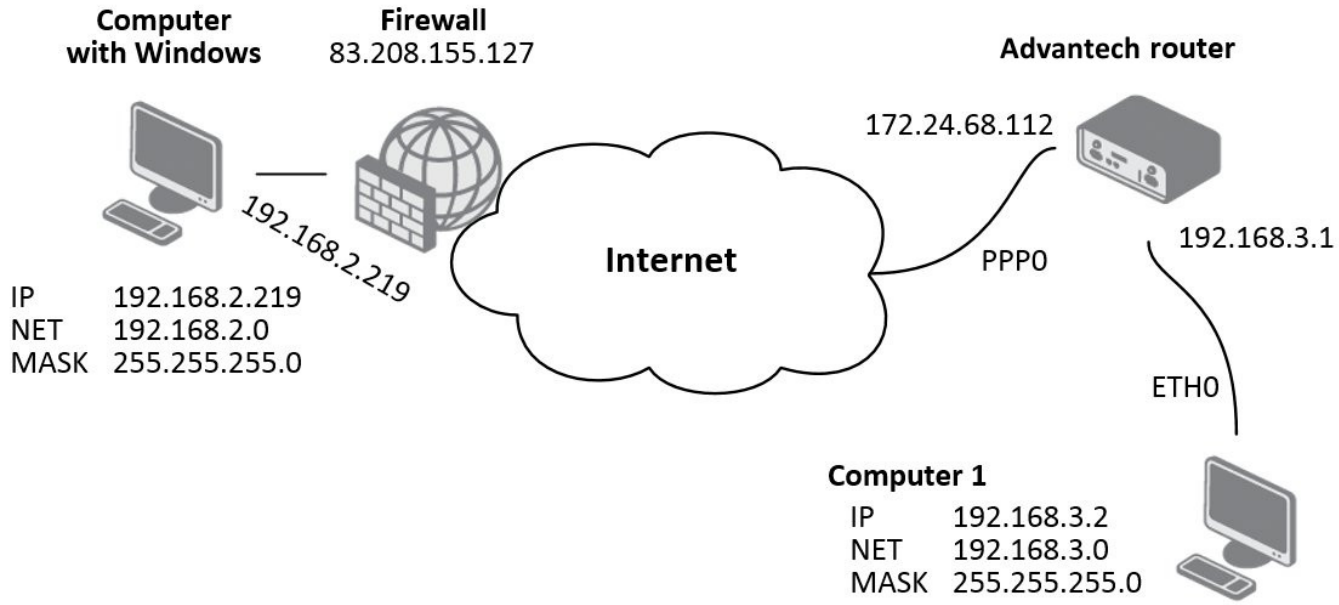

Figure 18: IPsec tunnel — Windows

The recommended IPsec client for Windows is *NCP Secure Entry Client*, which is the basis for the following configuration description.

3.3.1 Windows IPsec Configuration — NCP Secure Entry Client

The figure below shows the main window of NCP Secure Entry Client (version 9.32, build 218).

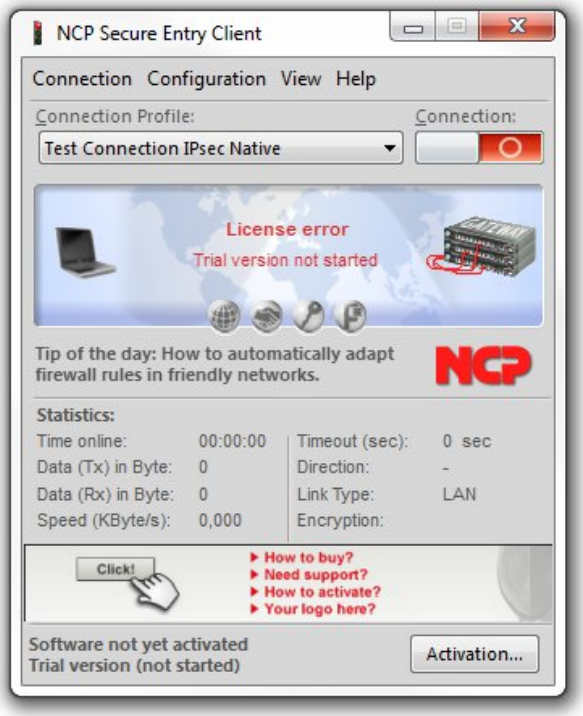

Figure 19: NCP Secure Entry Client

First, create a new profile for the IPsec tunnel. Select the *Configuration* tab and then select *Profiles*. The following window will open:

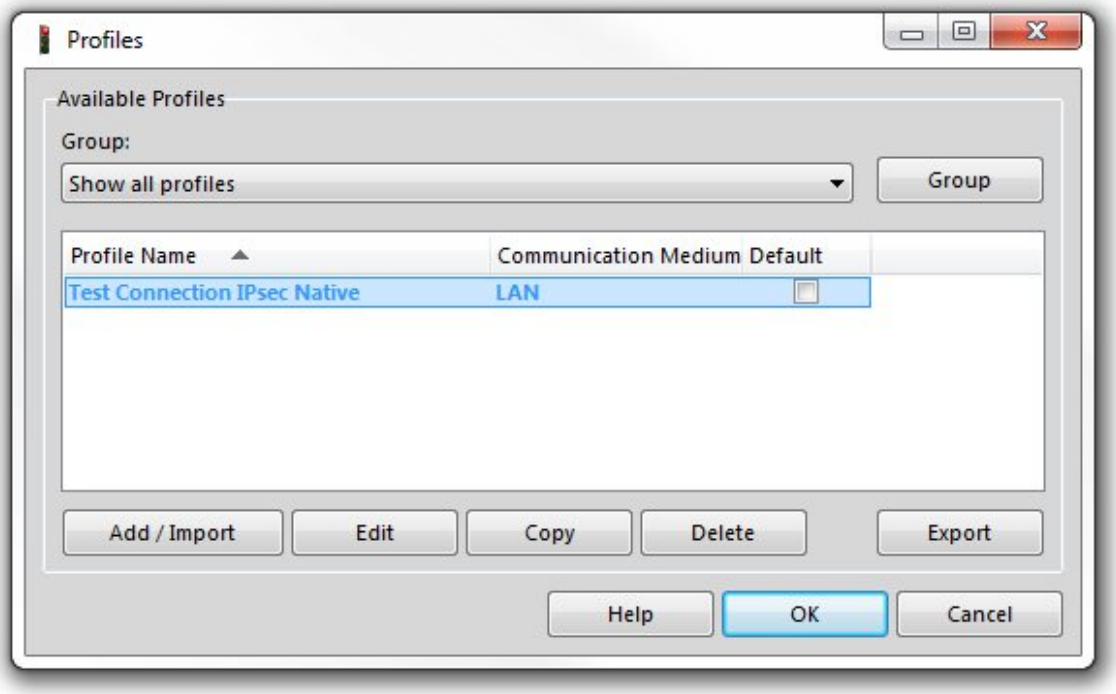

Figure 20: NCP Secure Entry Client — profiles

Click *Add/Import* to create a new profile. On the second screen, enter a profile name. The remaining screens can be confirmed with *Next* (or *Finish* on the last screen); all other settings will be configured later.

To configure the IPsec tunnel, select the profile and click *Edit*.

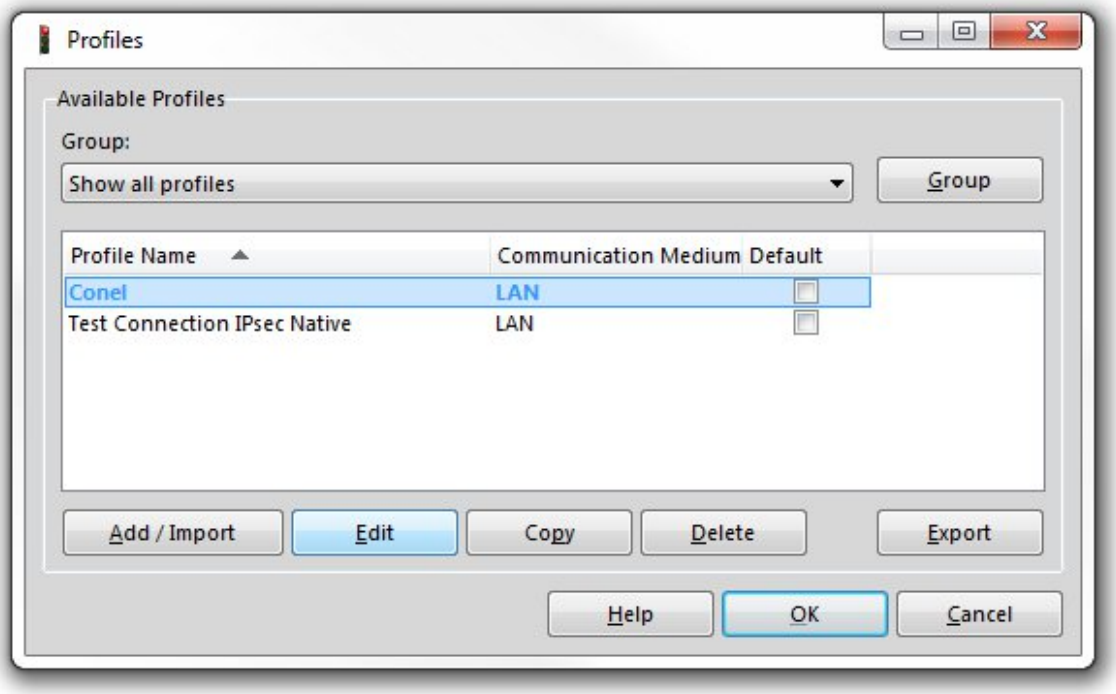

Figure 21: NCP Secure Entry Client — edit

Select *IPsec General Settings* in the left-hand menu, then click the *Policy Editor…* button on the right.

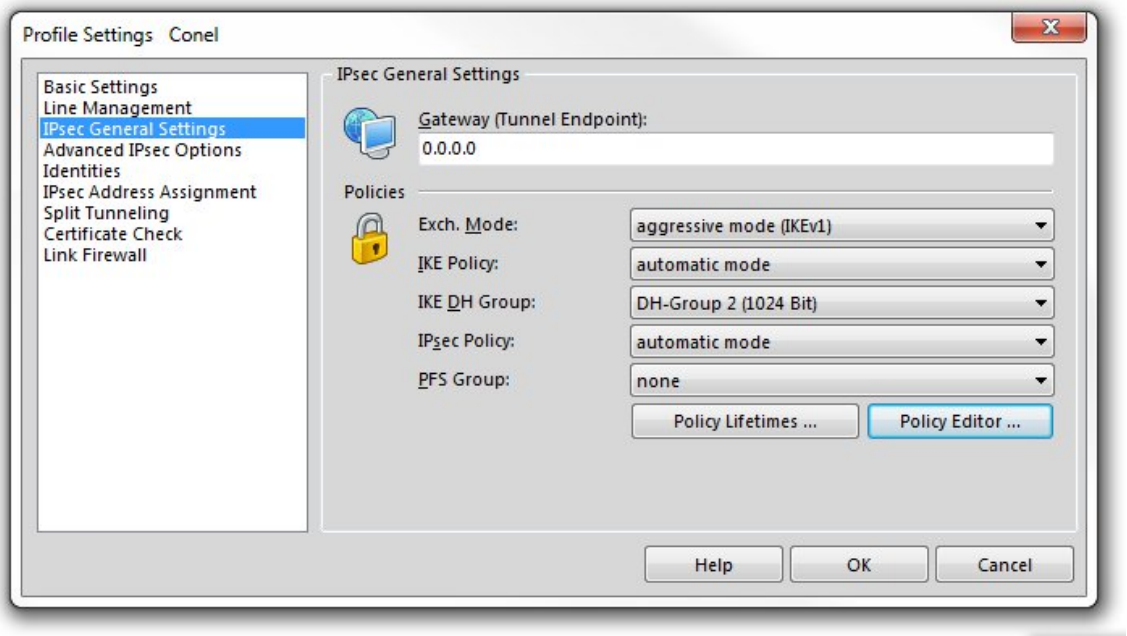

Figure 22: NCP Secure Entry Client — IPsec general settings

In the Policy Editor window, select the *Pre-shared Key* item under the *IKE Policy* section and click *Edit*.

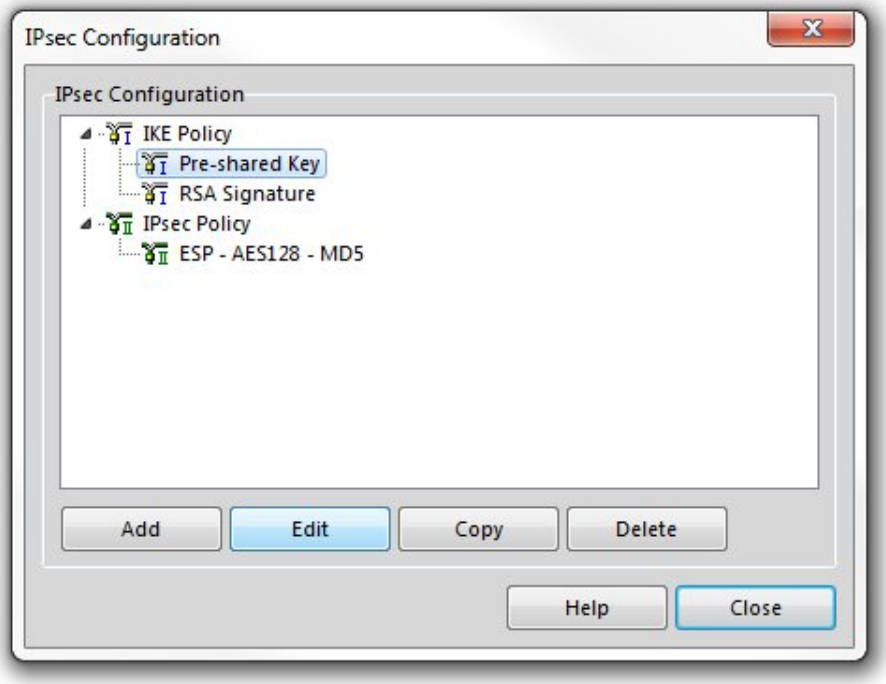

Figure 23: NCP Secure Entry Client — policy editor

In the window that opens, select the desired encryption and hash algorithms (for example *Triple DES* and *MD5*), then confirm by clicking *OK*.

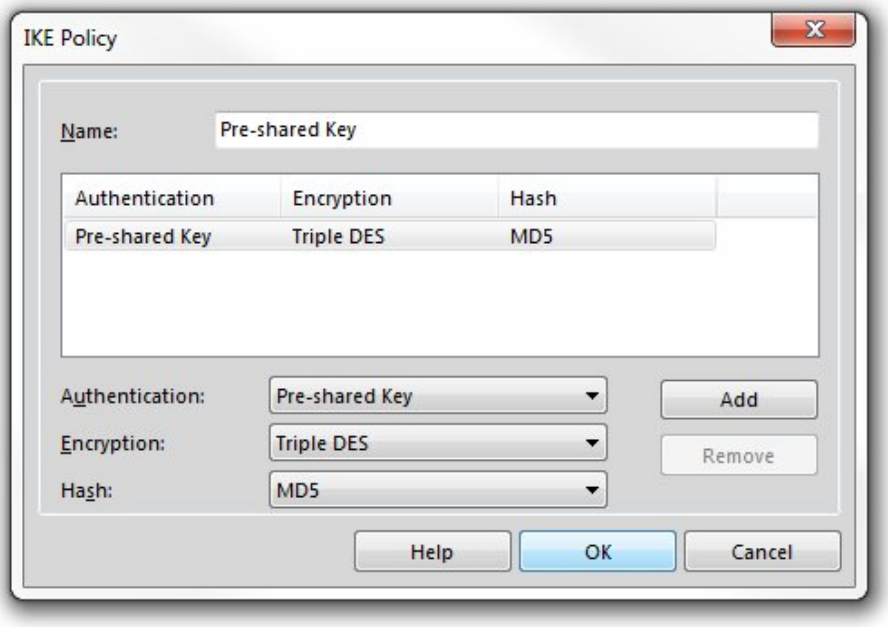

Figure 24: NCP Secure Entry Client — pre-shared key

Next, select the *ESP - AES128 - MD5* item in the *IPsec Policy* section and click *Edit*.

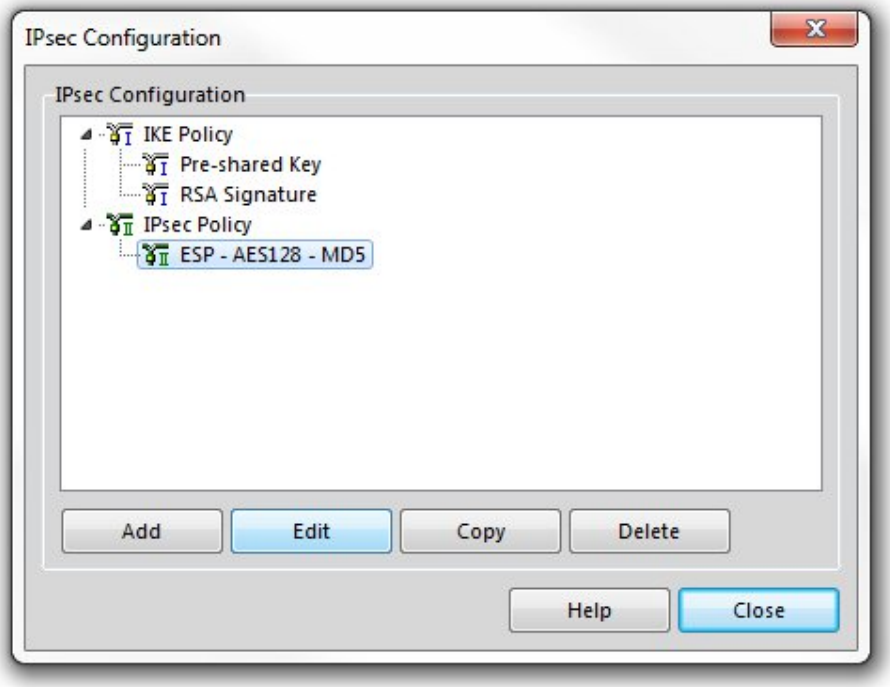

Figure 25: NCP Secure Entry Client — policy editor

In the new window, enter the desired policy name (for example *IPsec*) and select the encryption and hash algorithms (for example *Triple DES* and *MD5*). Confirm by clicking *OK*.

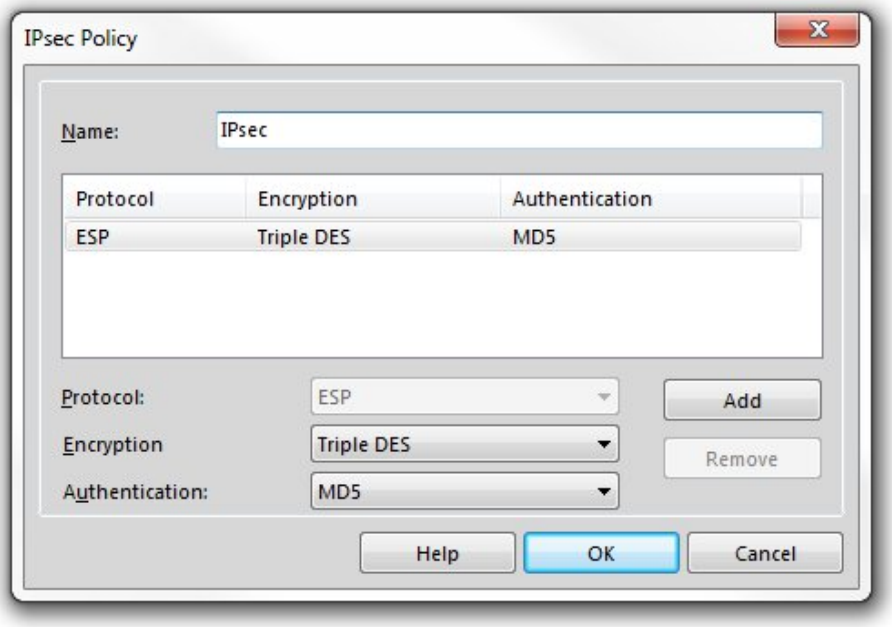

Figure 26: NCP Secure Entry Client — IPsec policy

Return to the main *IPsec General Settings* window and set the *IKE Policy* and *IPsec Policy* fields according to the configuration defined in the previous steps (see figure below). Set *IKE DH Group* to *DH-Group 2 (1024 bit)*.

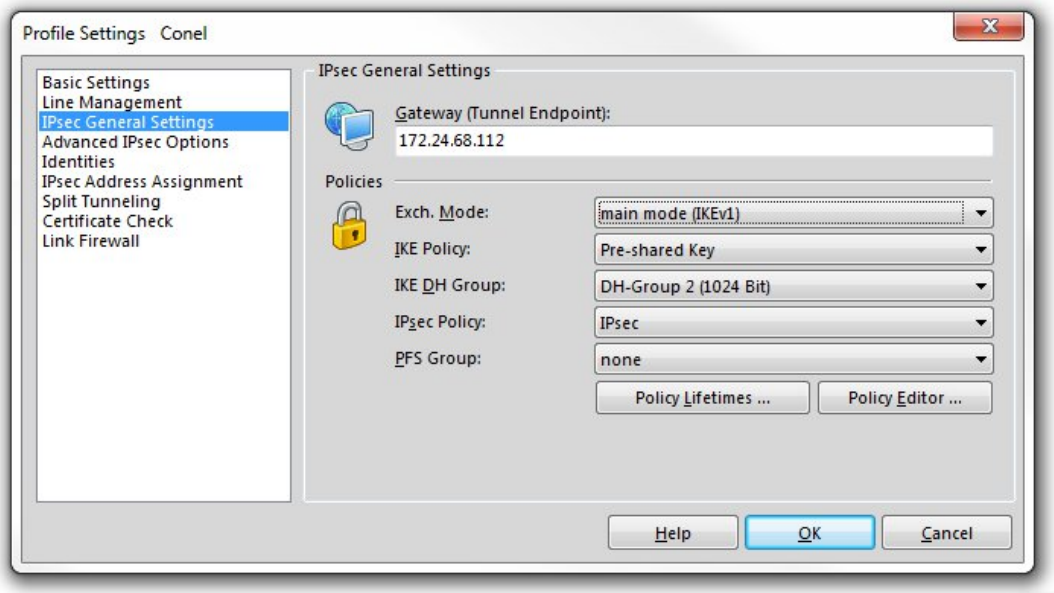

Figure 27: NCP Secure Entry Client — IPsec general settings

Select *Identities* in the left-hand menu and fill in the configuration form as shown below. Note that the IP address corresponds to the example scenario from the beginning of this section.

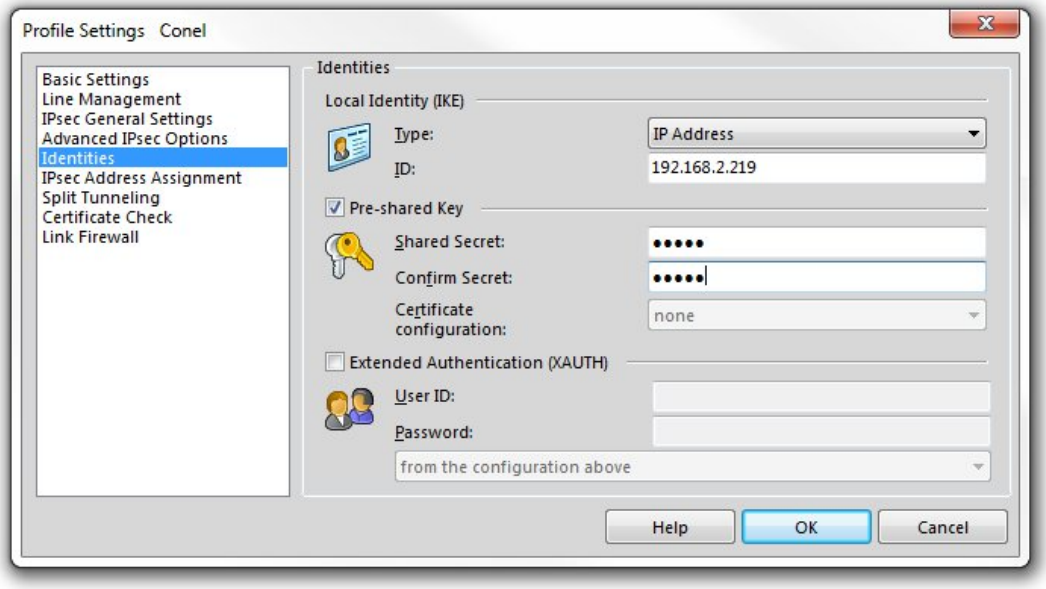

Figure 28: NCP Secure Entry Client — identities

The same IP address (192.168.2.219 in the example scenario) must also be entered on the *IPsec Address Assignment* page.

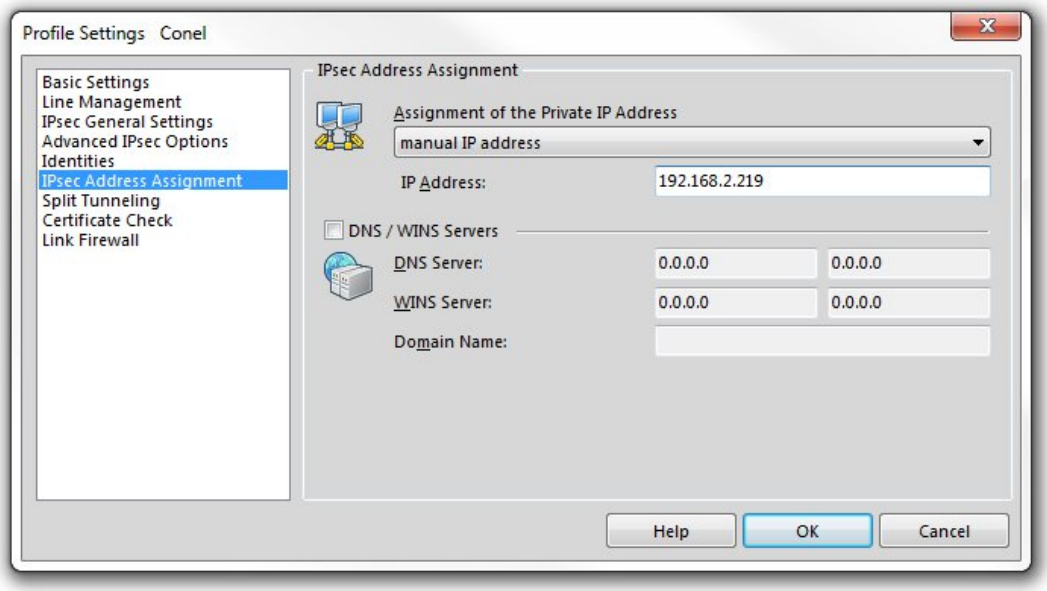

Figure 29: NCP Secure Entry Client — IPsec address assignment

On the *Split Tunneling* page, click *Add* and enter the IP address of the subnet behind the Advantech router (192.168.3.0 in the example scenario) and the corresponding subnet mask (255.255.255.0). Confirm by clicking *OK*.

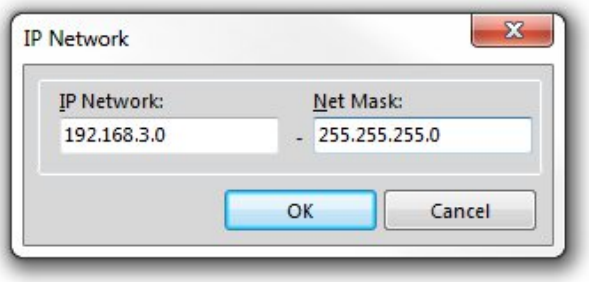

Figure 30: NCP Secure Entry Client — add IP network

The configured network is then listed in the *Split Tunneling* page.

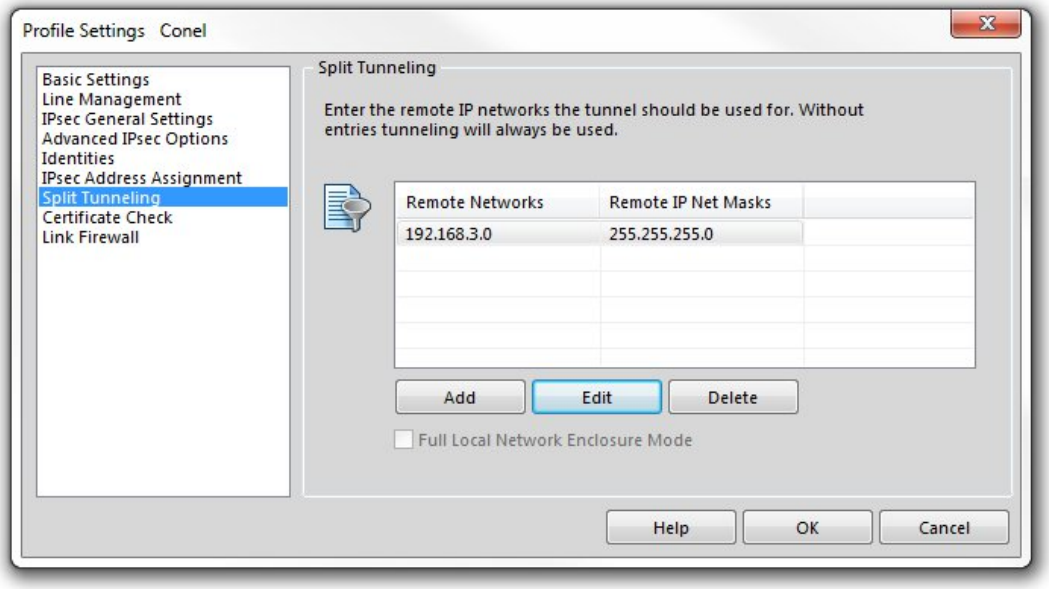

Figure 31: NCP Secure Entry Client — split tunneling

3.3.2 Advantech Router IPsec Configuration

The figure below shows the IPsec configuration page of the Advantech router with settings corresponding to the example scenario from the beginning of this section.

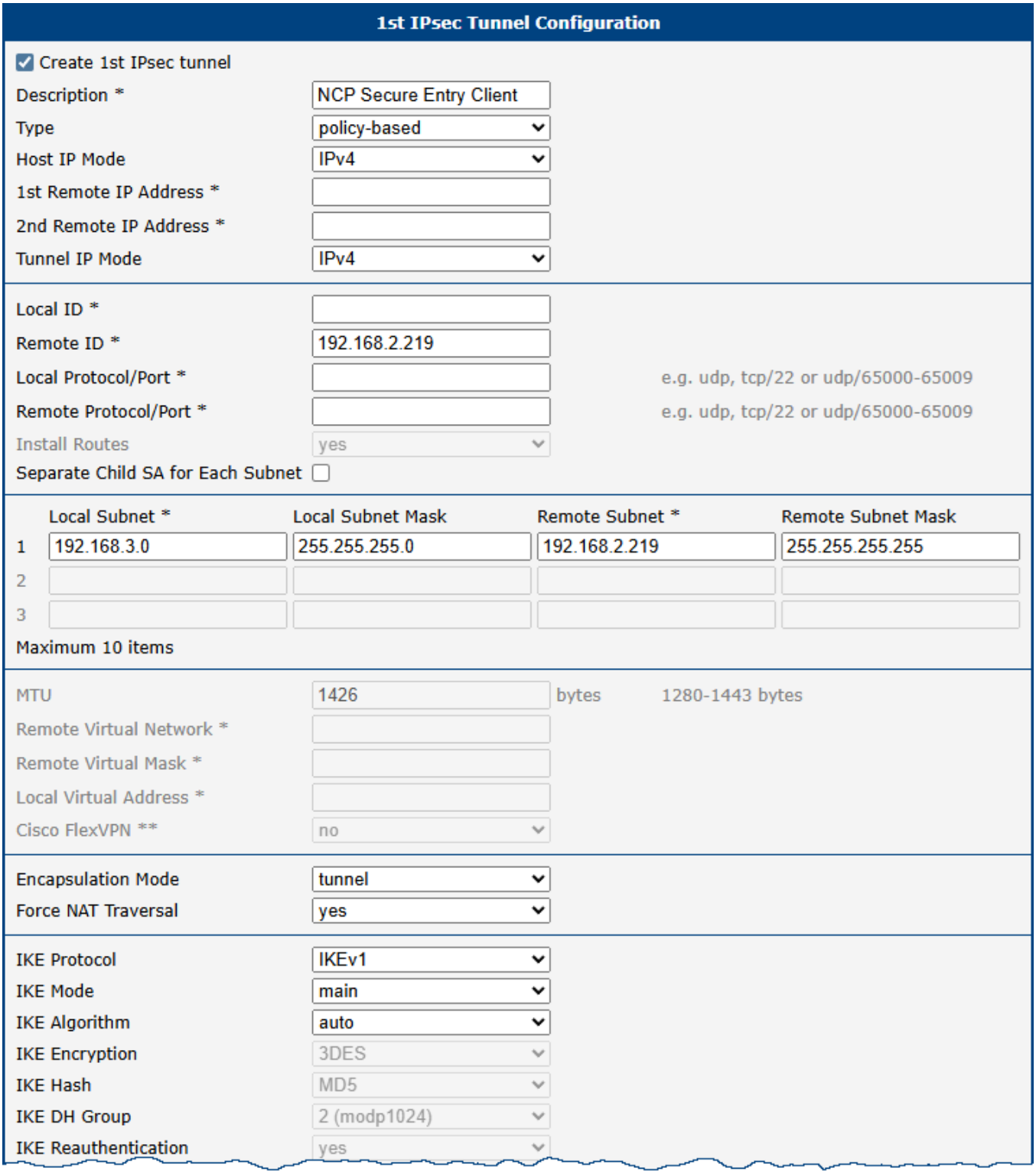

Figure 32: Advantech router IPsec configuration

3.4 Advanced IPsec Configurations

This section covers advanced IPsec configuration scenarios: Multiple Clients, Static Routes, and Dynamic Routing. The examples use route-based mode, but all scenarios are equally applicable to policy-based mode when the *Enabled Installing Routes* option is used — see Section 2.2 for details.

For further background on route-based VPN configuration, refer to the *Route-based VPNs* page on the strongSwan website.

3.4.1 Multiple Clients

This example demonstrates the configuration of multiple IPsec clients, where one Advantech router (IP 10.65.0.64) acts as the server and assigns IP addressed to all the clients (IP 10.64.0.65) on the network. For more information see the Virtual IP strongSwan webpage.

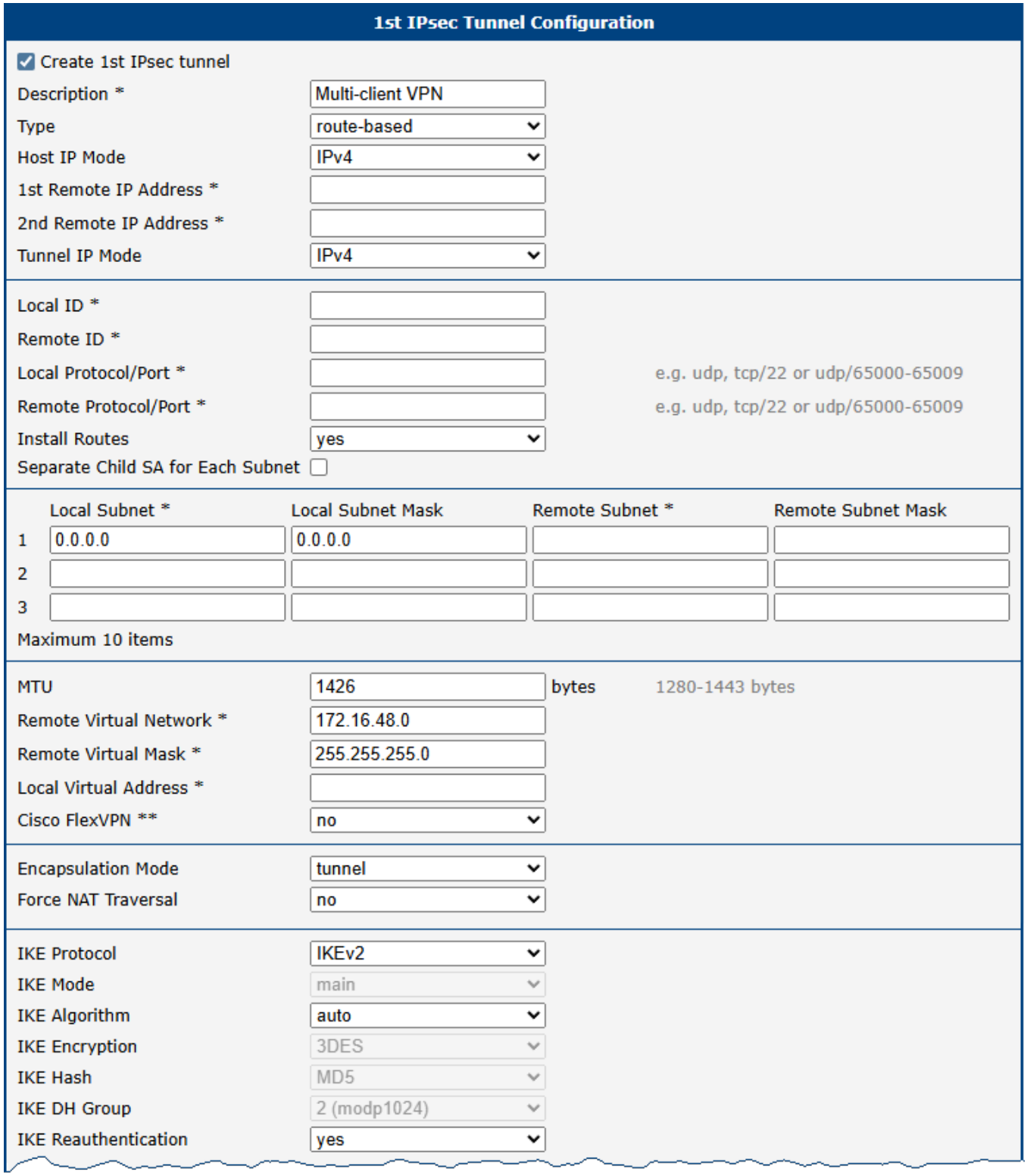

Figure 33: Server configuration

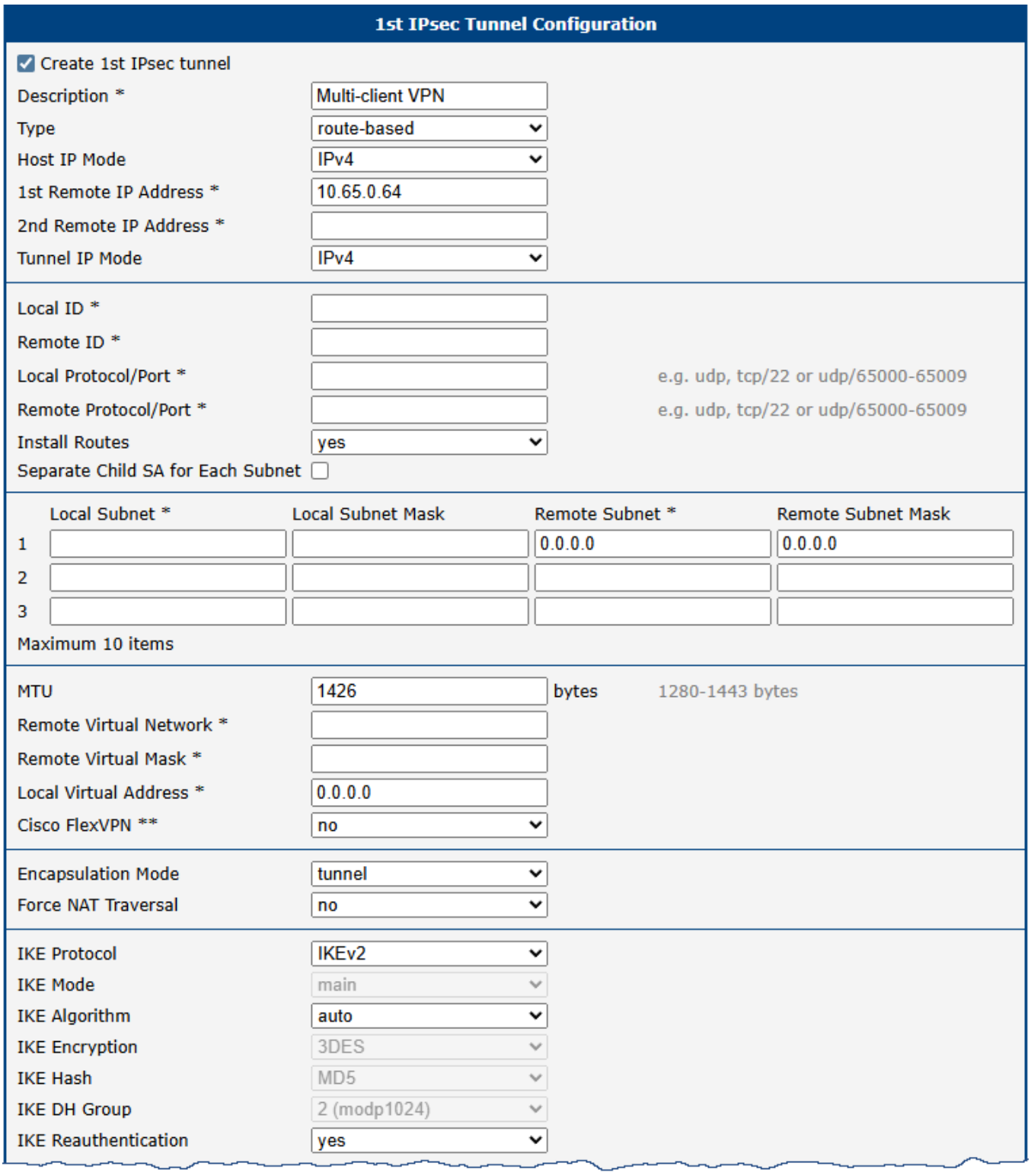

Figure 34: Client configuration

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 35 minutes, since May 10 08:35:02 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 6
IKE_SAs: 2 total, 0 half-open
mallinfo: sbrk 671744, mmap 0, used 465992, free 205752
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec1: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 0.0.0.0
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec1: TUNNEL, rekeying every 3060s
    local:  0.0.0.0/0
    remote: dynamic

Security Associations:

ipsec1: #3, ESTABLISHED, IKEv2, 99b579c0cd7af3a0_i 689b4c428785f7e5_r*
  local  '10.65.0.64' @ 10.65.0.64[4500]
  remote '10.65.0.65' @ 10.65.0.65[4500] [172.16.48.1]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 100s ago, reauth in 2837s
  ipsec1: #3, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 100s ago, rekeying in 2556s, expires in 3500s
    in  c3c1434b (-|0x00000001),      0 bytes,     0 packets
    out c10a9e02 (-|0x00000001),      0 bytes,     0 packets
    local  0.0.0.0/0
    remote 172.16.48.1/32
ipsec1: #2, ESTABLISHED, IKEv2, 8b9e0a6637a3c663_i 3bb02d38d4c3a93c_r*
  local  '10.65.0.64' @ 10.65.0.64[4500]
  remote '10.65.0.66' @ 10.65.0.66[4500] [172.16.48.2]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 2059s ago, reauth in 934s
  ipsec1: #2, reqid 2, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 2059s ago, rekeying in 840s, expires in 1541s
    in  c5197826 (-|0x00000001),   1176 bytes,    14 packets
    out cf5f7a8e (-|0x00000001),   1176 bytes,    14 packets,  1958s ago
    local  0.0.0.0/0
    remote 172.16.48.2/32


pool-ipsec1          172.16.48.0                       2 / 0 / 254
```

Figure 35: Server IPsec status

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 40 minutes, since May 10 08:34:50 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 7
IKE_SAs: 1 total, 0 half-open
mallinfo: sbrk 708608, mmap 0, used 577056, free 131552
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec1: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 10.65.0.64
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec1: TUNNEL, rekeying every 3060s
    local:  dynamic
    remote: 0.0.0.0/0

Security Associations:

ipsec1: #3, ESTABLISHED, IKEv2, 99b579c0cd7af3a0_i* 689b4c428785f7e5_r
  local  '10.65.0.65' @ 10.65.0.65[4500] [172.16.48.1]
  remote '10.65.0.64' @ 10.65.0.64[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 387s ago, reauth in 2009s
  ipsec1: #3, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 388s ago, rekeying in 2150s, expires in 3213s
    in  c10a9e02 (-|0x00000001),      0 bytes,     0 packets
    out c3c1434b (-|0x00000001),      0 bytes,     0 packets
    local  172.16.48.1/32
    remote 0.0.0.0/0
```

Figure 36: Client IPsec status

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.253.254	0.0.0.0	UG	0	0	0	usb0
10.64.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
10.65.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth1
172.16.48.1	0.0.0.0	255.255.255.255	UH	0	0	0	ipsec0
172.16.48.2	0.0.0.0	255.255.255.255	UH	0	0	0	ipsec0
192.168.253.254	0.0.0.0	255.255.255.255	UH	0	0	0	usb0

Figure 37: Server route table

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	0.0.0.0	128.0.0.0	U	0	0	0	ipsec0
0.0.0.0	192.168.253.254	0.0.0.0	UG	0	0	0	usb0
10.64.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
10.65.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth1
128.0.0.0	0.0.0.0	128.0.0.0	U	0	0	0	ipsec0
192.168.253.254	0.0.0.0	255.255.255.255	UH	0	0	0	usb0

Figure 38: Client route table

3.4.2 Static Routes

This example demonstrates the configuration of IPsec server (IP 10.64.0.64) and client (IP 10.64.0.65), where the routes are installed statically by *FRR/zebra* and *FRR/staticd* applications configured in the FRR Router App, which has to be installed and configured on both routers. For more information about the FRR, free software IP routing suite, see FRRouting User Guide.

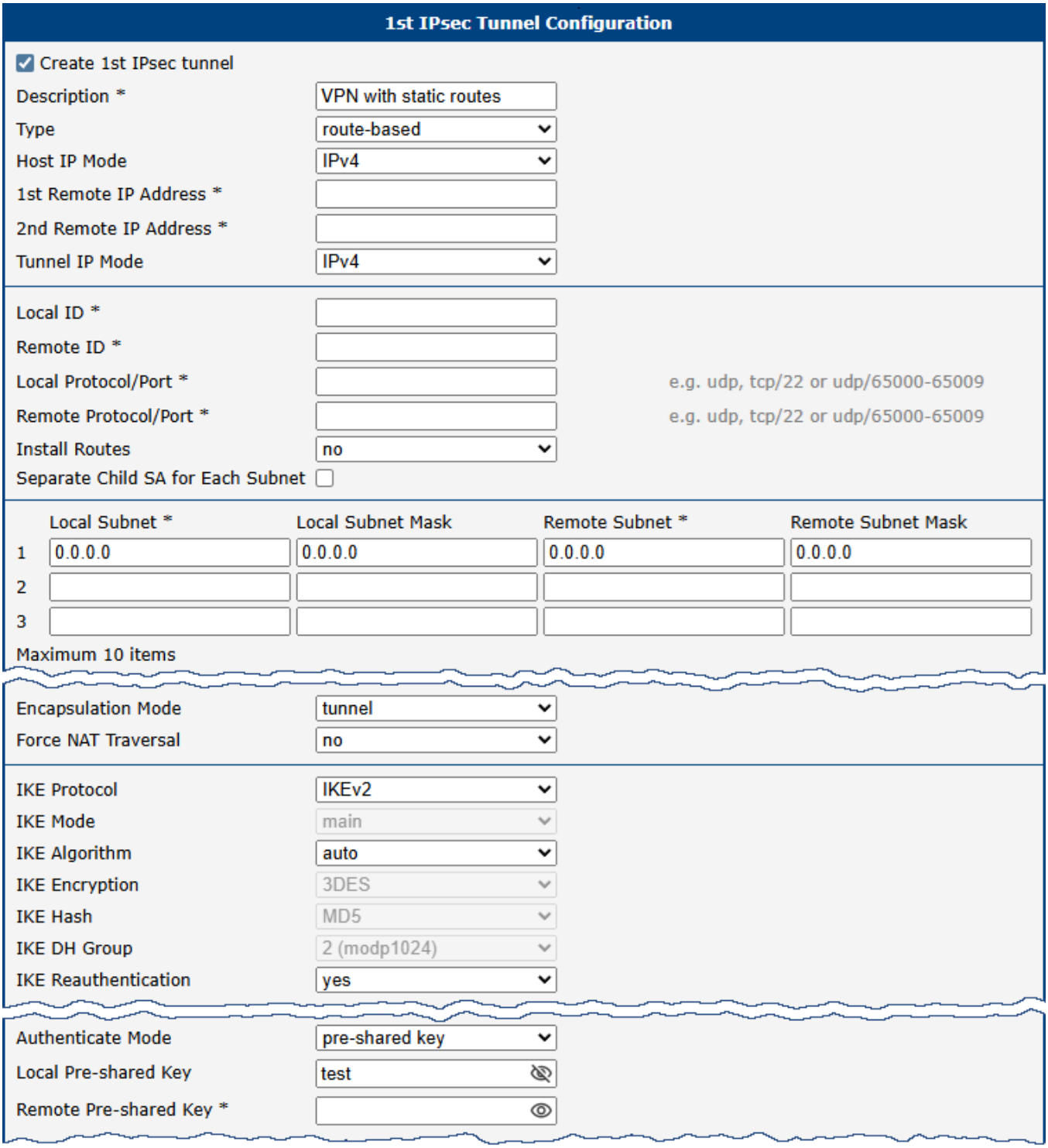

Figure 39: Server configuration

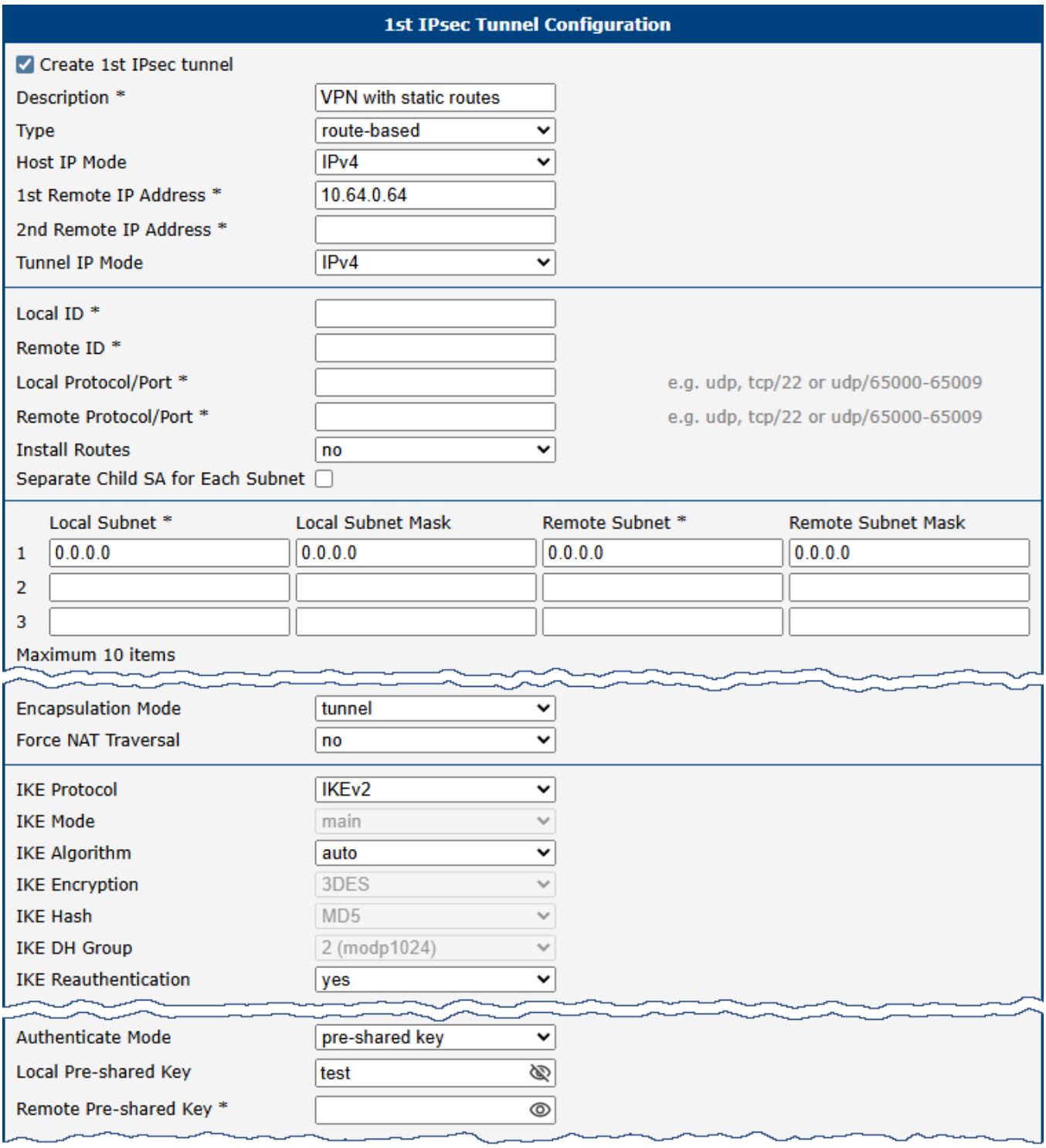

Figure 40: Client configuration

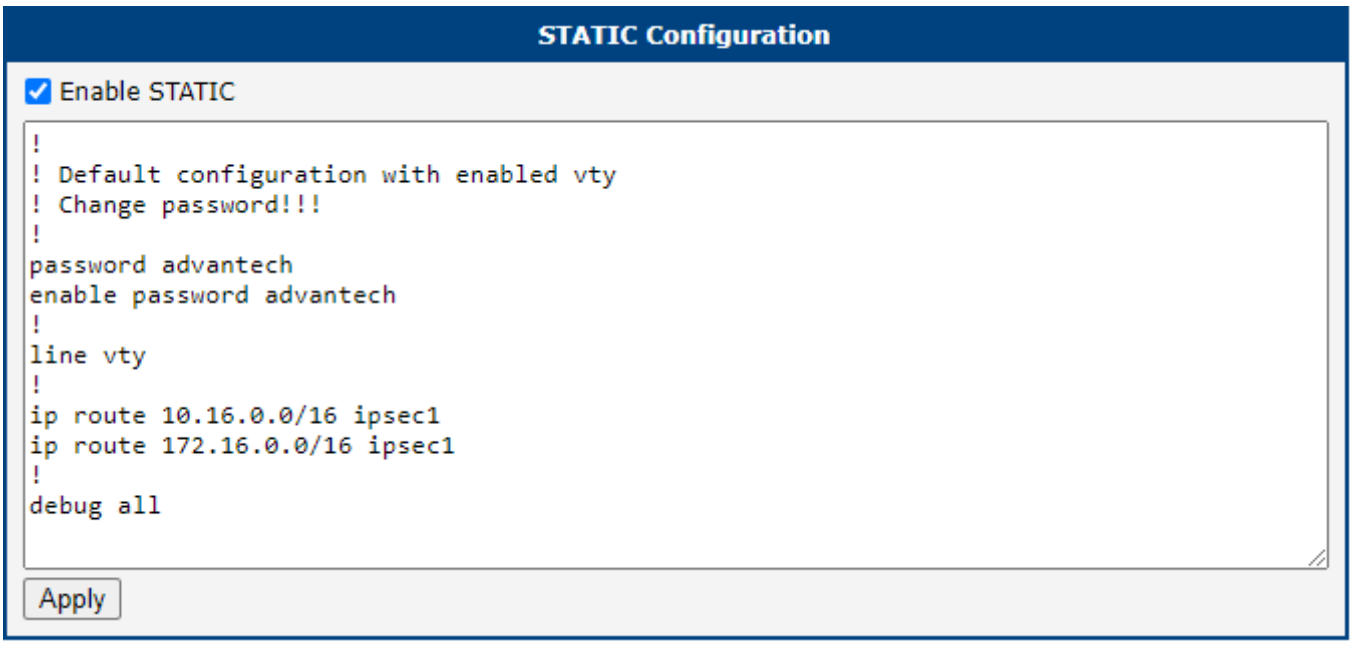

Figure 41: Server FRR static configuration

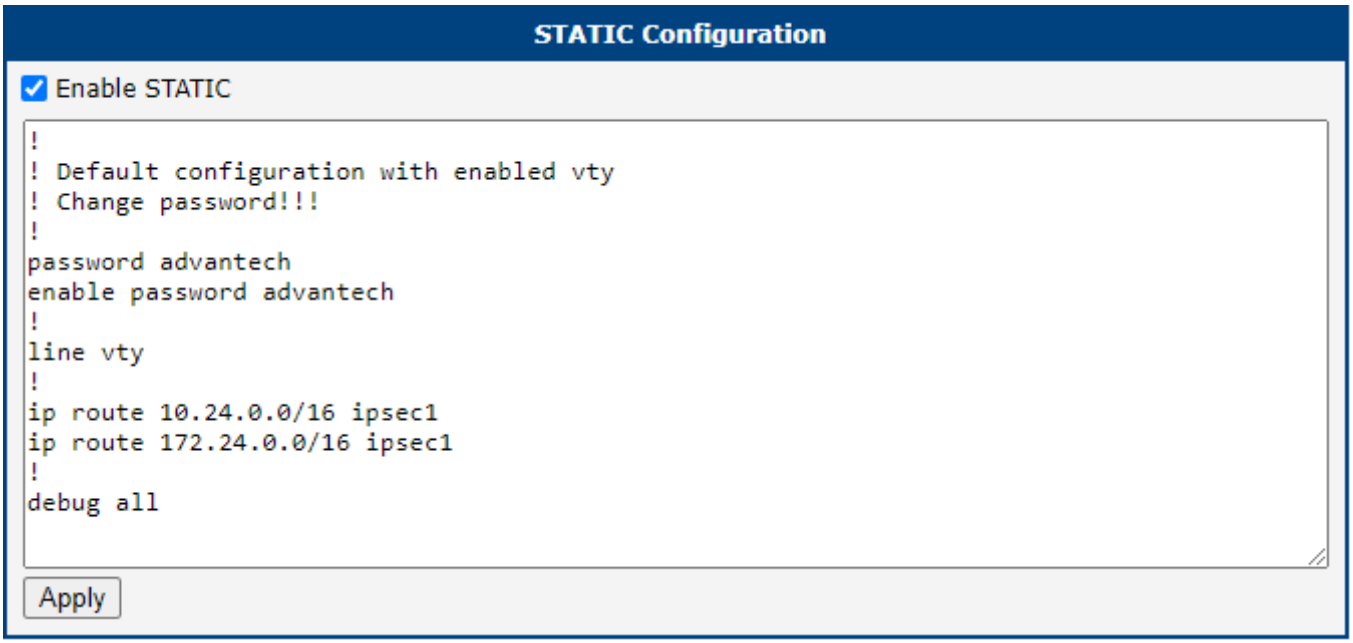

Figure 42: Client FRR static configuration

ZEBRA Configuration

Enable ZEBRA

```
!
! Default configuration with enabled vty
! Change password!!!
!
password conel
enable password conel
!
interface ipsec1
!
line vty
!
```

Apply

Figure 43: Client and server FRR Zebra configuration

Status Overview

Services

```
------------------------------------------------------------------------
Protocol zebra is running
------------------------------------------------------------------------
FRRouting 7.5 (Router).
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

S>* 10.16.0.0/16 [1/0] is directly connected, ipsec1, weight 1, 00:05:36
C>* 10.64.0.0/22 is directly connected, eth0, 00:37:12
C>* 10.65.0.0/22 is directly connected, eth1, 00:37:12
C>* 10.80.0.72/32 is directly connected, usb0, 00:37:12
S>* 172.16.0.0/16 [1/0] is directly connected, ipsec1, weight 1, 00:05:36
K>* 192.168.253.254/32 [0/0] is directly connected, usb0, 00:37:12
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 64:ff9b::/96 is directly connected, nat64, 00:37:12
C>* fd00:a40::/56 is directly connected, eth0, 00:37:12
C>* fd00:a41::/56 is directly connected, eth1, 00:37:12
C * fe80::/64 is directly connected, ipsec1, 00:05:36
C * fe80::/64 is directly connected, nat64, 00:37:12
C * fe80::/64 is directly connected, eth1, 00:37:12
C>* fe80::/64 is directly connected, eth0, 00:37:12
```

Figure 44: Server FRR status overview

Status Overview

Services

```
------------------------------------------------------------------------------
Protocol zebra is running
------------------------------------------------------------------------------
FRRouting 7.5 (Router).
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 10.0.7.150/32 is directly connected, usb0, 00:36:56
S>* 10.24.0.0/16 [1/0] is directly connected, ipsec1, weight 1, 00:05:40
C>* 10.64.0.0/22 is directly connected, eth0, 00:36:56
C>* 10.65.0.0/22 is directly connected, eth1, 00:36:56
S>* 172.24.0.0/16 [1/0] is directly connected, ipsec1, weight 1, 00:05:40
K>* 192.168.253.254/32 [0/0] is directly connected, usb0, 00:36:56
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 64:ff9b::/96 is directly connected, nat64, 00:36:56
C>* fd00:a40::/56 is directly connected, eth0, 00:36:56
C>* fd00:a41::/56 is directly connected, eth1, 00:36:56
C * fe80::/64 is directly connected, ipsec1, 00:05:40
C * fe80::/64 is directly connected, nat64, 00:36:56
C * fe80::/64 is directly connected, eth1, 00:36:56
C>* fe80::/64 is directly connected, eth0, 00:36:56
```

Figure 45: Client FRR status overview

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 111 minutes, since May 10 08:35:03 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 8
IKE_SAs: 1 total, 0 half-open
mallinfo: sbrk 733184, mmap 0, used 652480, free 80704
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec2: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 0.0.0.0
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec2: TUNNEL, rekeying every 3060s
    local:  0.0.0.0/0
    remote: 0.0.0.0/0

Security Associations:

ipsec2: #8, ESTABLISHED, IKEv2, fff77f54b0bfeda5_i 84ca9e337120c74b_r*
  local  '10.64.0.64' @ 10.64.0.64[4500]
  remote '10.64.0.65' @ 10.64.0.65[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 1646s ago, reauth in 1078s
  ipsec2: #6, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 1646s ago, rekeying in 1129s, expires in 1954s
    in  cf56c495 (-|0x00000002),      0 bytes,     0 packets
    out c1daa6f7 (-|0x00000002),      0 bytes,     0 packets
    local  0.0.0.0/0
    remote 0.0.0.0/0
```

Figure 46: Server IPsec status

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.253.254	0.0.0.0	UG	0	0	0	usb0
10.16.0.0	0.0.0.0	255.255.0.0	U	20	0	0	ipsec1
10.64.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
10.65.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth1
172.16.0.0	0.0.0.0	255.255.0.0	U	20	0	0	ipsec1
192.168.253.254	0.0.0.0	255.255.255.255	UH	0	0	0	usb0

Figure 47: Server route table

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 70 minutes, since May 10 09:58:36 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 3
IKE_SAs: 1 total, 0 half-open
mallinfo: sbrk 745472, mmap 0, used 626296, free 119176
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec2: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 10.64.0.64
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec2: TUNNEL, rekeying every 3060s
    local:  0.0.0.0/0
    remote: 0.0.0.0/0

Security Associations:

ipsec2: #2, ESTABLISHED, IKEv2, 97722e1fac5db468_i* 2ec31fcec00ae96a_r
  local  '10.64.0.65' @ 10.64.0.65[4500]
  remote '10.64.0.64' @ 10.64.0.64[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 2065s ago, reauth in 172s
  ipsec2: #2, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 2066s ago, rekeying in 720s, expires in 1535s
    in  c960339f (-|0x00000002),      0 bytes,     0 packets
    out ca9dee4e (-|0x00000002),      0 bytes,     0 packets
    local  0.0.0.0/0
    remote 0.0.0.0/0
```

Figure 48: Client IPsec status

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.253.254	0.0.0.0	UG	0	0	0	usb0
10.24.0.0	0.0.0.0	255.255.0.0	U	20	0	0	ipsec1
10.64.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
10.65.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth1
172.24.0.0	0.0.0.0	255.255.0.0	U	20	0	0	ipsec1
192.168.253.254	0.0.0.0	255.255.255.255	UH	0	0	0	usb0

Figure 49: Client route table

3.4.3 Dynamic Routing

This example demonstrates the configuration of two routers, where the routes are installed dynamically by *FRR/zebra* and *FRR/BGP* applications configured in the FRR Router App, which has to be installed and configured on both routers. For more information about the FRR, free software IP routing suite, see FRRouting User Guide.

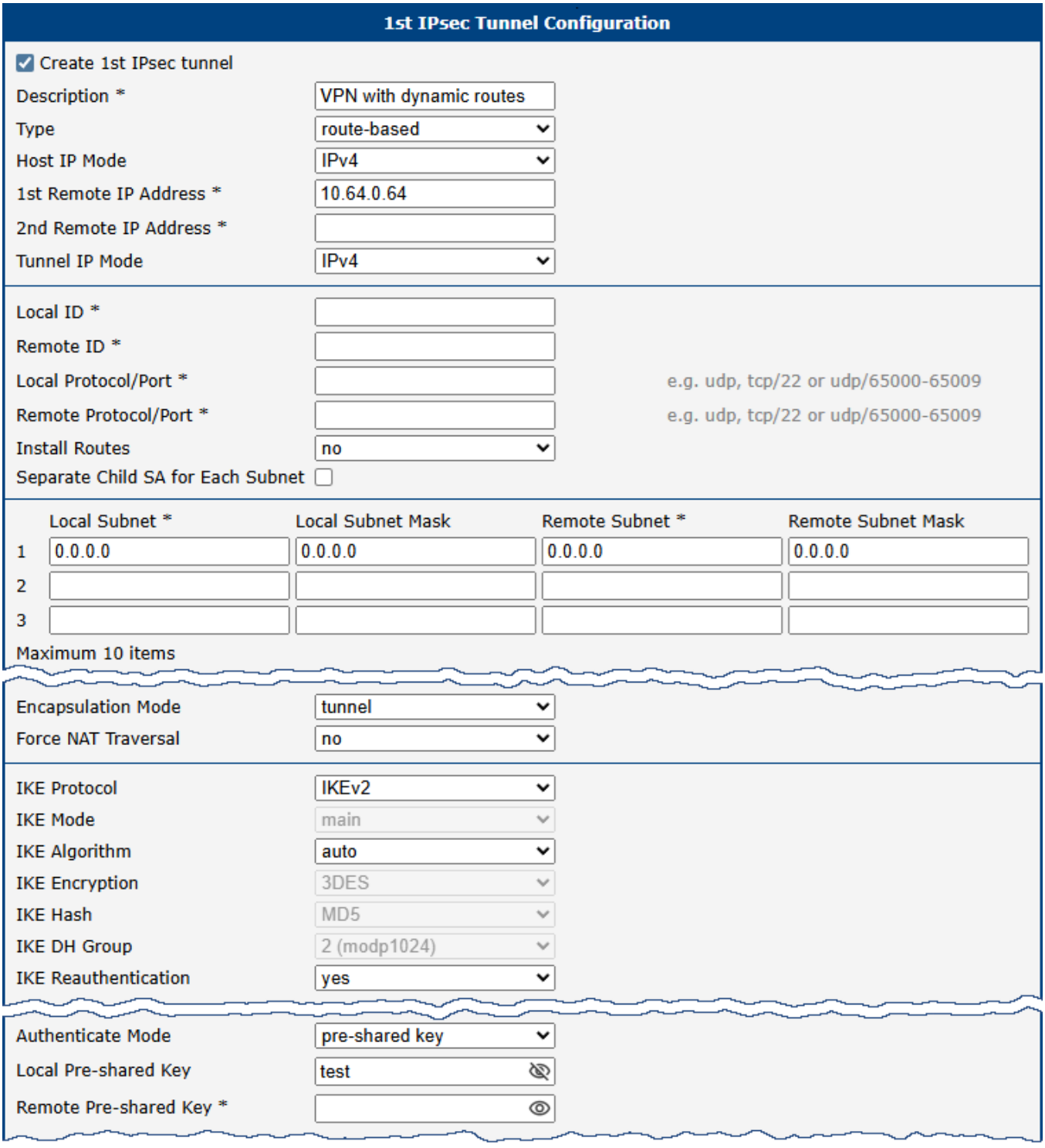

Figure 50: Client 1 configuration

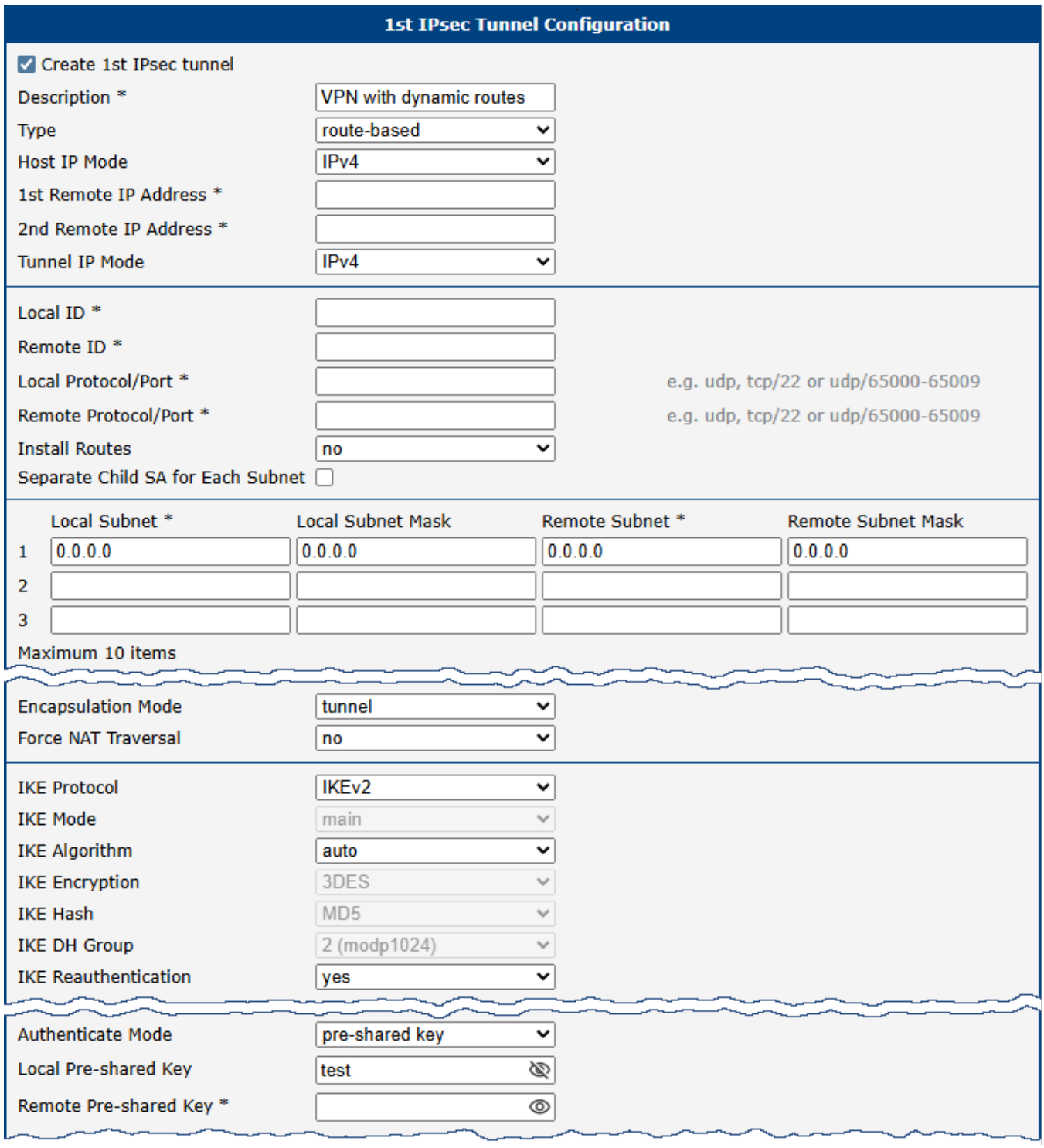

Figure 51: Client 2 configuration

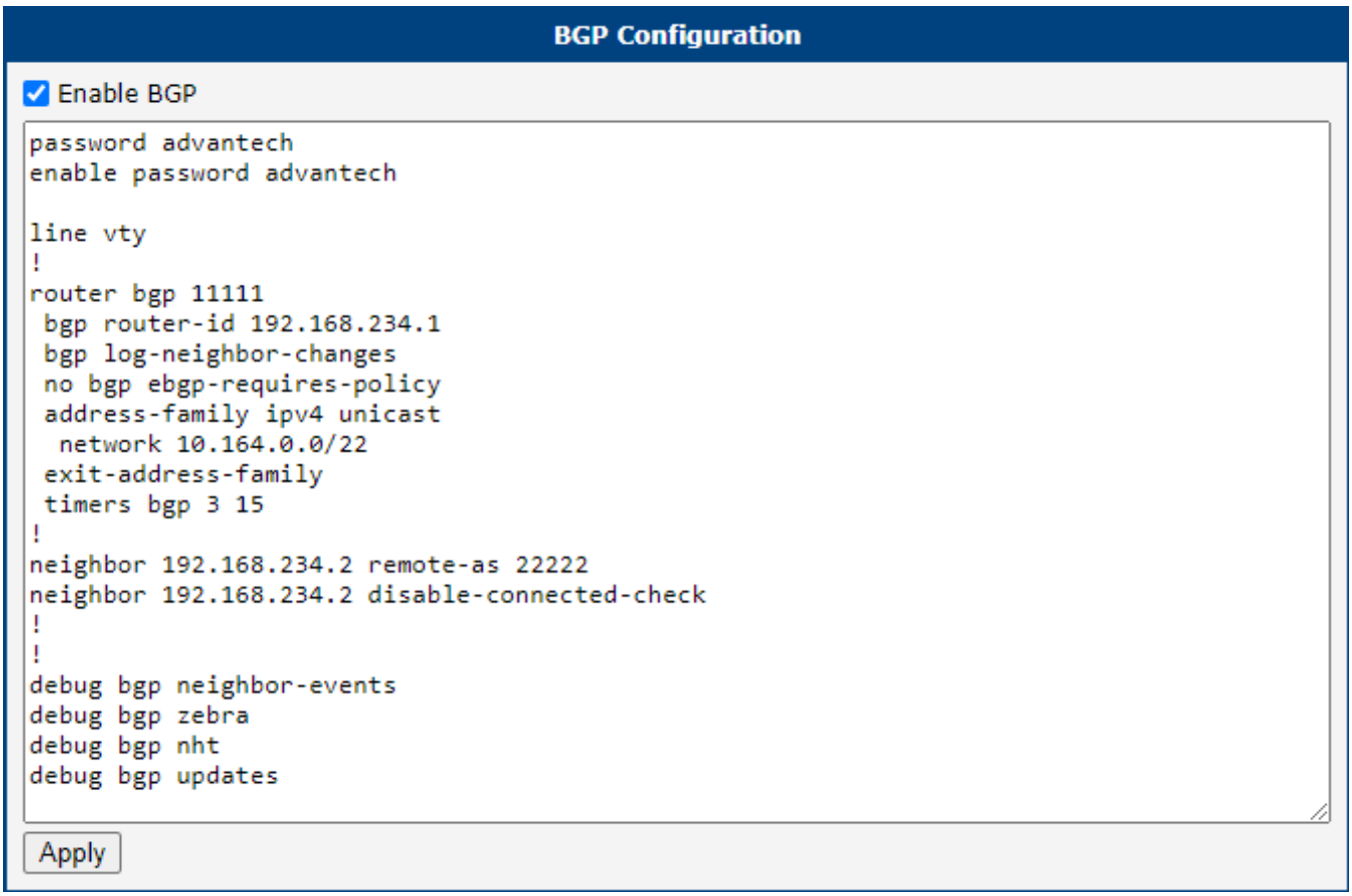

Figure 52: Client 1 FRR BGP configuration

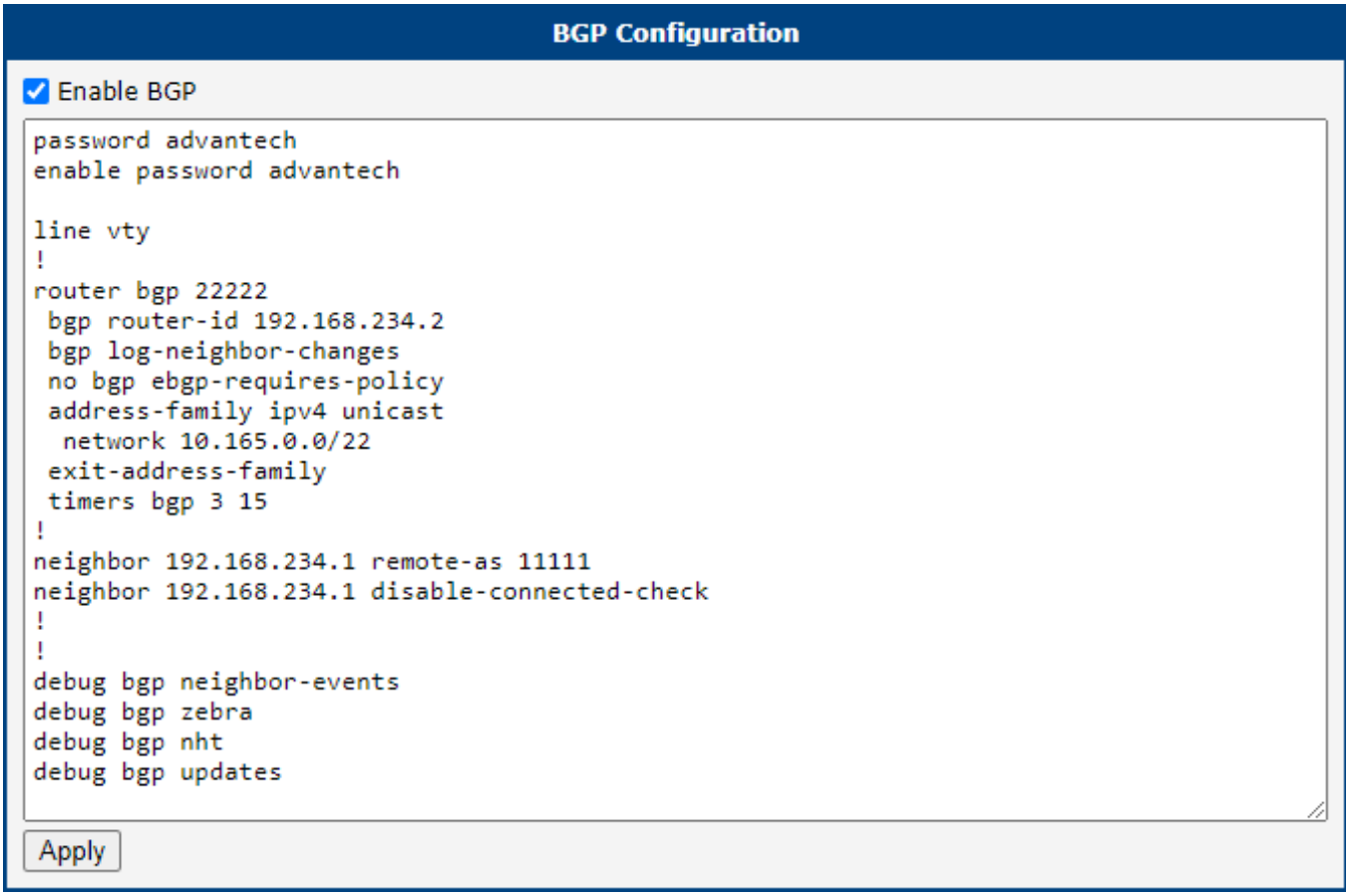

Figure 53: Client 2 FRR BGP configuration

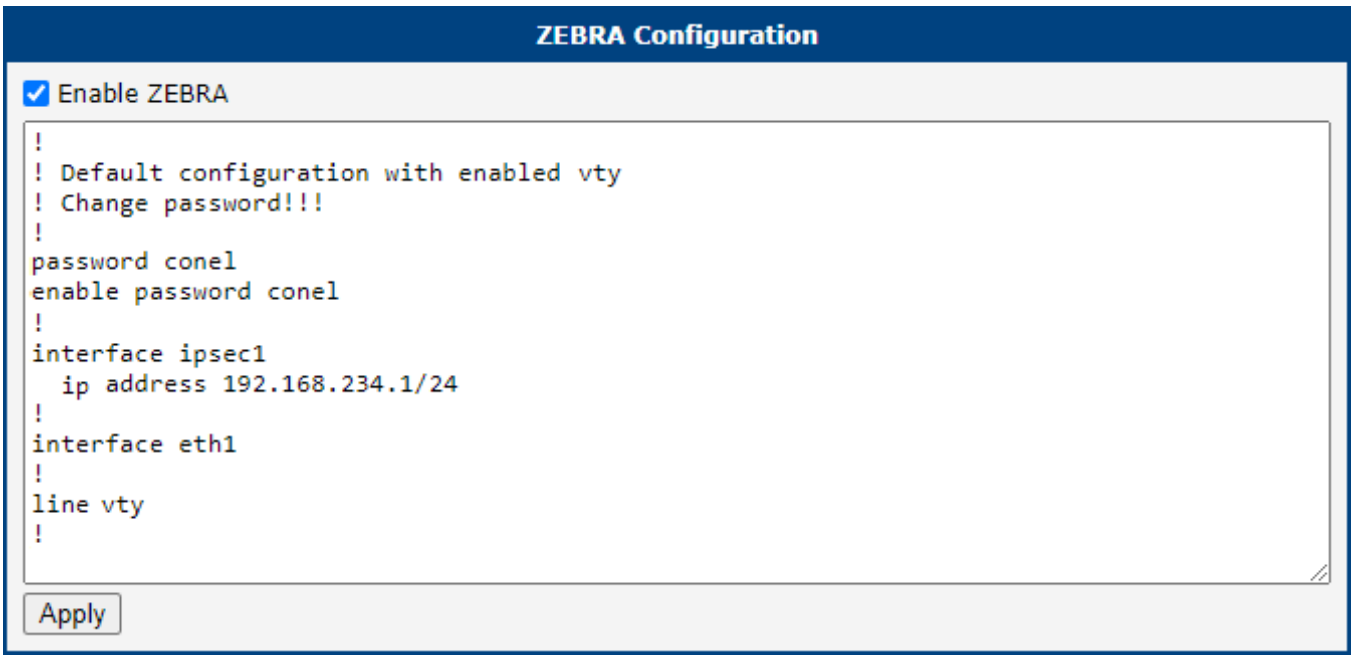

Figure 54: Client 1 FRR Zebra configuration

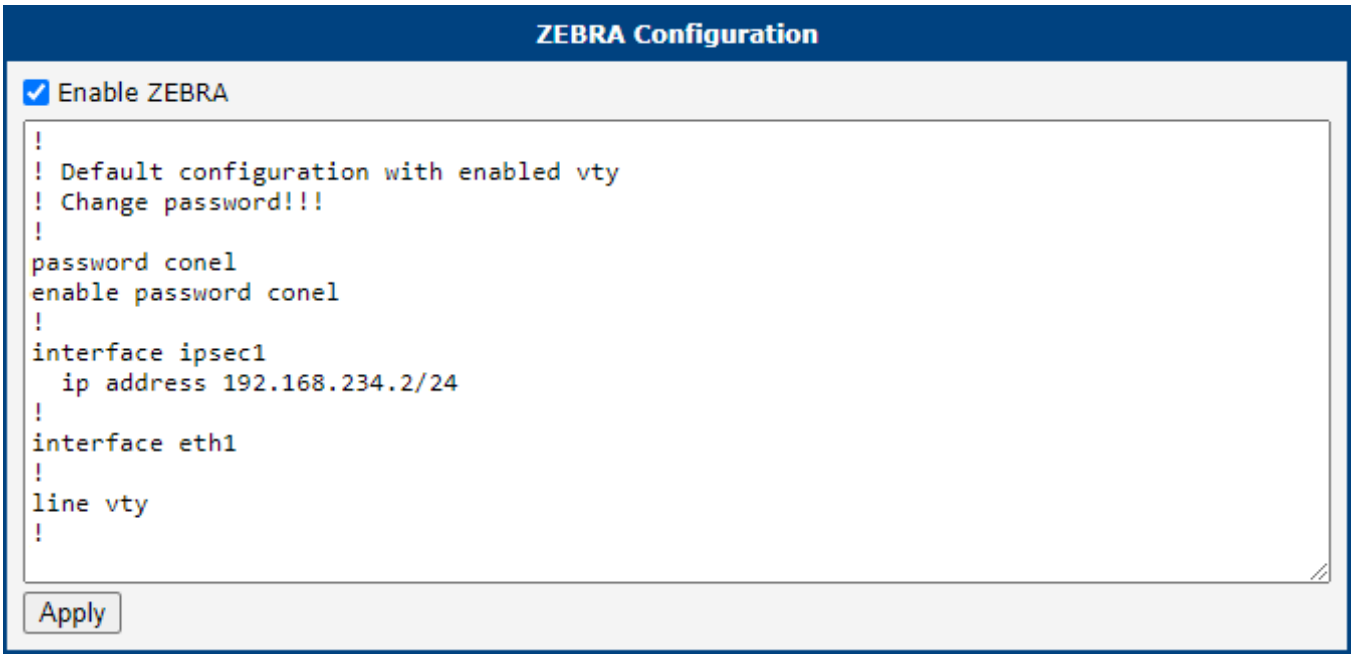

Figure 55: Client 2 FRR Zebra configuration

Status Overview

Services

```
------------------------------------------------------------------------------
Protocol zebra is running
------------------------------------------------------------------------------
FRRouting 7.5 (Router).
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backu

K>* 0.0.0.0/0 [0/0] via 192.168.253.254, usb0, 00:28:29
C>* 10.64.0.0/22 is directly connected, eth0, 00:28:29
C>* 10.80.0.72/32 is directly connected, usb0, 00:28:29
C>* 10.164.0.0/22 is directly connected, eth1, 00:28:29
B>* 10.165.0.0/22 [20/0] via 192.168.234.2, ipsec1, weight 1, 00:27:19
C>* 192.168.234.0/24 is directly connected, ipsec1, 00:28:29
K>* 192.168.253.254/32 [0/0] is directly connected, usb0, 00:28:29
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 64:ff9b::/96 is directly connected, nat64, 00:28:29
C>* fd00:a40::/56 is directly connected, eth0, 00:28:29
C>* fd00:a41::/56 is directly connected, eth1, 00:28:29
C * fe80::/64 is directly connected, ipsec1, 00:28:29
C * fe80::/64 is directly connected, nat64, 00:28:29
C * fe80::/64 is directly connected, eth1, 00:28:29
C>* fe80::/64 is directly connected, eth0, 00:28:29


------------------------------------------------------------------------------
Protocol nhrp is stopped
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Protocol bgp is running
------------------------------------------------------------------------------
Router# show ip bgp
BGP table version is 4, local router ID is 192.168.234.1, vrf id 0
Default local pref 100, local AS 11111
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.164.0.0/22    0.0.0.0                  0         32768 i
*> 10.165.0.0/22    192.168.234.2            0             0 22222 i

Displayed  2 routes and 2 total paths
```

Figure 56: Client 1 FRR status overview

Status Overview

Services

```
------------------------------------------------------------------------------
Protocol zebra is running
------------------------------------------------------------------------------
FRRouting 7.5 (Router).
Router# show ip route
Codes: K - kernel route, C - connected, S - static, R - RIP,
       O - OSPF, I - IS-IS, B - BGP, E - EIGRP, N - NHRP,
       T - Table, v - VNC, V - VNC-Direct, A - Babel, D - SHARP,
       F - PBR, f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

K>* 0.0.0.0/0 [0/0] via 192.168.253.254, usb0, 00:28:35
C>* 10.0.7.150/32 is directly connected, usb0, 00:28:35
C>* 10.64.0.0/22 is directly connected, eth0, 00:28:35
B>* 10.164.0.0/22 [20/0] via 192.168.234.1, ipsec1, weight 1, 00:27:15
C>* 10.165.0.0/22 is directly connected, eth1, 00:28:35
C>* 192.168.234.0/24 is directly connected, ipsec1, 00:28:35
K>* 192.168.253.254/32 [0/0] is directly connected, usb0, 00:28:35
Router# show ipv6 route
Codes: K - kernel route, C - connected, S - static, R - RIPng,
       O - OSPFv3, I - IS-IS, B - BGP, N - NHRP, T - Table,
       v - VNC, V - VNC-Direct, A - Babel, D - SHARP, F - PBR,
       f - OpenFabric,
       > - selected route, * - FIB route, q - queued, r - rejected, b - backup

C>* 64:ff9b::/96 is directly connected, nat64, 00:28:35
C>* fd00:a40::/56 is directly connected, eth0, 00:28:35
C>* fd00:a41::/56 is directly connected, eth1, 00:28:35
C * fe80::/64 is directly connected, ipsec1, 00:28:35
C * fe80::/64 is directly connected, nat64, 00:28:35
C * fe80::/64 is directly connected, eth1, 00:28:35
C>* fe80::/64 is directly connected, eth0, 00:28:35


------------------------------------------------------------------------------
Protocol nhrp is stopped
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Protocol bgp is running
------------------------------------------------------------------------------
Router# show ip bgp
BGP table version is 2, local router ID is 192.168.234.2, vrf id 0
Default local pref 100, local AS 22222
Status codes:  s suppressed, d damped, h history, * valid, > best, = multipath,
               i internal, r RIB-failure, S Stale, R Removed
Nexthop codes: @NNN nexthop's vrf id, < announce-nh-self
Origin codes:  i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 10.164.0.0/22    192.168.234.1            0             0 11111 i
*> 10.165.0.0/22    0.0.0.0                  0         32768 i

Displayed  2 routes and 2 total paths
```

Figure 57: Client 2 FRR status overview

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 37 minutes, since May 10 13:45:47 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 6
IKE_SAs: 1 total, 0 half-open
mallinfo: sbrk 688128, mmap 0, used 511256, free 176872
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec2: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 0.0.0.0
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec2: TUNNEL, rekeying every 3060s
    local:  0.0.0.0/0
    remote: 0.0.0.0/0

Security Associations:

ipsec2: #3, ESTABLISHED, IKEv2, b0acaf0bd7172747_i bb23ac60586d8534_r*
  local  '10.64.0.64' @ 10.64.0.64[4500]
  remote '10.64.0.65' @ 10.64.0.65[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 110s ago, reauth in 2502s
  ipsec2: #3, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 109s ago, rekeying in 2532s, expires in 3491s
    in  ccc1a077 (-|0x00000002),   5288 bytes,    84 packets
    out c76ae1f3 (-|0x00000002),   3476 bytes,    49 packets,     0s ago
    local  0.0.0.0/0
    remote 0.0.0.0/0
```

Figure 58: Client 1 IPsec status

Route Table

Destination	Gateway	Genmask	Flags	Metric	Ref	Use	Iface
0.0.0.0	192.168.253.254	0.0.0.0	UG	0	0	0	usb0
10.64.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth0
10.164.0.0	0.0.0.0	255.255.252.0	U	0	0	0	eth1
10.165.0.0	192.168.234.2	255.255.252.0	UG	20	0	0	ipsec1
192.168.234.0	0.0.0.0	255.255.255.0	U	0	0	0	ipsec1
192.168.253.254	0.0.0.0	255.255.255.255	UH	0	0	0	usb0

Figure 59: Client 1 route table

IPsec Status

IPsec Tunnels Information

```
Daemon Information:

strongSwan swanctl 5.9.2
uptime: 35 minutes, since May 10 13:47:27 2021
worker threads: 16 total, 11 idle, working: 4/0/1/0
job queues: 0/0/0/0
jobs scheduled: 5
IKE_SAs: 1 total, 0 half-open
mallinfo: sbrk 540672, mmap 0, used 444800, free 95872
loaded plugins: charon nonce revocation pubkey pem openssl curl kernel-netlink socket-default vici updown xauth-generic

Connections:

ipsec2: IKEv2, reauthentication every 3060s, no rekeying
  local:  0.0.0.0
  remote: 10.64.0.64
  local pre-shared key authentication:
  remote pre-shared key authentication:
  ipsec2: TUNNEL, rekeying every 3060s
    local:  0.0.0.0/0
    remote: 0.0.0.0/0

Security Associations:

ipsec2: #2, ESTABLISHED, IKEv2, b0acaf0bd7172747_i* bb23ac60586d8534_r
  local  '10.64.0.65' @ 10.64.0.65[4500]
  remote '10.64.0.64' @ 10.64.0.64[4500]
  AES_CBC-128/HMAC_SHA2_256_128/PRF_HMAC_SHA2_256/MODP_3072
  established 96s ago, reauth in 1976s
  ipsec2: #2, reqid 1, INSTALLED, TUNNEL, ESP:AES_CBC-128/HMAC_SHA1_96
    installed 97s ago, rekeying in 2564s, expires in 3504s
    in  c76ae1f3 (-|0x00000002),   3061 bytes,    43 packets,    95s ago
    out ccc1a077 (-|0x00000002),   4673 bytes,    74 packets,     2s ago
    local  0.0.0.0/0
    remote 0.0.0.0/0
```

Figure 60: Client 2 IPsec status

Route Table

```
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         192.168.253.254 0.0.0.0         UG    0      0        0 usb0
10.64.0.0       0.0.0.0         255.255.252.0   U     0      0        0 eth0
10.164.0.0      192.168.234.1   255.255.252.0   UG    20     0        0 ipsec1
10.165.0.0      0.0.0.0         255.255.252.0   U     0      0        0 eth1
192.168.234.0   0.0.0.0         255.255.255.0   U     0      0        0 ipsec1
192.168.253.254 0.0.0.0         255.255.255.255 UH    0      0        0 usb0
```

Figure 61: Client 2 route table

3.5 Known Issues

3.5.1 Several Subnets in one CHILD_SA

When using IKEv2, multiple subnets can be combined into a single CHILD_SA, provided that both peers support this feature. Some devices do not — including those from Checkpoint, Cisco, and Fortinet. For details, see the interoperability[1] page on the strongSwan wiki.

Info

If you are using strongSwan with different IPsec solution, please consult `https://wiki.strongswan.org/projects/strongswan/wiki/Interoperability` in case of any problems before contacting our technical support.

[1] `https://wiki.strongswan.org/projects/strongswan/wiki/Interoperability`

4. Related Resources

You can obtain all product-related documents, software updates, and supplementary materials on the Advantech *Engineering Portal* at *icr.advantech.com*.

For easy access to specific resources, please refer to the following sections of the portal:

- **Router Support Materials:** To access your router's supporting documents (such as the *Hardware Manual* and *Configuration Manual*), the latest firmware, or other technical resources, navigate to *Support* → *Router Models*. Locate your specific model and select the appropriate tab under the *Documents to download* section. Available tabs include *Brochures*, *Manuals*, *Certificates*, *Firmware*, *Images/3D Models*, *PCN/SA*, and *Others*.
- **Router Apps:** To extend your router's functionality, installation packages and comprehensive manuals for various extension modules are available by navigating to *Download* → *Router Apps*.
- **Application Notes:** For detailed guides, configuration examples, and step-by-step instructions for implementing specific networking features and use cases, navigate to *Download* → *Application Notes*.
- **Development Documents:** If you are interested in custom scripting, programming your own applications, or compiling custom modules, navigate to *Development* page.

Appendix A: opensssl.conf

```
# OpenSSL configuration file for IPsec certificate generation.

HOME = .

[ ca ]
default_ca = CA_default

####################################################################
[ CA_default ]

dir               = ./
certs             = $dir                        # where issued certs are kept
crl_dir           = $dir                        # where issued CRLs are kept
database          = $dir/index.txt              # database index file
new_certs_dir     = $dir                        # default place for new certs
certificate       = $dir/ca.crt                 # the CA certificate
serial            = $dir/serial                 # the current serial number
crlnumber         = $dir/crlnumber              # the current CRL number; comment out to
↪  leave a V1 CRL
crl               = $dir/crl.pem                # the current CRL
private_key       = $dir/private/ca.key         # the private key
x509_extensions   = usr_cert                    # extensions to add to signed certs
name_opt          = ca_default                  # subject name options
cert_opt          = ca_default                  # certificate field options
default_days      = 365                         # how long to certify for
default_crl_days  = 30                          # how long before next CRL
default_md        = default                     # use public key default MD
preserve          = no                          # keep passed DN ordering
policy            = policy_match

[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

[ policy_anything ]
countryName             = optional
stateOrProvinceName     = optional
localityName            = optional
organizationName        = optional
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
default_bits        = 2048
default_keyfile     = privkey.pem
distinguished_name  = req_distinguished_name
attributes          = req_attributes
x509_extensions     = v3_ca                     # extensions to add to self-signed cert
string_mask         = utf8only



[ req_distinguished_name ]
```

```
countryName                     = Country Name (2 letter code)
countryName_default             = AU
countryName_min                 = 2
countryName_max                 = 2
stateOrProvinceName             = State or Province Name (full name)
stateOrProvinceName_default     = Some-State
localityName                    = Locality Name (eg, city)
0.organizationName              = Organization Name (eg, company)
0.organizationName_default      = Internet Widgits Pty Ltd
organizationalUnitName          = Organizational Unit Name (eg, section)
commonName                      = Common Name (e.g. server FQDN or YOUR name)
commonName_max                  = 64
emailAddress                    = Email Address
emailAddress_max                = 64

[ req_attributes ]
challengePassword       = A challenge password
challengePassword_min   = 4
challengePassword_max   = 20
unstructuredName        = An optional company name

#####################################################################
[ usr_cert ]
# Extensions added when 'ca' signs a request.
basicConstraints        = CA:FALSE
subjectKeyIdentifier    = hash
authorityKeyIdentifier = keyid,issuer

[ v3_req ]
# Extensions for a certificate request.
basicConstraints        = CA:FALSE
keyUsage                = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName          = @alt_names

[ alt_names ]
IP = <IP address>

[ v3_ca ]
# Extensions for a typical CA certificate.
subjectKeyIdentifier    = hash
authorityKeyIdentifier = keyid:always,issuer
basicConstraints        = critical,CA:true

[ crl_ext ]
# CRL extensions.
authorityKeyIdentifier = keyid:always
```

Appendix B: server_req.conf

```
# OpenSSL configuration file for server certificate request (server_req.conf).

[ ca ]
default_ca = CA_default

####################################################################
[ CA_default ]

dir               = ./
certs             = $dir                        # where issued certs are kept
crl_dir           = $dir                        # where issued CRLs are kept
database          = $dir/index.txt              # database index file
new_certs_dir     = $dir                        # default place for new certs
certificate       = $dir/ca.crt                 # the CA certificate
serial            = $dir/serial                 # the current serial number
crlnumber         = $dir/crlnumber              # the current CRL number; comment out to
↪  leave a V1 CRL
crl               = $dir/crl.pem                # the current CRL
private_key       = $dir/private/ca.key         # the private key
name_opt          = ca_default                  # subject name options
cert_opt          = ca_default                  # certificate field options
default_days      = 365                         # how long to certify for
default_crl_days  = 30                          # how long before next CRL
default_md        = default                     # use public key default MD
preserve          = no                          # keep passed DN ordering
policy            = policy_match

[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
distinguished_name = server
req_extensions     = v3_req
prompt             = no

[ server ]
C  = CZ
ST = Czechia
L  = Usti
O  = Advantech
OU = Advantech CZ
CN = server@cisco

[ v3_req ]
extendedKeyUsage = serverAuth
subjectAltName   = @alt_names

[ alt_names ]
IP    = 85.207.4.118
DNS   = server.cisco
email = server@cisco
```

Appendix C: client_req.conf

```
# OpenSSL configuration file for client certificate request (client_req.conf).

[ ca ]
default_ca = CA_default

####################################################################
[ CA_default ]

dir               = ./
certs             = $dir                        # where issued certs are kept
crl_dir           = $dir                        # where issued CRLs are kept
database          = $dir/index.txt              # database index file
new_certs_dir     = $dir                        # default place for new certs
certificate       = $dir/ca.crt                 # the CA certificate
serial            = $dir/serial                 # the current serial number
crlnumber         = $dir/crlnumber              # the current CRL number; comment out to
↪  leave a V1 CRL
crl               = $dir/crl.pem                # the current CRL
private_key       = $dir/private/ca.key         # the private key
name_opt          = ca_default                  # subject name options
cert_opt          = ca_default                  # certificate field options
default_days      = 365                         # how long to certify for
default_crl_days  = 30                          # how long before next CRL
default_md        = default                     # use public key default MD
preserve          = no                          # keep passed DN ordering
policy            = policy_match

[ policy_match ]
countryName             = match
stateOrProvinceName     = match
organizationName        = match
organizationalUnitName  = optional
commonName              = supplied
emailAddress            = optional

####################################################################
[ req ]
distinguished_name = client
req_extensions     = v3_req
prompt             = no

[ client ]
C  = CZ
ST = Czechia
L  = Usti
O  = Advantech
OU = Advantech CZ
CN = client@router

[ v3_req ]
extendedKeyUsage = clientAuth
subjectAltName   = @alt_names

[ alt_names ]
IP    = 62.141.23.118
DNS   = client.router
email = client@router
```