1 IPsec VPN – Basic functions

Document ID: 108944_en_00
Document designation: AH EN MGUARD IPSEC VPN OVERVIEW

Make sure you always use the latest documentation.
This is available to download at phoenixcontact.net/products.

Contents of this document

This document describes general application possibilities and the basic function of IPsec VPN connections.

1.1 Introduction

Data packets are normally sent via the Internet unprotected and therefore do not meet the basic security requirements:

– Encryption (data confidentiality)
– Authentication (proof of the identity of the sender)
– Integrity (assurance that the data packets have not been modified).

A *Virtual Private Network* (VPN) is a communications channel that uses encryption and authentication to protect data transmitted over a public medium (e.g. the Internet).

The most commonly used VPN protocol today is *Internet Protocol Security* (IPsec). Most VPN devices and clients are IPsec-compliant. IPsec is scalable and can be used in both small applications and in large VPM gateways with over 1,000 VPN connections.

IPsec supports transport connections that connect two individual hosts, as well as tunnel connections that connect two networks.

1.1.1 Setup of ISAKMP SA and IPsec SA

A VPN connection is established in two phases: Phase I (ISAKMP SA key exchange) and Phase II (IPsec-SA data exchange). SA stands for *Security Association*.

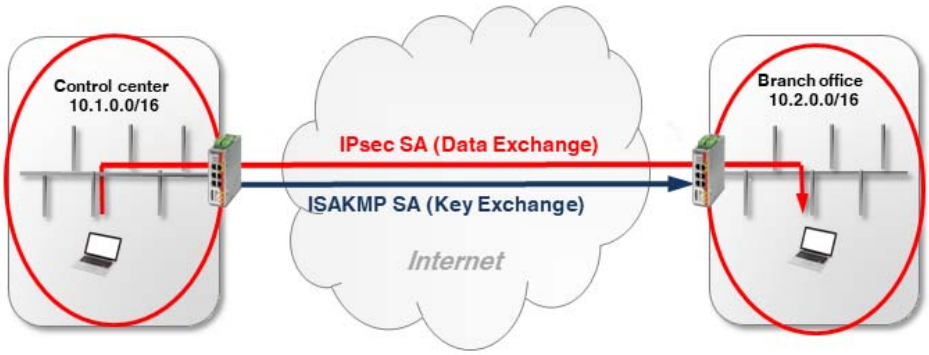


Figure 1-1 Setting up an IPsec connection (ISAKMP SA and IPsec SA)

Phase I (ISAKMP SA):

ISAKMP SA is a secure connection (*Security Association*) between two VPN peers, via which the secure exchange of the *keys* necessary for VPN encryption is agreed as the first step.

For this, both VPN peers negotiate the encryption and hash algorithm for phase I and authenticate one other mutually using *pre-shared keys* (PSK) or X.509 certificates (see Section 1.3).

Both peers then agree upon a *key* for encrypting the phase II data exchange.

Phase II (IPsec SA):

The IPsec SA (*Security Association*) is a secure connection via which the internal networks of the VPN peers are connected and data exchanged.

For this, both peers negotiate the encryption and hash algorithm for phase II and exchange information regarding the networks to be connected.

1.1.2 Configuration of IPsec VPN connections

The IPsec VPN connections between an mGuard device and a VPN peer are configured in the menu **IPsec VPN >> Connections** (see also mGuard firmware user manual). A VPN connection is normally *initiated* by a device, while the peer device *waits* for the connection request from the initiator.

The VPN connection is configured in the following tabs:

- **"General" tab**
- **"Authentication" tab**
- **"Firewall" tab**
- **"IKE Options" tab**

1.2 "General" tab

The settings in the "General" tab depend upon the network environment in which the VPN connection is established (e.g. network mode *Stealth, Router, PPPoE*) and on the VPN properties that are to be used (e.g. *1:1 NAT for local networks* or *hub and spoke*). See also Section 1 and 1.

1.2.1 Example

An encrypted IPsec VPN tunnel is to be established between **company network 1** (192.168.1.0/24) and **company network 2** (192.168.2.0/24). The VPN connection is initiated by *mGuard 1*. Both devices are operated in the *Router* network mode.

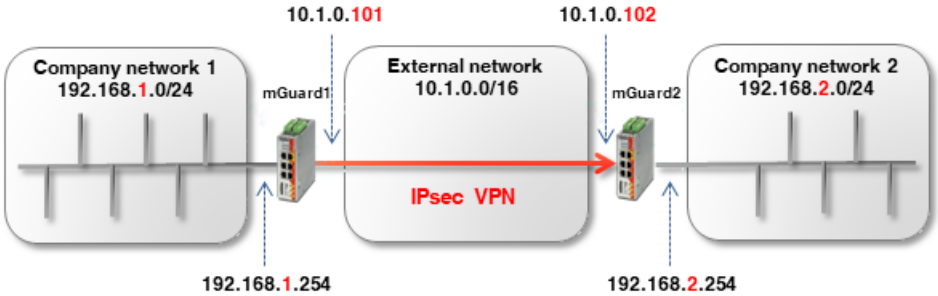


Figure 1-2 Connecting two networks via IPsec VPN

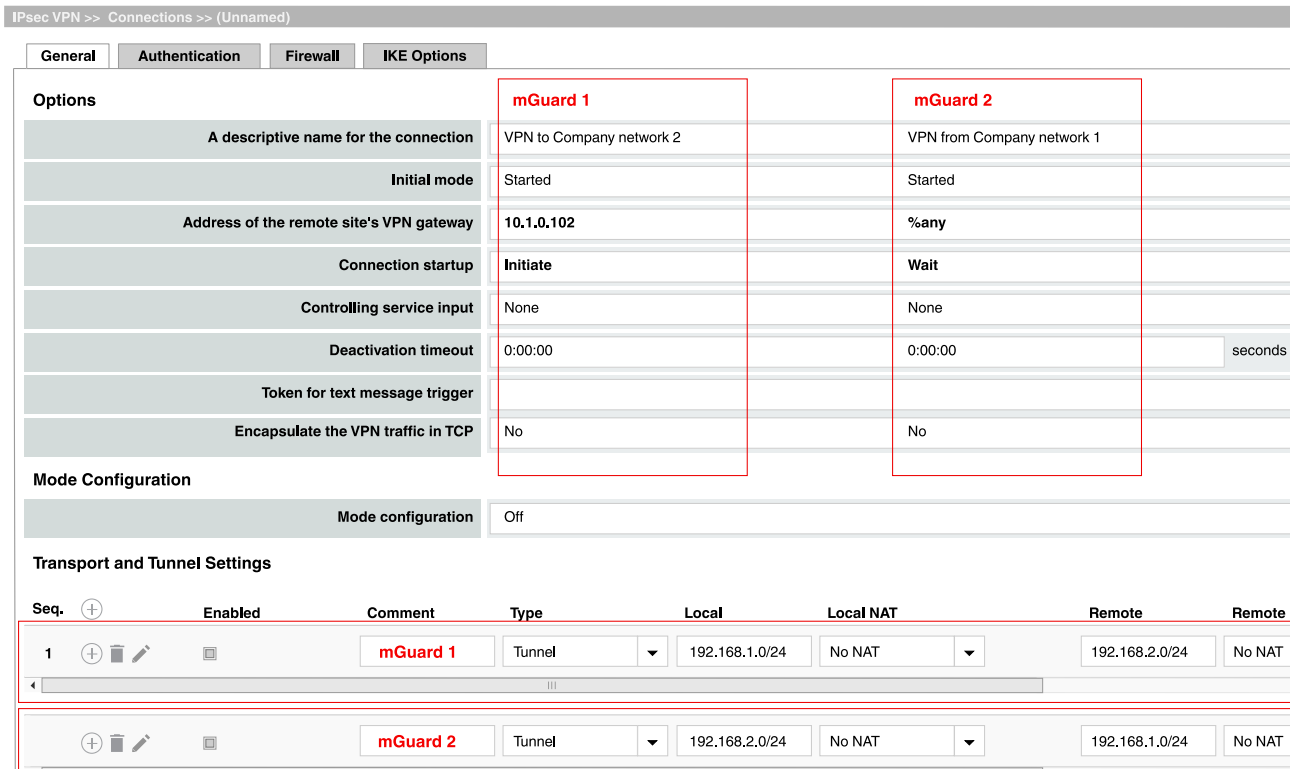


Figure 1-3 Menu: IPsec VPN >> Connections >> (Edit) >> General

1.3 "Authentication" tab

Mutual authentication of the two VPN peers can be performed in two ways:

– X.509 certificates
– Pre-shared key (PSK)

Pre-shared key (PSK)

This procedure is mainly supported by older IPsec implementations. In this case, both sides of the VPN connection authenticate each other using the same password (PSK). The PSK is made up of a string consisting of alphanumeric characters. The PSK procedure can be used in the secure *Main Mode* or in the unsecure *Aggressive Mode* (see also mGuard firmware user manual, section "IPsec VPN >> Connections >> Authentication").

X.509 certificates

This procedure is supported by most IPsec implementations. Here, each VPN device has a (secret) private key and a public key in the form of an X.509 certificate, which contains further information on its owner and a *Certification Authority (CA*) (see also mGuard firmware user manual, section "IPsec VPN >> Connections >> Authentication").

Which procedure is to be used?

Certificates are generally more secure and can be applied in all network scenarios. Creating a certificate, however, requires a certain amount of effort and precise planning.

Using a PSK in *Main Mode* with a sufficiently complex password is also relatively secure. In some network environments, however, PSKs cannot be used or can only be used with difficulty:

– PSKs in the secure *Main Mode* cannot be used if the VPN connection is established via one or more gateways with *Network Address Translation* (NAT) enabled. This means that PSKs can only be used if both devices are connected to the same external network or are connected directly to the Internet. Otherwise, the unsecure *Aggressive mode* would be necessary.
– When using PSKs, the external (or public) IP address of the peer VPN gateway must be entered in each location in the VPN configuration. The generic entry *%any* cannot be used on the responding side. For this, the unsecure *Aggressive mode* would be necessary.

1.3.1 Example: Creating X.509 certificates

A certificate acts as a unique ID and must therefore be unique for each device. The X.509 certificates can either be obtained from a commercial certification authority (e.g. *VeriSign*), or a Microsoft CA server, or can be created with software tools such as *OpenSSL* and *XCA* (see also application note "Creating X.509 certificates with OpenSSL/XCA".

When creating a certificate, the parameters that can be used to clearly determine the ownership of the certificate must first be specified (*Common Name, Organization, Organization Unit,* etc.).

Next, a key pair is generated: a private key and the corresponding public key. The private key *must* be carefully protected, while the public key can be published.

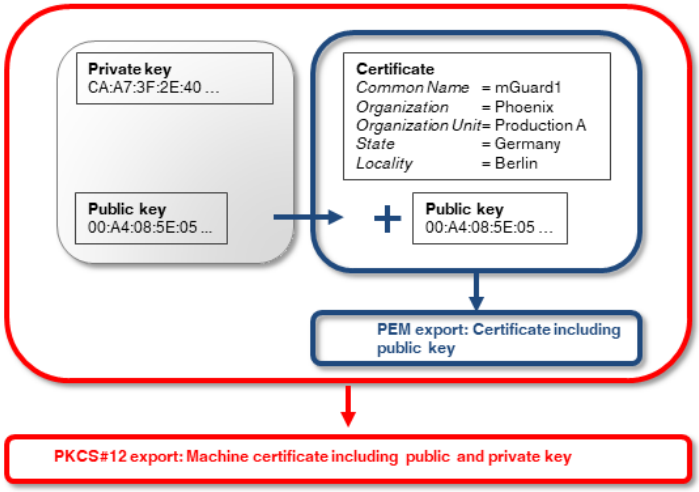


Figure 1-4 PEM and PKCS#12 exports of X.509 certificates with public or public and private keys

1.3.2 Example: Using X.509 certificates

In a VPN connection, the following must be determined

– How the mGuard device is authenticated by the peer and
– How the mGuard device authenticates the remote peer.

If authorization occurs via X.509 certificates, the VPN connection can only be established if the private key on one side “corresponds“ to the public key on the other side (see also Section 1.3, “Importing machine certificates (PKCS)”).

The certificates created must therefore be exported in two different formats, and imported into the respective devices:

1. **PEM format:**

   The certificate in PEM format only contains the public key. It must be imported into every device that attempts to establish a VPN connection with the device to which the certificate (PKCS#12 export = *Machine Certificate*) belongs (see Figure 1-5).

2. **PKCS#12 format:**

   The certificate in PKCS#12 format contains both the public and the associated (corresponding) private key. It will only be imported into a particular device as the unique *Machine Certificate* of this device (see Figure 1-5).

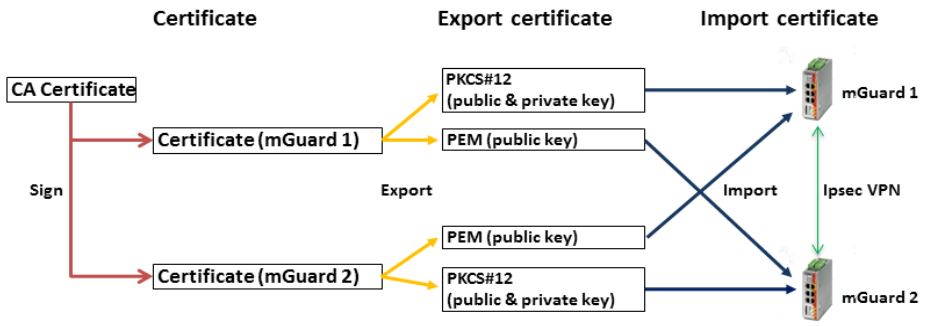


Figure 1-5 Necessary certificates in an IPsec VPN connection

Table 1-1 Example: Certificates in an IPsec VPN connection

Device	Machine certificate (also contains the private key)	Client certificate (only contains the public key)
mGuard 1	*mGuard1.p12*	*mGuard1.pem*
mGuard 2	*mGuard2.p12*	*mGuard2.pem*

mGuard devices also support CA authentication. With this function, the peer is authenticated via the CA certificate used to sign the peer certificate (remote certificate). Authentication via the remote certificate itself is then no longer necessary. This function is mainly used in VPN tunnel groups.

Multiple use of a certificate (as a device-specific ID) on different devices is not recommended and will normally lead to problems.

Uploading X.509 certificates to devices and using in VPN connections

The use of X.509 certificates in mGuard devices is described in Section 1, “VPN Kickstart – Connecting two networks together via IPsec VPN”.

1.4 "Firewall" tab

VPN-specific firewall rules can be specified when configuring the VPN connection. The VPN firewall allows access via the VPN tunnel to be restricted. It can be configured as necessary. In the default configuration, all incoming and outgoing connections are accepted.

(See also mGuard firmware user manual, section "IPsec VPN >> Connections >> Firewall").

1.4.1 Example

An encrypted IPsec VPN tunnel is to be established between **company network 1** (192.168.1.0/24) and **company network 2** (192.168.2.0/24).

Two clients in company network 1 are to have access to two controllers in company network 2. All other clients are to be denied access to company network 2. All connections from company network 2 to company network 1 are not permitted.

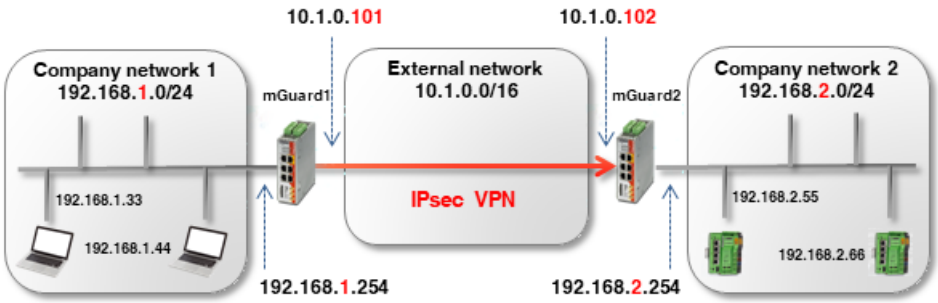


Figure 1-6 VPN connection between two networks with firewall

The firewall settings can, in principle, be configured on *mGuard 1* or *2* or on both devices. In this example, the firewall is configured on *mGuard 1*. The use of firewall rule records is also possible (see also Section 1).

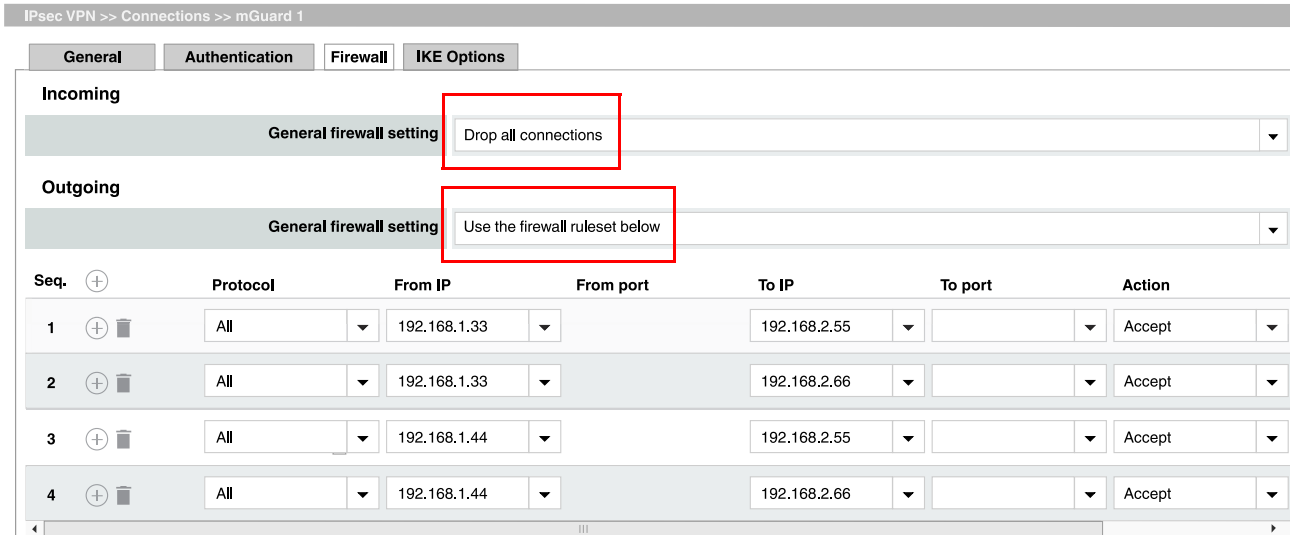


Figure 1-7 *mGuard 1*: IPsec VPN >> Connections >> (Edit) >> Firewall

1.5 "IKE Options" tab

Internet Key Exchange (IKE) is a protocol used for management and exchange of the keys involved within the IPsec protocol.

The IKE options specify

– The encryption and hash algorithms that are to be used for ISAKMP SA and IPsec SA
– The service life of the SAs and
– the parameters for Dead Peer Detection (DPD).

The strongest or most secure encryption method and/or hash algorithms are to be used wherever possible. Otherwise, the standard settings can in principle be used. (See also mGuard firmware user manual, section "IPsec VPN >> Connections >> IKE Options").

For information on secure encryption, see mGuard firmware user manual (Section "Secure encryption").

1.6 mGuard behind a NAT router

If the VPN connection is established via one or more gateways on which *Network Address Translation* (NAT) is enabled

1. X.509 certificates must be used for secure authentication. *Pre-Shared Keys* (PSK) can only be used in the unsecure *Aggressive Mode*.
2. only one of the mGuard devices can *initiate* the VPN connection. The other mGuard device must *wait* for the connection.
3. *%any* must be specified as *Address of the remote site's VPN gateway* on the responding mGuard, even if the NAT router of the peer has a static public IP address.
4. it must be ensured that the VPN connection is established via the UDP ports 500 and 4500.

The network and NAT settings shown in the following example are to be observed.

1.6.1 VPN initiator behind NAT router

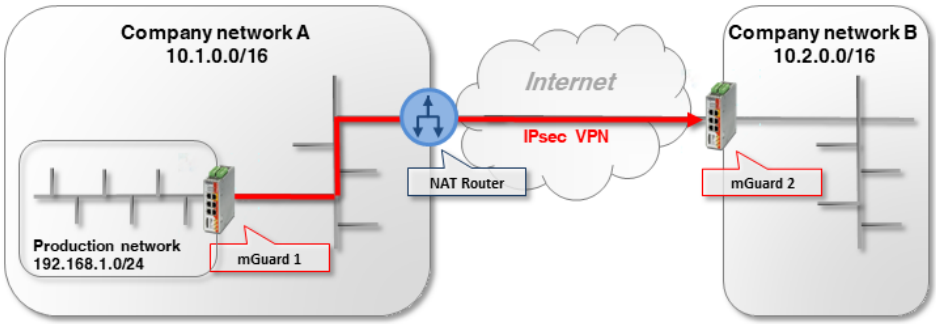


Figure 1-8 VPN initiator behind NAT router

mGuard 1 (*initiator*) initiates the VPN connection to *mGuard 2* (*responder*).

The NAT router firewall must allow outgoing UDP packets to the ports 500 and 4500. If these ports cannot be opened for any particular reason, *TCP encapsulation* or the *Path Finder* function can be used to establish the VPN connection (see Section 1.7).

1.6.2 VPN responder behind NAT router

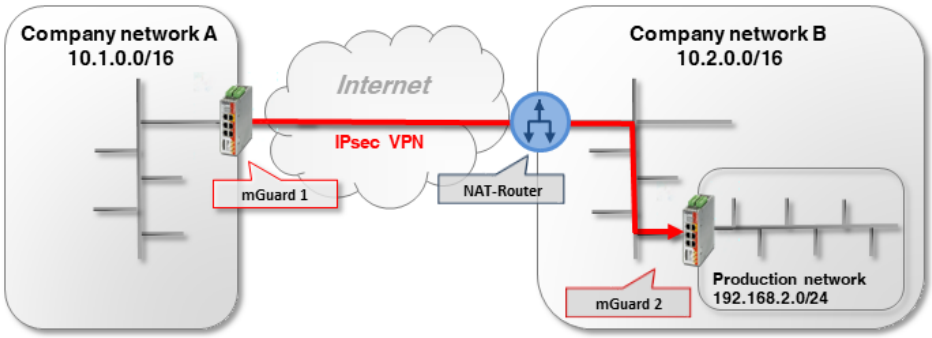


Figure 1-9 VPN responder behind NAT router

mGuard 1 (*initiator*) initiates the VPN connection to *mGuard 2* (*responder*).

Port forwarding to the external IP address (WAN port) of *mGuard 2* must be configured on the NAT router for the UDP ports 500 and 4500. (If it is an mGuard device, this can be set via **Network >> NAT >> IP and port forwarding**).

As port forwarding is necessary on the NAT router for the UDP ports 500 and 4500, no further VPN connections to the NAT router can be established (termination). (This would be possible using TCP encapsulation/Path Finder function.) Likewise, it will not be possible to establish VPN connections to further mGuard devices in the company network B.

If this is to be the case, *mGuard 2* would have to initiate the VPN connection to *mGuard 1*. It would then not be necessary to configure port forwarding on the NAT router.

1.6.3 VPN initiator and responder behind NAT routers

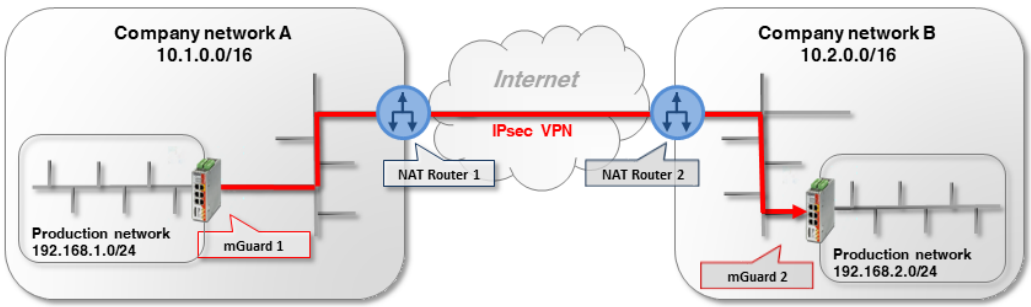


Figure 1-10 VPN initiator and VPN responder behind NAT routers

mGuard 1 (*initiator*) initiates the VPN connection to *mGuard 2* (*responder*).

The NAT router 1 firewall must allow outgoing UDP packets to the ports 500 and 4500.

Port forwarding for the UPD ports 500 and 4500 to the external IP address (WAN port) of *mGuard 2* must be configured on NAT router 2.

1.7 TCP encapsulation

In order to be able to establish an IPsec VPN connection, the UDP ports 500 and 4500 must be open in an outgoing firewall. If these ports are blocked, the VPN connection can be established using *TCP encapsulation* or the *Path Finder* function via a permitted TCP port.

Here, the UDP packets are packed in TCP packets (encapsulated) and sent to a TCP port which allows outgoing TCP packets (e.g. port 80 or 8080) according to the firewall settings of the NAT router.

TCP encapsulation can also be used for establishing the VPN connection if access to the Internet is only possible via a customer proxy server. In this case, the access parameters must be configured in the proxy server (menu **Network >> Proxy Settings**).

1.7.1 Example

A customer would like to access the server of a manufacturing company via a VPN connection. However, the customer firewall blocks the UDP ports 500 and 4500 for outgoing connections.

TCP connections via TCP port 80 are, on the other hand, permitted. The VPN connection is therefore to be established using TCP encapsulation via the TCP port 80. (The configuration of VPN connections is described in detail in Section 1 and 1).

Certificates must be used for secure authentication because the VPN connection is established via a NAT router. If authentication is to be via *pre-shared key*, the unsecure *Aggressive Mode* must be used (see Section 1.3).

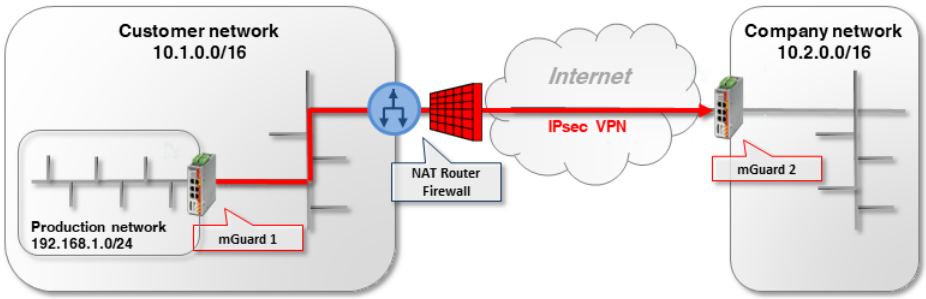
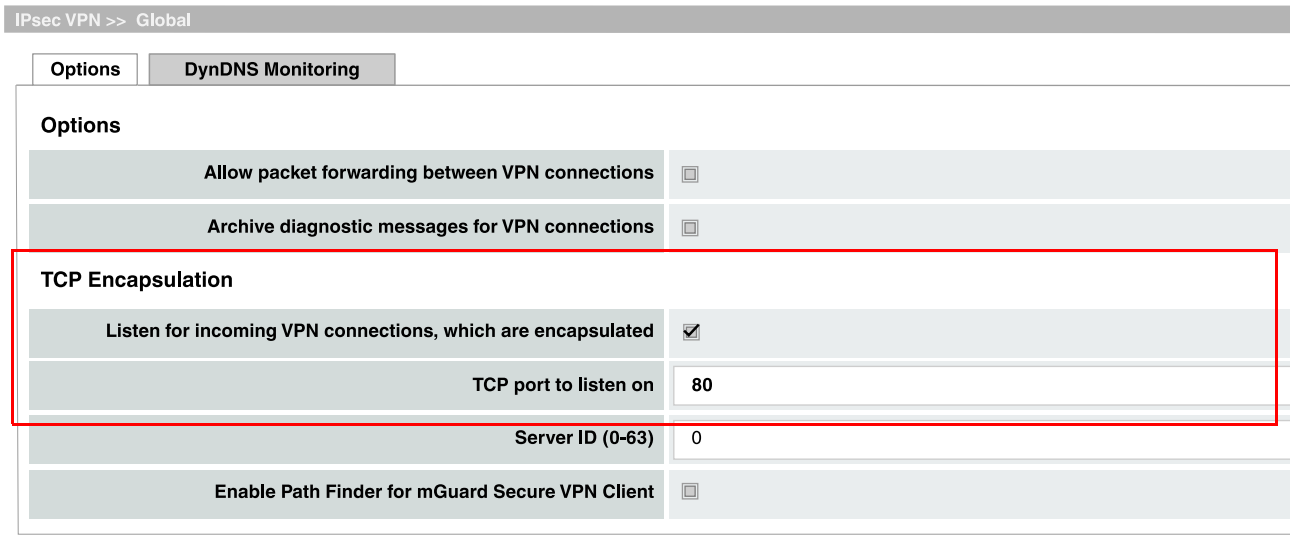


Figure 1-11 VPN initiator behind NAT router and firewall

mGuard 1 (*initiator*) initiates the VPN connection to *mGuard 2* (*responder*). Normally, a VPN connection is established using NAT via the UDP ports 500 and 4500. These, however, are blocked by the customer firewall of the NAT router.

The encrypted ESP packets are also encapsulated in UDP packets by NAT-T. They would also be affected if UDP ports 500 and 4500 were blocked.

1.7.2 mGuard 2 settings (responder)



To set which port the *VPN responder* should be listening on for encapsulated VPN connections, proceed as follows:

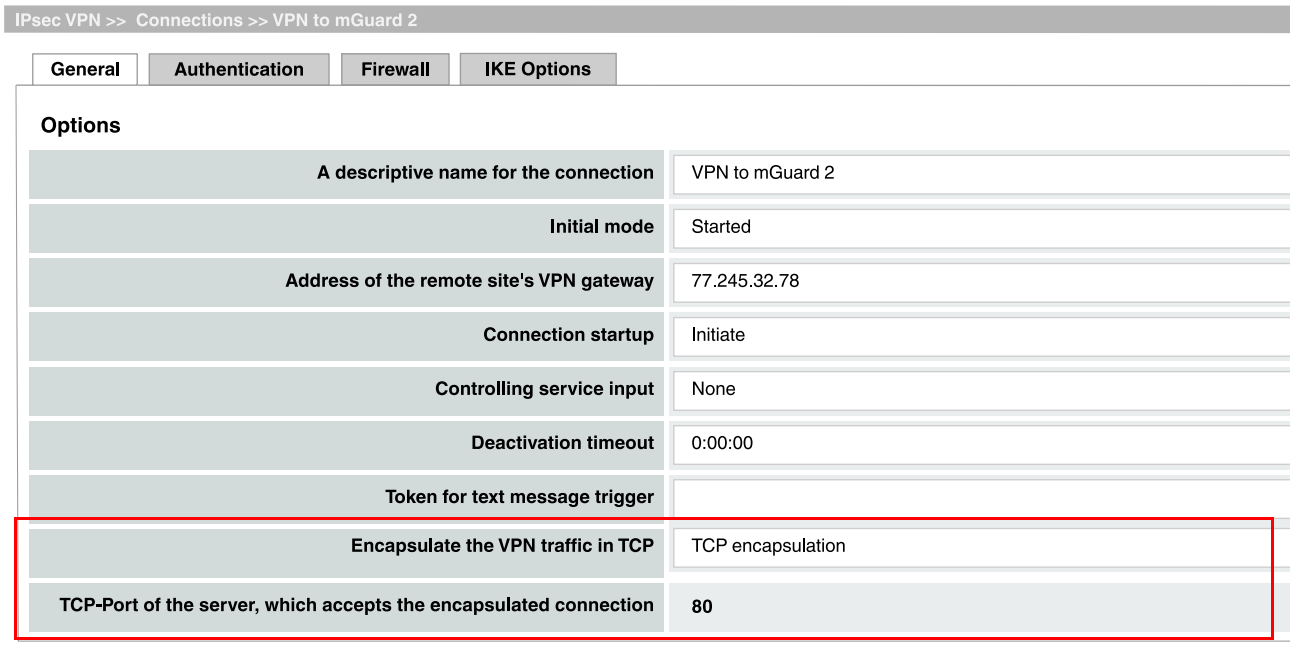
1. Log in to the *mGuard 2* web interface.
2. Go to **IPsec VPN >> Global** (*Options* tab).
3. In the section **TCP Encapsulation**: activate the option **Listen for incoming encapsulated VPN connections**. This will start the IPsec TCP server on the device.
4. In this example, enter port *80* in the field **TCP port to listen on**. This port must also be entered for TCP encapsulation in the *VPN initiator* (*mGuard 1*) (see Section 1.7.3).

Do not select TCP port 443, as it is already used by default to access the device's Web-based management via HTTPS remote access.

If TCP encapsulation also uses port 443, HTTPS remote access to the web interface is no longer possible.

Either specify a different TCP port for remote access (menu **Management >> Web Settings**, *Access* tab), e.g. Port 4443, or select another TCP port for TCP encapsulation.

1.7.3 mGuard 1 settings (initiator)



To inform the *VPN initiator* which port the peer device (*VPN responder*) listens for on encapsulated VPN connections, proceed as follows:

1. Log in to the *mGuard 1* web interface.
2. Go to **IPsec VPN >> Connections**.
3. Click on the ⊕ icon to add a new VPN connection.
4. Specify a unique name for the connection and click on the ✎ icon to edit the connection.
5. Enter either the DynDNS name or the public IP address of the peer (*mGuard 2*) (e.g. *mGuard2.dyndns.org* or 77.245.32.78) as the **Address of the remote site's VPN gateway**.
6. Select *Initiate* in the **Connection startup** field.
7. Select TCP encapsulation in the field **Encapsulate the VPN traffic in TCP**.
8. In this example, enter port *80* in the field **TCP port of the server, which accepts the encapsulated connection**. This port must also be entered for TCP encapsulation under *VPN responder* (*mGuard 2*) (see Section 1.7.2).

1.8 Starting/stopping or analyzing VPN connections using URLs

It is possible to start or stop or query the status of a VPN connection configured on the mGuard using the command line command *curl*:

https://<user>:<password>@<mGuard IP>/nph-vpn.cgi?name=<name>&cmd=[up|down|status]

<user>: the following users can be used: *admin*, *root* and *user*.

<name>: name of the VPN connection, as displayed in the menu **IPsec VPN >> Connections**.

Using the command line command **wget** only works in conjunction with **mGuard firmware versions <8.4.0**. The *curl* command line tool can be used with mGuard firmware version 8.4.0, or higher.

The user password and the name that an action relates to may only contain the following characters:

– Letters: A - Z, a - z
– Numbers: 0 - 9
– Characters: - . _ ~

Other characters, such as a space or question mark, must be encoded accordingly (see also mGuard firmware user manual).

1.8.1 Examples

The mGuard device on which, for example, the "*Athen*" VPN connection is configured can be reached via the IP address 192.168.1.1.

1. Starting the "*Athen*" VPN connection:

wget --no-check-certificate "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up"

curl --insecure "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=up“

2. Stopping the "*Athen*" VPN connection:

wget --no-check-certificate "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=down"

curl --insecure "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=down“

3. Requesting the status of the "*Athen*" VPN connection:

wget --no-check-certificate "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=status"

curl --insecure "https://admin:mGuard@192.168.1.1/nph-vpn.cgi?name=Athen&cmd=status“

The option ***--no-check-certificate*** (*wget*) or **--insecure** (*curl*) ensures that the HTTPS certificate of the mGuard device will not subsequently be checked.

1.9 Starting or stopping a VPN connection via button or switch

Service contacts (I/Os) can be connected to some mGuard devices:

TC MGUARD RS4000/RS2000 3G, TC MGUARD RS4000/RS2000 4G,
FL MGUARD RS4004/RS2005, FL MGUARD RS4000/RS2000, FL MGUARD RS,
FL MGUARD GT/GT

The user manual describes how the service contacts are connected to the devices (see mGuard Hardware Manual – UM EN MGUARD DEVICES).

Input (CMD I1, I2, and I3)

Buttons or on/off switches (e.g. a key switch) can be connected to the inputs. One or more freely selectable VPN connections or firewall rule records can be switched via the corresponding switch. A combination of VPN connections and firewall rule records is also possible.

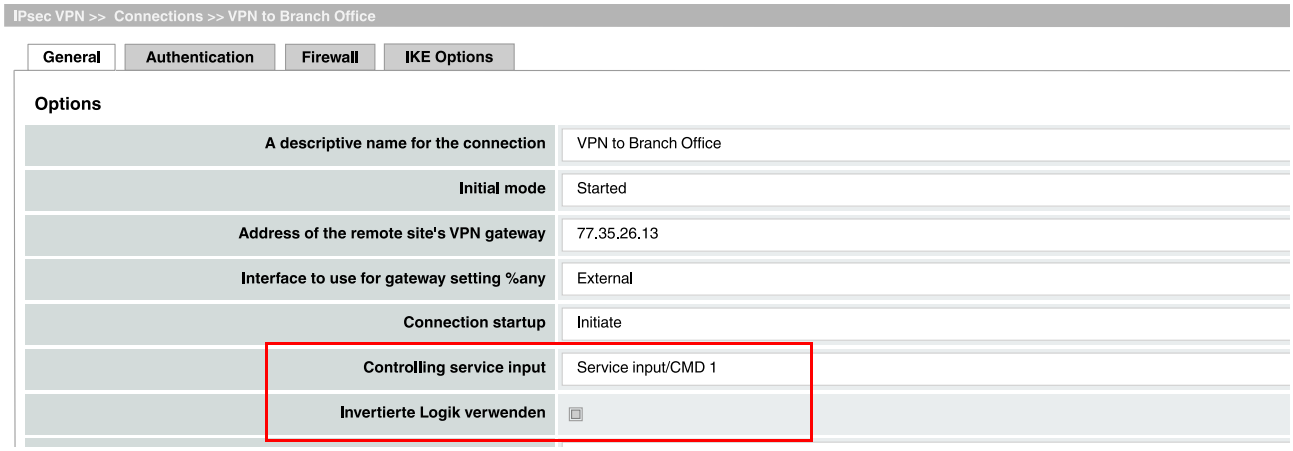


Figure 1-12 A service input is assigned to the VPN connection via which it can be started or stopped via button or on/off switch.

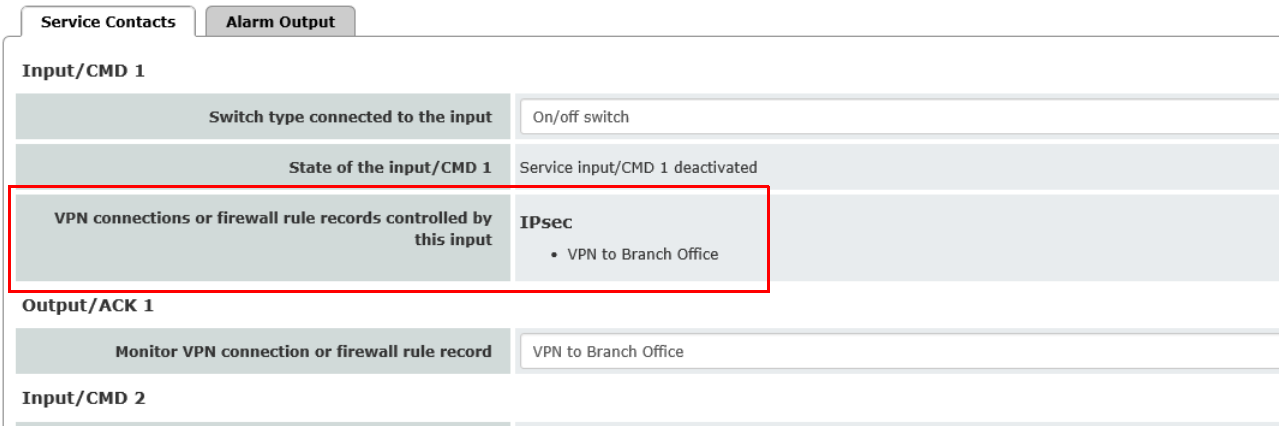


Figure 1-13 The web interface displays which VPN connections and which firewall rule records are controlled via a service input.

Signal contact (signal output) ACK 1/2 (O1, O2)

You can set whether specific VPN connections or firewall rule records are monitored and displayed via the signal output ACK 1, ACK 2, or LEDs.