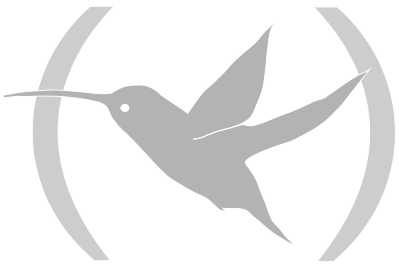
Teldat Router

IPSec

Doc. *DM739-I* Ver. *10.91*
April, 2013

INDEX

Chapter 1
Introduction

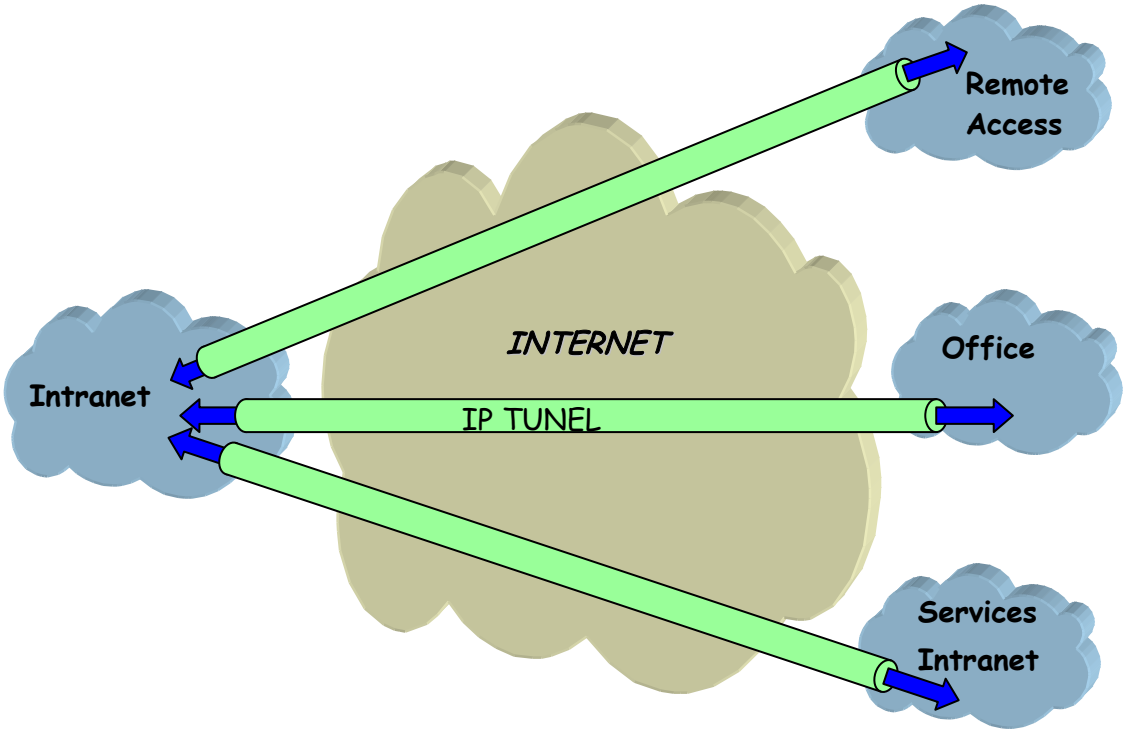


1. Virtual Private Networks

Until now, companies have traditionally used the Internet to promote their services and products through Web Sites. Today more and more companies use the Internet to communicate between their branches, offices or R+D centers. In short, the Internet could take the place of expensive private and less flexible telephone lines. Furthermore, the e-business requires global access (World Wide Web) offered by the Internet.

The packets which circle public networks, such as the Internet, are moved by multiple nodes that cannot be controlled or watched over. The route of these packets for the same destination is variable and therefore security mechanisms need to be established to prevent any intruder from accessing the information that you send through this type of network.

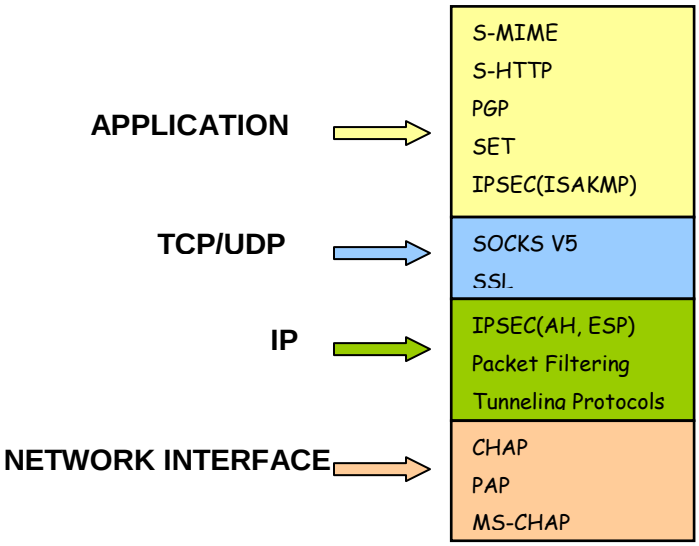
The purpose of a Virtual Private Network (VPN) is to extend a company's Intranet through a public network such as Internet: creating secure communications with Private Tunnels.



Different types of VPN solutions exist that can be classified depending on the OSI level of the protocol where these are implemented:

- The implemented VPNs in the *application* level: Authenticate and/or encrypt the message but not the source and destination address of the packets that these route.
- The VPNs based in the *link* level: Like L2TP, these can only authenticate the Tunnel's extreme end nodes but not each packet separately.
- The VPNs implemented in the *network* level: Like IPSec, this protects the data and IP source and destination address without the user having to modify the applications. However outside of the Tunnel, for example in the company's Intranet, no protection is provided.

In conclusion, it is best to combine application level VPNs with the network level VPNs to obtain an adequate security level.



2. IPSec

IPSec is a security platform at the *network* level developed by the *IETF IPSec Working Group*. This provides the ability to accommodate new encryption and authentication algorithms in a flexible and robust way.

IPSec focuses on the following security problems:

- Authentication of data sources: verifies that the received data has been sent by the person who says they have sent it.
- Data integrity: verifies that the received data has not been modified en route.

  The term data authentication is usually used to indicate both the integrity of the data as well as source authentication.
- Data Confidentiality: conceals the data using an encryption algorithm.
- Protection Anti-Replay: prevents an intruder from re-sending one of your messages and you are unable to detect it.
- Automatic cryptography keys management.

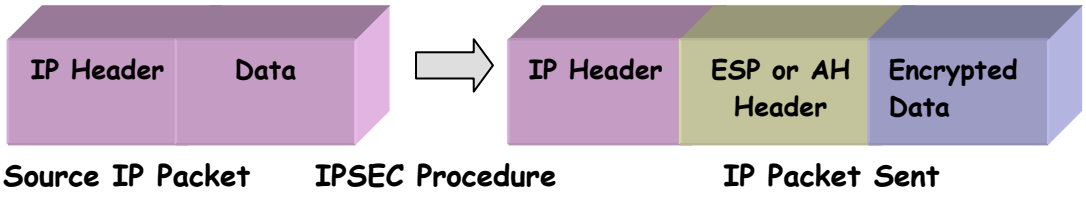
In order to resolve these aspects, IPSec defines two distinct security services:

- **ESP:** *Encapsulating Security Payload*: provides confidentiality, address source authentication in each IP packet, integrity and protection from copies being made.
- **AH:** *Authentication Header*: provides address source authentication in each IP packet, integrity and protection against copies being made, however this does not offer data confidentiality. This service is appropriate in cases where you only need to affirm the origin of the data.

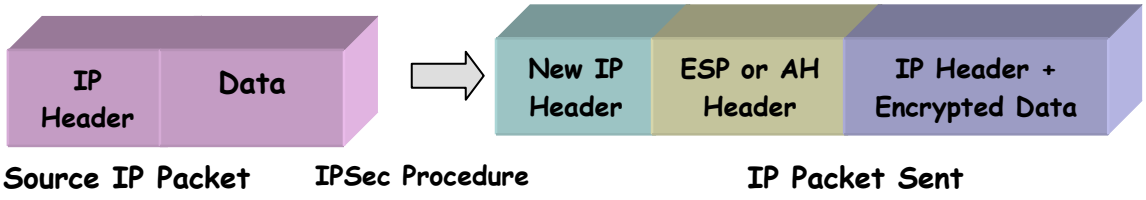
2.1. IPSec Tunnels

The IPSec platform permits two operation modes. You can use either of the two security services, ESP or AH, in each of them:

- The **Transport Mode** permits secure communications, normally established between the two hosts (e.g. communication between a workstation and a server or between two servers). However, in neither case does this mask the source or destination address of the packet to be sent. In transport mode, IPSec only acts over the IP packet internal data, without modifying the packet header. E.g. over a TCP or UDP segment or an ICMP packet.



- The IPSec **Tunnel Mode** encapsulates the whole of the original IP packet in a new IP packet, thus hiding all the original content. In this way the information is routed through a 'tunnel' from one point in the network to another without anyone being able to examine the content. This mode is the most appropriate one to be used in communications between a router and an external host or between two routers.



2.2. IPSec Architecture

a) Security Policy Database (SPD)

The IPSec platform must know which *security policies* to apply to the IP packet, depending on the header fields, also known as *selectors*. The security policies decide which encryption and authentication algorithms should be used in the secure connection.

The ***Security Policy Database*** (**SPD**) stores the entries that contain the selectors and the associated security policies.

After checking the security policies database, within the policies applicable to an IP packet, three possibilities exist:

- Discard the packet
- Route the packet normally.
- Apply the IPSec Security with some determined encryption or authentication algorithms that depend on the obligations of the security-efficiency adopted. For example, if you consider the processing speed as being more important than security, choose the DES encryption policy instead of the Triple DES.

b) Security Association (SA´s)

A packet whose selector coincides with one of the ***SPD*** entrances will be processed in accordance to the policy associated to this selector. A *Security Association* is the security connection that is created after the ***SPD*** has been consulted and contains the security information (authentication keys and encryption) required to process the packet.

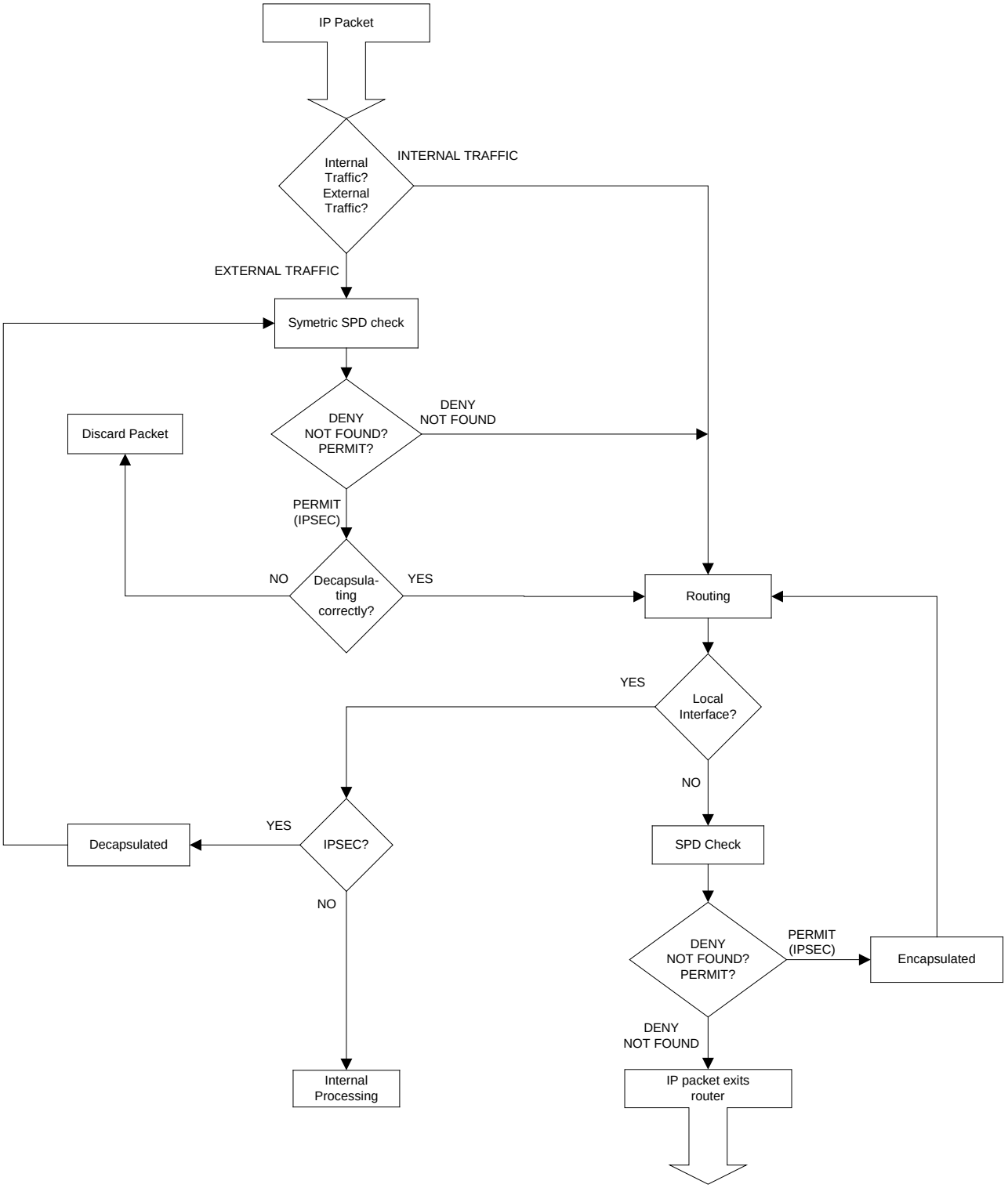
Within each of these security services (ESP or AH) we can choose different types of encryption algorithms, (DES, TRIPLE DES etc), or authentication (MD5, SHA1, etc.).

c) Packet processing with IPSEC-Router

There exists a single ***SPD*** or policies database that the user defines. This database is defined for the outgoing router traffic, while the incoming traffic is controlled through an ***implicit SPD***, symmetric to the previous one. In this way, all the incoming packets are processed in the same way the outgoing packets are sent: if certain outgoing traffic is defined to be sent with a specific security IPSec policy, this waits for the corresponding incoming traffic to comply with the same policy. In the same way, if the action defined for the outgoing traffic is route / discard, the incoming traffic will also be route / discard.

After carrying out the internal routing, the ***SPD*** is checked, this time for the outgoing traffic and similarly the decision must be taken between IPSec encapsulation, routing or elimination.

The following diagram describes the processing of an IP packet in **Teldat Router** with IPSec protocol:


IP Packet
Internal Traffic? External Traffic?
INTERNAL TRAFFIC
EXTERNAL TRAFFIC
Symetric SPD check
DENY NOT FOUND? PERMIT?
DENY NOT FOUND
Discard Packet
PERMIT (IPSEC)
NO
Decapsula-ting correctly?
YES
Routing
YES
Local Interface?
NO
YES
IPSEC?
Decapsulated
SPD Check
NO
DENY NOT FOUND? PERMIT?
PERMIT (IPSEC)
Encapsulated
DENY NOT FOUND
Internal Processing
IP packet exits router

2.3. Advanced IPSec

a) Keys management

The entire security platform based on secret keys stops being secure if the keys are not periodically renewed.

The shorter the refresh time, the greater security of our system against Cryptanalysis tools.

There are two possible general work modes for the management of the security parameters and passwords in IPSec: manual (IPSec manual) and automatic or dynamic (IPSec IKE). These modes refer to the way in which an agreement is reached between peers on security parameters established for the Tunnel.

b) Manual IPSec

In the IPSEC manual, "manual-keying", the keys used in the encryption and/or authentication process for each SA are introduced by the user. The user should introduce the same security parameters (keys, encryption and authentication algorithms) for both ends of the Tunnel so that secure communication can be carried out. This is practical for small relatively static environments. When your VPN begins to grow, the manual renewal of the keys can be a costly task.

c) IKE IPSec

The IPSec platform permits this process to be automated, thanks to the *IKE Internet Key Exchange* protocol (based on the OAKLEY key exchange protocol and the ISAKMP platform). The two ends of the Tunnel automatically negotiate the secure communication parameters (keys, encryption and authentication algorithms). In order to generate this negotiation, the ends must first carry out a **first phase** where they agree on the security parameters that will protect the negotiation. Additionally in this first phase, authentication of the Tunnel ends is carried out, using a common key (*Pre-Shared Key*) manually introduced at both ends, digital signatures or with a public key algorithm.

There are two pre negotiation modes: *Main Mode* and *Aggressive Mode*.

- *Main Mode* masks the identities of the Tunnel's end routers. This type of negotiation is required when both ends know the security server's IP addresses that they confront.
- *Aggressive Mode* does not mask these identities and improves the authentication processing rate. Additionally, it is unnecessary to know the IP address at the other end of the Tunnel. This permits you to establish a Tunnel with an unknown security router provided that the security policy applicable to the packet permits this.

IPSec IKE has four operation modes for the first phase, depending on the type of Authentication used to negotiate the SAs security parameters.

• *Authentication with Pre-shared Key*

The same key (Pre-shared Key) is manually introduced in the two SECURITY ROUTERs permitting mutual authentication.

Two types of exchanges exist with the Pre-shared Key: *Main Mode* and *Aggressive Mode*.

- The Main Mode masks the identities of the Tunnel end Routers.
- The Aggressive Mode does not mask these identities and improves the authentication processing speed.

Every time the life span of a SA times out, new key material will be exchanged between the two security routers prior to authentication with the manual Pre-shared key.

Conversely, IPSEC "manual-keying" and IPSEC with Pre-shared Key means you need to know the IP address of the Tunnel end (Security Router IP address with which you are operating).

However the following types of IPSec IKEs permit, automatically and dynamically, to establish a Tunnel with an unknown Security Router if the security policy applied to the packet permits this. In these types of IPSec IKE, you do not need to introduce a common key at the Tunnel ends as this is automatically obtained through the below described processes.

- *Authentication with Signatures*

The authentication of the two Tunnel ends is carried out through a digital signature and the key exchange system "Diffie Hellman".

Two types of exchanges exist: *Main Mode* and *Aggressive Mode*.

- The *Main Mode* masks the identities of the Tunnel end Routers.
- The *Aggressive Mode* does not mask the identities and improves the authentication processing speed.

- *Authentication with Public Key Encryption*

Authentication is carried out by RSA with previous knowledge of the public key of the other router. The public keys of the other end of the Tunnel can be obtained through *certificates.*

Two types of exchanges also exist: *Main Mode* and *Aggressive Mode*. If the public key is frequently updated, the Aggressive Mode is just as secure as the Main Mode and is faster.

In addition the Authentication with Public Key Encryption provides greater security with respect to the Signature Authentication and Authentication with a Pre-shared Key, by combining the RSA public key system and the "Diffie Hellman" key exchange system. However the processing time of the Authentication with Public Key Encryption is greater.

- *Authentication with a Revised Public Key Encryption*

Authentication is also carried out by RSA with previous knowledge of the public key of the other ROUTER. The public keys of the other end of the Tunnel can be obtained through *certificates.*

However, operations are reduced with public key with an insignificant loss of security, but improving the authentication services.

Two types of exchanges exist: Main Mode and Aggressive Mode. If the public key is frequently updated, the Aggressive Mode is just as secure as the Main Mode and is faster.

d) High Security

The keys used to encrypt or authenticate a communication are obtained from *Material for Keys*. If this material has not originated nor will originate other keys to encrypt or authenticate other communications, then we say that ***Perfect Forward Secrecy*** has been attained.

The **Teldat Router** in high security mode permits you to achieve Perfect Forward Secrecy at the cost of a higher computation rate when establishing the IPSec Tunnels.

The high security mode also generates more secure keys material using the OAKELEY Groups, which are more resistant to Cryptanalysis.

e) Certificates

The certificates permit you to know the public keys of other security Routers through which it is possible to establish an IPSec Tunnel. These public keys will be used in the two IKE authentication modes with public key.

f) TED (Tunnel End-Point Discovery)

The TED protocol is an addition to IPSec, and permits you to dynamically determine the end router used to open an IPSec tunnel with the Host router in order to guarantee communication confidentiality between the hosts which both routers protect.

In order to have an extensive network totally interconnected, you need to define static security parameters for all the possible pairs in the network. By using TED and a single set of dynamic security parameters, you can find the pair you are looking for without having to previously define them. It is also possible to add new links to the network without having to modify the configuration of each router residing in the said network.

When using the TED protocol, you need to bear in mind that the IP addresses of the hosts protected by the routers must be routable. Additionally these addresses are sent in clear and therefore the use of this protocol in scenarios where this information is considered confidential should be avoided. You also need to ensure that the associated access list only contains entries referring to IP (i.e. this cannot be used with UDP, TCP or any other protocol).

The protected IP addresses must be routable.

g) RRI (Reverse Route Injection)

RRI is an algorithm which permits the router on the other side of an IPSec tunnel to insert static routes in the networks protecting this tunnel in their corresponding routing tables. These routes are inserted when an IPSec tunnel is up and they indicate how to reach the network (with mask) protected by the access list associated to the tunnel, with a next hop defined by configuration (this next hop can be the local end of the tunnel, the remote end or an IP address defined by the user).

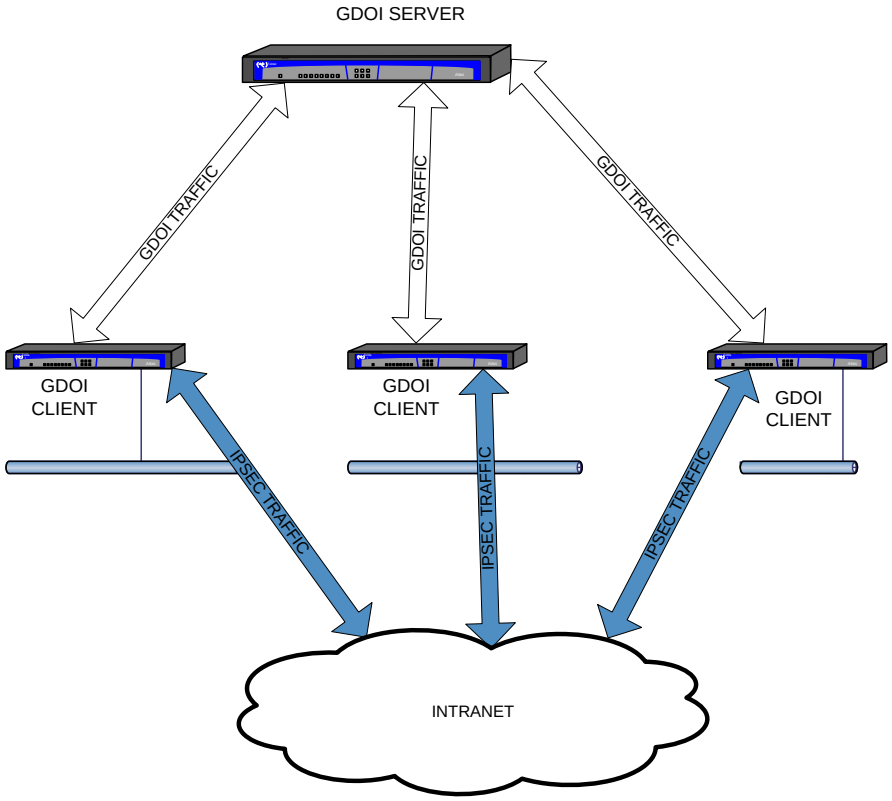
The ultimate aim of this functionality is to broadcast these routes backwards through a routing algorithm (RIP or OSPF for example), thus permitting the devices located behind the router to find out the path needed to send the encrypted traffic to the network or networks protected by the tunnel.

h) GDOI (Group Domain Of Interpretation)

Definition:

GDOI (Group Domain Of Interpretation) is defined as the Internet Security Association Key Management Protocol (ISAKMP) Domain of Interpretation (DOI) to manage group keys. The GDOI protocol operates between a client or a group member (GM) and a central server or key server (GCKS). This central server establishes security associations (SAs) between the authorized clients. The ISAKMP described in the RFC 2048 defines two negotiation phases: the GDOI protocol is protected by the ISAKMP phase 1, phase 2 changes completely and is defined in the RFC 3547.

In the schema, you can see two types of traffic that intervenes, GDOI between the clients and the server, and IPSEC traffic between clients using the policies downloaded through the GDOI protocol.



Operative:

A client registers in the server to obtain the SAs needed to communicate with the rest of the clients in the group. The client, during negotiations, sends a group ID to the server so the latter can send the policies and keys for this particular group. These keys are periodically refreshed through rekey messages before the current keys expire so traffic is not lost. The server handles the maintenance and updates the keys and the IPSec policies.

There are two types of key that the server can send: encryption keys for rekey messages known as KEK and keys to encrypt traffic known as TEK. The TEK keys are used to encrypt the IPSec packets exchanged between clients while the KEK keys are used to encrypt rekey messages exchanged between the server and the client.

The server sends the rekey messages when the TEK keys or the KEK keys need updating, and also send a rekey message if the server configuration has changed. Retransmission can be configured for these rekey messages a certain number of times so loss of the said rekey packets is avoided. Rekey messages can be send through unicast IP packets addressed to each client registered in the server or through a packet addressed to a configurable multicast IP.

IPSec encapsulation:

Packets encapsulated by GDOI clients are encapsulated in transport mode; consequently the IP destination and source addresses are not changed. This means that the IP routes for the devices in scenarios where GDOI is configured are not modified.

Client access list:

The GDOI client receives the access list from the server, which indicates what traffic is going to be encrypted and what traffic is clear. The entries on this received access list must match at least one of the entries on the access list configured in the GDOI client. Each entry on the received access list is installed in front of the entry it matches (this doesn't look at the permit/deny field).

There are two basic configurations:

- Configure a "permit" all entry:

```
access-list x
   entry 1 default
   entry 1 permit
exit
```

In this case,

- Before connecting to the server:
  - If non-encrypted traffic is received, it is dropped.
  - No traffic is sent until the server is connected.
- After connecting to the server:
  - The traffic is encrypted or not, depending on the downloaded access list entries which are installed in front of "entry 1".

- Configuring a "deny" all entry:

```
access-list x
   entry 1 default
   entry 1 deny
exit
```

In this case,

- Before connecting to the server:
  - Non-encrypted traffic received is admitted.
  - The traffic is sent in clear.
- After connecting to the server:
  - The traffic is encrypted or not, depending on the downloaded access list entries which are installed in front of "entry 1".

In addition, you can add exceptions to that indicated by the server, adding entries to the access list associated to the GDOI template.

E.g. if you associate this list to a GDOI template in a client, this forces the traffic between hosts 172.24.1.1 and 172.24.1.2 to be sent in clear (unless the server has explicitly indicated that this specific traffic must be sent encrypted):

```
access-list x
   entry 1 default
   entry 1 deny
   entry 1 source address 172.24.1.1 255.255.255.255
   entry 1 destination address 172.24.1.2 255.255.255.255
;
   entry 2 default
   entry 2 permit
;
exit
```

In the example below, if this list is associated to a GDOI template in a client, this forces traffic between hosts 172.24.1.1 and 172.24.1.2 to be sent encrypted (unless the server has explicitly indicated that this specific traffic must be sent in clear):

```
access-list x
   entry 1 default
   entry 1 permit
   entry 1 source address 172.24.1.1 255.255.255.255
   entry 1 destination address 172.24.1.2 255.255.255.255
```

```
;
        entry 2 default
        entry 2 deny
;
      exit
```

Anti-replay based on a timestamp:

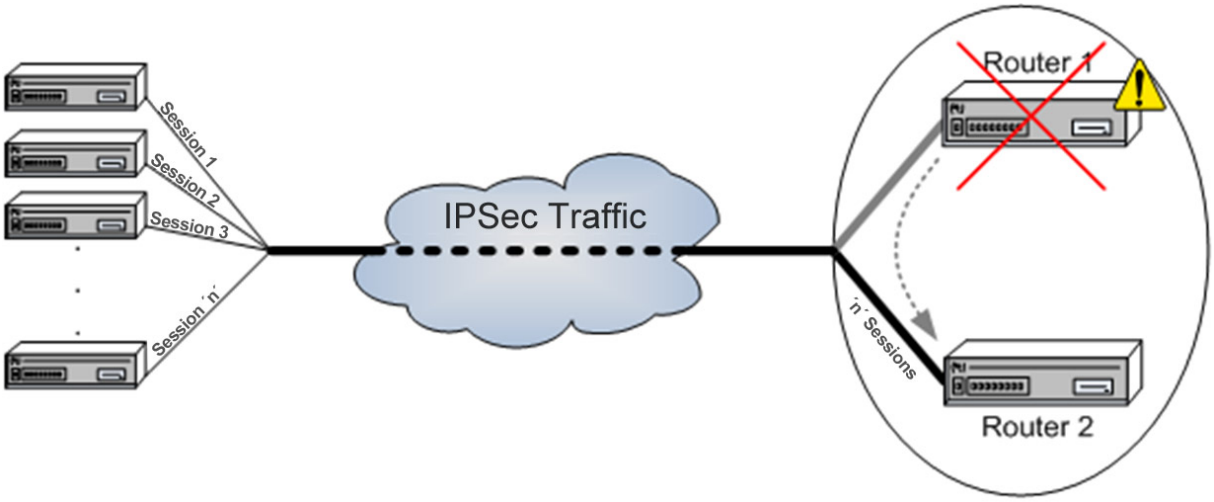
Anti-replay is an important feature in the IPSec protocol. This prevents third parties from listening to packets from an IPSec conversation and subsequently replaying them later as if they had been generated at that point. Anti-replay, based on a timestamp, ensures that illegally reproduced packets are detected and dropped. The Teldat implementation of GDOI uses a synchronous anti-play (SAR) independent of the real time and date of the devices. A global timestamp is sent by the server (GCKS) synchronizing the clients to the said timestamp, the clients updating it while the seconds lapse. When a client sends a packet, a timestamp is introduced in the IP packet in proprietor format, which is compared in the client receiving the packet with the current timestamp. If the received timestamp and the current timestamp differ more than the configurable value given, the packet is dropped.

i) Fault Tolerant IPSec Recovery

Fault Tolerant IPSec Recovery is a feature that permits Teldat devices to continue managing IPSec packets even in cases where one of the tunnel terminator devices is inoperative.

- *First steps*

Fault Tolerant IPSec Recovery is based on the dynamic distribution of the IPSec sessions between a pair of Teldat routers, i.e. the sessions can be moved from one device to another depending on the current conditions and on the configuration. In this way, the IPSec sessions taken on by a device that stops working can be automatically and transparently established in the device that is operative.



Fault Tolerant IPSec Recovery is supported in VRRP and IPSecFT as well as in IPSec:

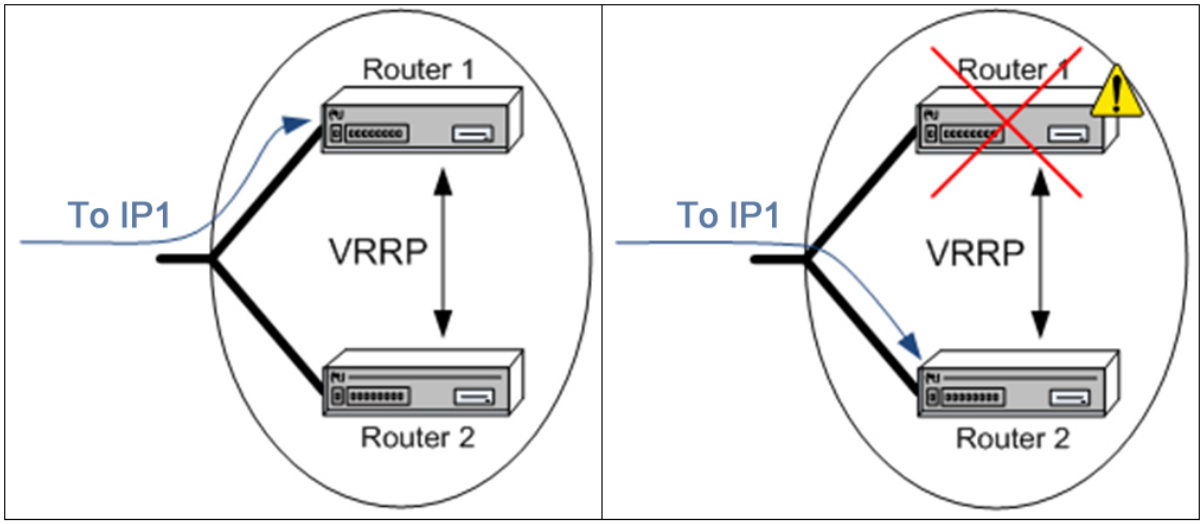
- VRRP (Virtual Router Redundancy Protocol) is defined in the RFC 3768 and dynamically assigns the virtual router function to one of the VRRP routers. This protocol decides which device should route the packets addressed to the IP address shared by VRRP and therefore, serves as a base to decide which router should have established what IPSec sessions at any point.
- IPSecFT (IP Security Fault Tolerant) is the protocol that exchanges information between the two devices that act as tunnel terminators. IPSecFT permits each of the two devices to maintain an updated database with sufficient information in order to inherit the IPSec sessions that its partner has established at any point,

Both protocols co-exist so that IPSec has the sessions corresponding to it established at any time.

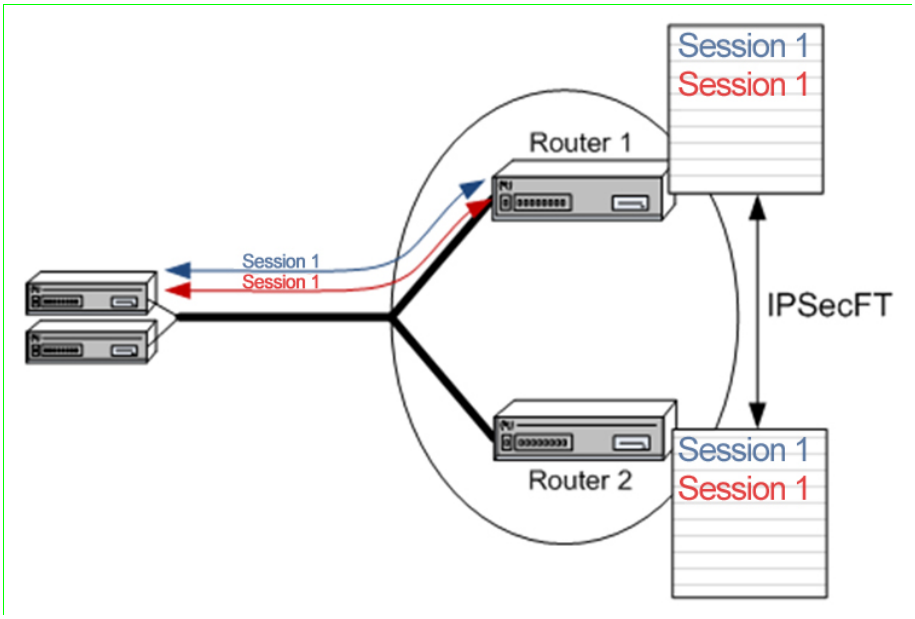
- *Operation*

Entering into further detail on the subsystem operation, this is based on externally presenting two devices as if they were one. This is achieved by sharing some virtual IP addresses between them. In this way the external devices beginning the IPSec session do them with the said shared addresses without worrying about which device is currently managing.

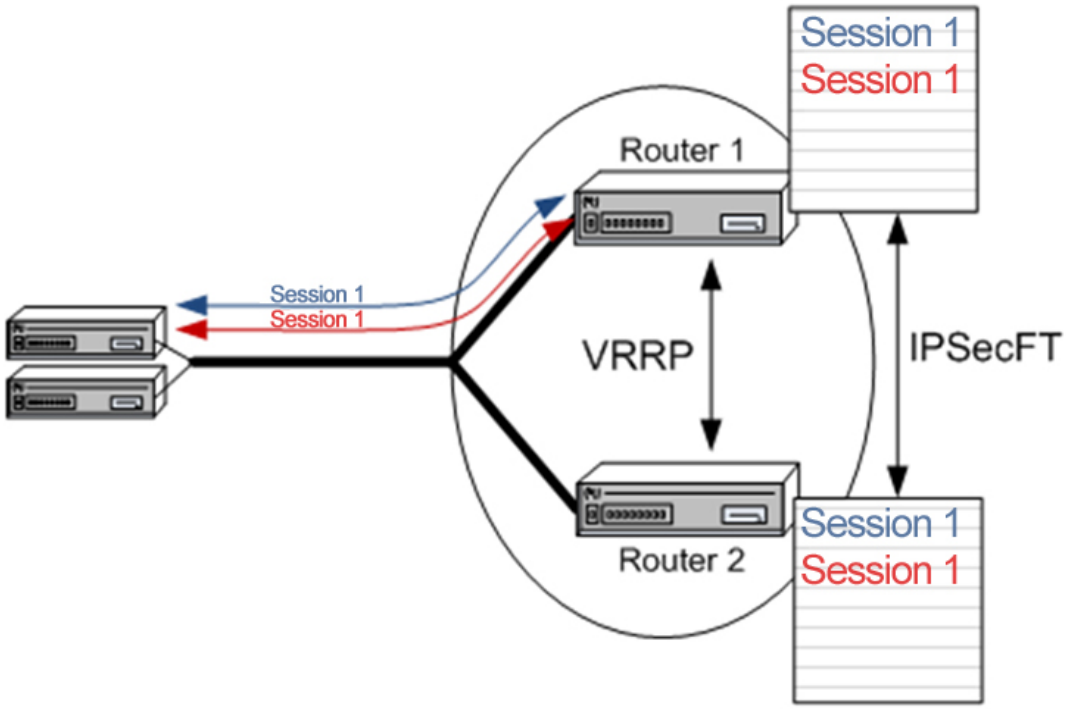
The conditions that one or the other device has the virtual IP address associated at each moment are delegated in VRRP. For further information on this please see manual "Dm759-I VRRP Protocol"



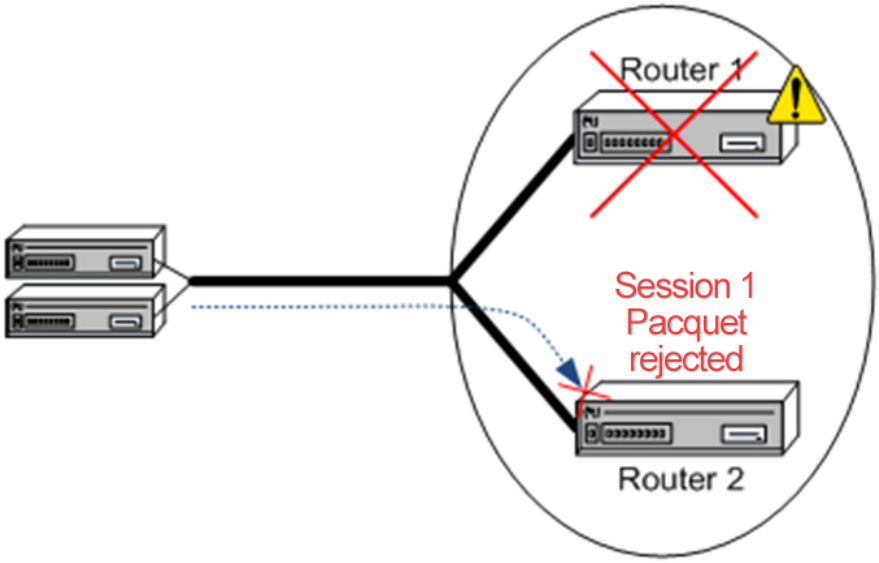
An IPSec session needs an establishment and exchange of keys so the device understands and accepts sent encrypted packets i.e. it's not as simple as resolving the management problem for the IP addresses used to connect the IPSec sessions, we also need to resolve the continuity of the said IPSec session and the IPSecFT protocol is used for this. As already said, this protocol maintains a database with the information on the sessions that the analogue has established, i.e. it would be able to establish these if necessary,



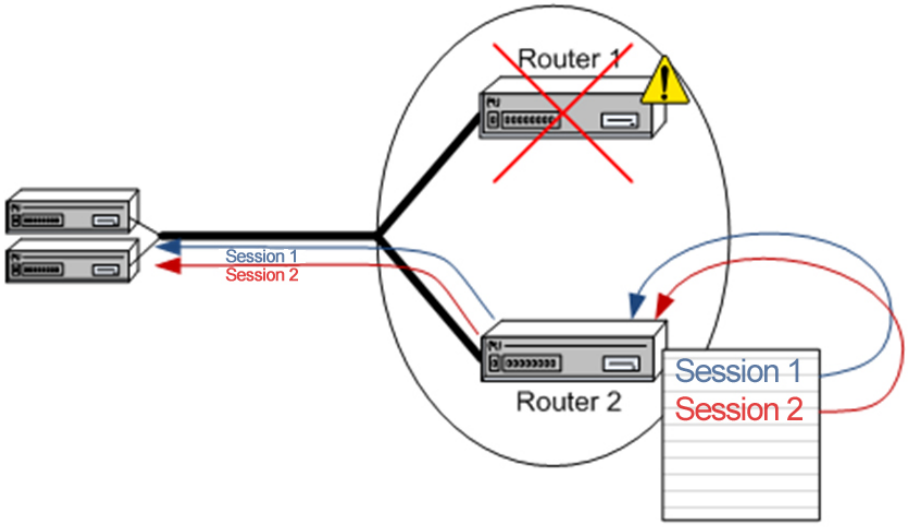
In a stable situation, the sessions are established with the device that manages the destination IP address for the said sessions, in this way the IPSec encrypted data is received and decrypted without difficulty.



Imagine a situation where the device with the established sessions fails for whatever reason and is incapable of managing the said IPSec sessions, e.g. it switches off. At this point the shared virtual IP address will be managed by the router that is still active and, therefore will begin to receive IPSec packets that it doesn't understand (the IPSec session hasn't been established with it.)



However, the active device doesn't expect to receive IPSec packets that it doesn't understand and immediately acts as it is now managing the shared virtual IP address. What it does at this point is to go to the IPSecFT database for the established sessions and on seeing that it is going to receive traffic from sessions it doesn't have established, it goes ahead and establishes them.



Consequently, the IPSec sessions established in one device that has failed, pass to another device that can take them, providing continuity in sending data and automatically and transparently resolving the problem.

- *Important operating considerations*

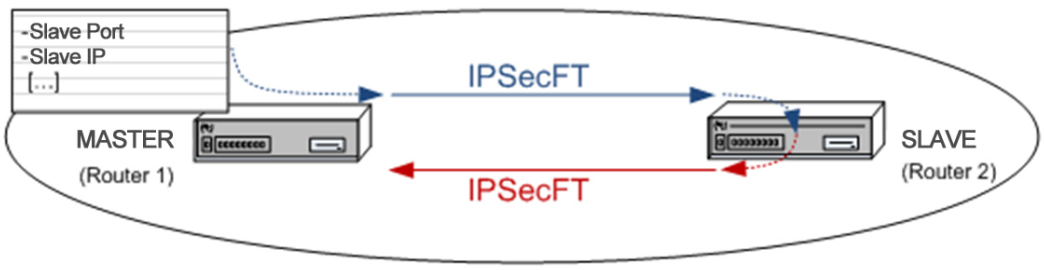
There are certain parameters when adjusting Fault Tolerant IPSec Recovery that need to be mentioned. As already said, this function depends on the IPSec, IPSecFT and VRRP protocols.

IPSec

IPSec configuration controls the process to establish the IPSec sessions, i.e. configure the keys, the type of encryption and the rest of the parameters for the IPSec conventional configuration. Given that both devices are going to establish the same sessions indistinctly, they must have the same IPSec configuration.

IPSecFT

IPSecFT establishes and maintains two TCP sessions through which information relative to the IPSec sessions that each device has established is exchanged. This uses a TCP session in each direction, from Router1 to Router2 and from Router2 to Router1. However for end user convenience, one device is declared as master and the other as slave, and only the master device is configured with which slave device it should connect to: the slave device that receives the connection automatically establishes the second in the opposite direction.



For the protocol function, it's doesn't matter which device is configured as master and which as slave, nor does it under any circumstances interfere with anything relative to the master and slave in VRRP.

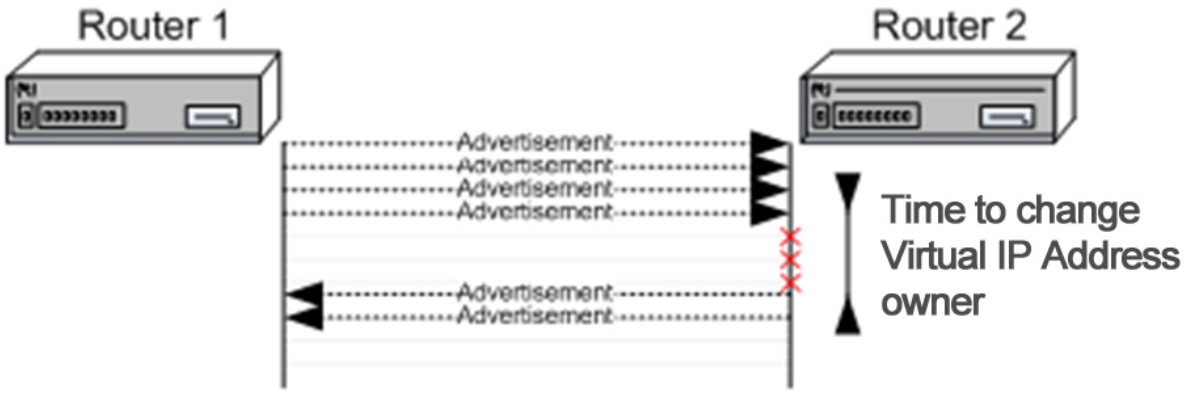
Once the TCP sessions are established they progress depending on the time configured, i.e. IPSecFT on time out sees to the tasks it has pending. The following two are the most important:

- Sending of packets to monitor the IPSecFT session, keepalive packets. If a preset time times out without having received the said packets, then the session is considered invalid and one minute later is released.
- Shared virtual IP address polling. If the case arises where the address we have unchains the establishment of the IPSec session.

In cases where the IPSecFT session is lost, this tries to reestablish it every second.

VRRP

VRRP decides which device manages the shared virtual IP address depending on some multicast packets that the managing device periodically sends, advertisement packets. When the listening device detects too much time has passed without receiving the said packet, then it begins to manage the virtual IP address. Once the device inherits the said shared address, it begins to establish the IPSec sessions needed to provide continuity on sending data. I.e. the less time that the device waits before inheriting the shared IP address, the less time the system will take in recovering when the device with IPSec sessions established fails; however, a too low a time can lead to an active device being considered as down as it cannot serve the advertisement packets as quickly as necessary.

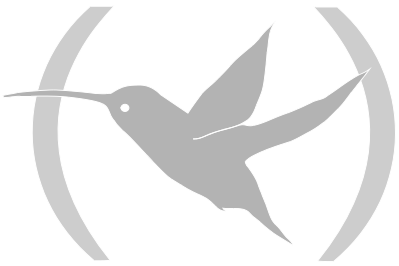


General considerations

When the IPSec sessions are transported from one device to another, the one receiving them is subject to a high work load. You need to be very careful in selecting an appropriate value in the number of input buffers in the interface where the sessions are established as otherwise not all the packets coming from the remote devices can be processed.

Likewise, you need to increase the IPSec encryption queue size to a value of at least the number of simultaneous sessions that are going to be established or up to the number of input buffers in the interface where the sessions are established, the greater of the two.

Chapter 2
Configuration



1. Introduction

As seen in the chapter 1 section 2.2 “IPSec Architecture”, the processing of an IP packet by the IPSEC module, is based on applying the security policies configured for the said packet. This information is stored in the ***Security Policy Database (SPD)***, where the selectors and the associated security policies are found. In this way, the IPSEC configuration in the device is reduced to the definition of the ***SPD*** elements.

In the **Teldat Router**, the configuration of an SPD element is carried out in three steps. Firstly an element or an Access Control List (LCA) entry is defined i.e. some determined control selectors, which assigns a previously configured generic access list to IPSec. A type of decision is configured for each entry in the list: permit a packet to pass without applying the corresponding process to the protocol or feature which was assigned to this list (Deny) or apply the corresponding process in this IPSec case (Permit). If none of the entries in the list is applicable, the packet will not be processed by IPSec. Subsequently the **Templates** or IPSec security policies are created where the IPSec Tunnel security parameters are defined. Finally an access control list assigned to IPSec is associated (mapped) with a specific **Template**.

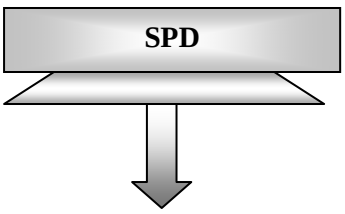
Access 1 control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

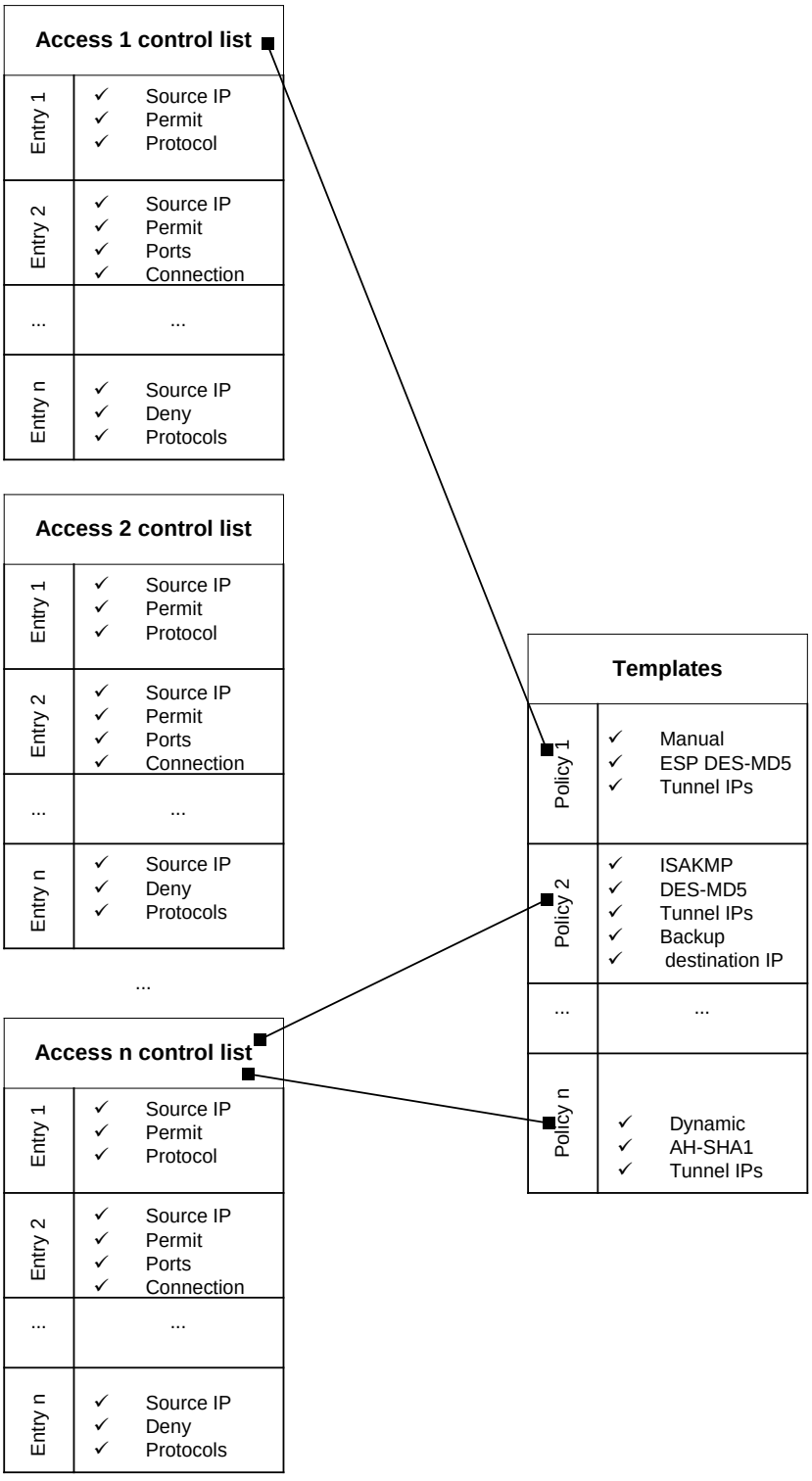
Access 2 control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

...

Access n control list	
Entry 1	✓ Source IP ✓ Permit ✓ Protocol
Entry 2	✓ Source IP ✓ Permit ✓ Ports ✓ Connection
...	...
Entry n	✓ Source IP ✓ Deny ✓ Protocols

Templates	
Policy 1	✓ Manual ✓ ESP DES-MD5 ✓ Tunnel IPs
Policy 2	✓ ISAKMP ✓ DES-MD5 ✓ Tunnel IPs ✓ Backup ✓ destination IP
...	...
Policy n	✓ Dynamic ✓ AH-SHA1 ✓ Tunnel IPs




Access 1 control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
Access 2 control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
...
Access n control list
Entry 1
✓ Source IP
✓ Permit
✓ Protocol
Entry 2
✓ Source IP
✓ Permit
✓ Ports
✓ Connection
...
...
Entry n
✓ Source IP
✓ Deny
✓ Protocols
Templates
Policy 1
✓ Manual
✓ ESP DES-MD5
✓ Tunnel IPs
Policy 2
✓ ISAKMP
✓ DES-MD5
✓ Tunnel IPs
✓ Backup
✓ destination IP
...
...
Policy n
✓ Dynamic
✓ AH-SHA1
✓ Tunnel IPs

2. First Steps

2.1. Initial configurations

Given that the access to the device permits modifying the IPSec parameters, you first need to configure the access passwords for Telnet and the device Console.

In cases of using certificates, you need to adequately configure the date and time of the device in order to prevent validation problems with these.

DISABLE / ENABLE Commands

The **DISABLE** command, found in IPSec configuration menu, permits you to disable the IPSec.

```
Config>PROTOCOL IP

-- Internet protocol user configuration --
IP config>IPSEC

-- IPSec user configuration --
IPSec config>DISABLE
IPSec config>
```

Simply write the **ENABLE** command to enable it.

In Nucleox Plus devices, you also need to enable the encryption card interruptions. The access password for this configuration, if this is not changed, is **teldat**.

```
Config>UCI CHANGE CFG

User Password? ******


Configuration

Interruption mode  (y/other)? (YES) y
Test RSA when starting (y/other)? (NO)
Max NRIs (10-500)? (100)
Flag Crypto? (NO)
You must restart so that the new configuration becomes effective
    Updating encrypt configuration...
```

3. IPSec Configuration

3.1. Commands for correct configuration

Once the device is connected to the private and public network, the ***SPD*** must be configured for incoming and outgoing packets.

The recommended steps to execute to generate a configuration are:

a) Configure the IPSec Access Control List.
b) Configure the Templates (security parameters).
c) Create the SPD.

3.2. Configuration

This section describes the steps to be followed in order to configure the IPSec in the **Teldat Router**. To access the IPSec configuration protocol environment, you must introduce the following commands:

```
Config>PROTOCOL IP

-- Internet protocol user configuration --
IP config>IPSEC

-- IPSec user configuration --
IPSec config>
```

Within the IPSec configuration protocol environment (indicated by the **IPSec config**> prompt) the following commands, among others, are available.

Command	Operation
? (HELP)	Lists the available commands or options.
ENABLE	Permits you to enable the IPSec and filter the events to be viewed.
DISABLE	Disable the IPSec.
ASSIGN-ACCESS-LIST	Assigns an access control list to the IPSec protocol.
TEMPLATE	Command to configure security policies parameters for the IPSec Tunnels.
MAP-TEMPLATE	Command that associates (mapping) an element in the access control list with a Template.
ASSOCIATE-KEY	Associates a key to an access control list.
ASSOCIATE-DEST-MASK	Associates a destination mask to an access control list.
KEY	This is used and described in the section on Dynamic Templates (IPSec IKE).
EVENT	Permits you to configure a filter to limit the events to be viewed or to display all of them.
QOS-PRE-CLASSIFY	Enables pre-filtering of packets (for BRS).
ADVANCED	Configuration of Advanced parameters.
LIST	Lists the IPSec configuration.
NO	Deletes elements from the Templates and Access Control lists, undoes mappings or deletes the whole of the configuration.

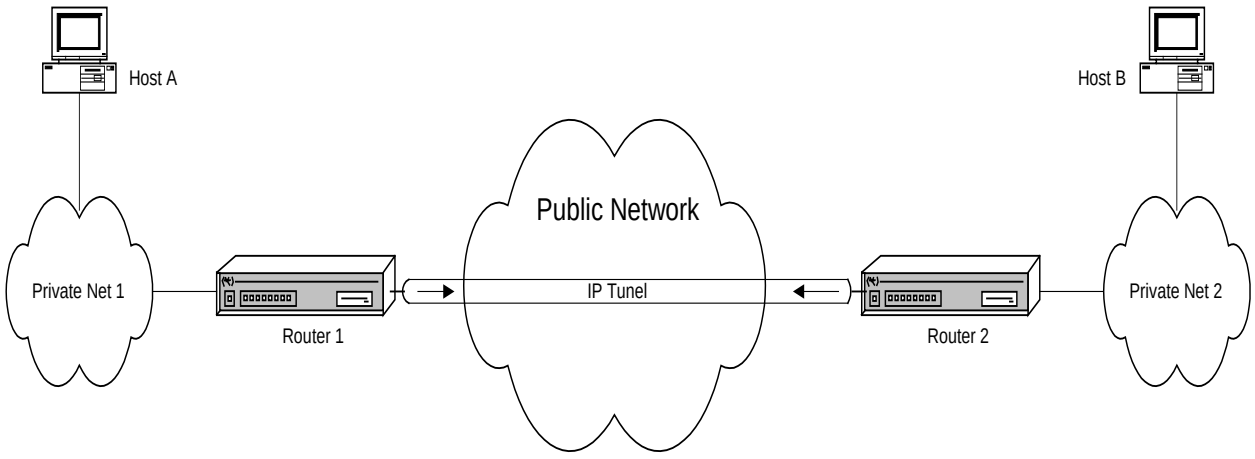
EXIT Exits the IPSec configuration prompt.

In general, if you do not introduce all of the parameters required in the line commands to complete the command, the device will then request the information, except where there is an option to write subcommands. In either case, you can always enter the command or subcommand followed by '**?**' in order to get help.

```
IPSec config>?
ENABLE                  Enables IPSec
DISABLE                 Disables IPSec
ASSIGN-ACCESS-LIST      Assigns access lists to IPSec (used as SPD selectors)
TEMPLATE                Configures security policies params for IPSec tunnels
MAP-TEMPLATE            Associates an element in the LCA with a template
ASSOCIATE-KEY           Associates a key to an access list
ASSOCIATE-DEST-MASK     Associates a destination mask with an access list
KEY                     Adds preshared or RSA keys
EVENT                   Adds a filter for IPSec events or enables all of them
QOS-PRE-CLASSIFY        Enables QOS Preclasiffy
ADVANCED                Configuration of advanced IPSec parameters
LIST                    Lists the IPSec configuration
NO                      Disables options, deletes items or sets default values
EXIT                    Exits IPSec configuration menu
IPSec config>
```

a) IPSec access control list configuration

As already mentioned, there exists an access control list. Each entry in this list is a block of selectors and an *action*, which is defined by a unique number (the entry identifier or ID field). The block of selectors is made up of the source IP address (or range of addresses), an IP destination address (or range of IP destination addresses), a protocol (or range of protocols), source and destination ports (or a range of ports), and the identifier of the connection between interfaces through which the packet is transported. It is not necessary to specify all of these, only those you wish. The *action* represents the procedure assigned to the packets coinciding with the associated block of selectors: PERMIT or DENY.



As already explained on analyzing the ***SPD***, the specification of the LCA entries or elements are always established for **outgoing packets** through the router interfaces. As an example, in the previous figure we wished to establish an IPSec secure Tunnel for packets being routed between host A and host B. For this the control entry to be established in the LCA will contain the following selectors (as a minimum):

- Host A source IP address;
- Host B destination IP address;

- Action: PERMIT (IPSec processing);

Any packet that travels from A to B in this way is encapsulated by IPSec. Implicitly on defining this entry, any packet arriving from B with address A must arrive with the same encapsulation. In this way the secure Tunnel between both ends is completely defined.

The order in the Access Control List is important in cases where the information offered the selectors overlaps between different LAC elements.

However, this order does not give the identifier ID for each entry, just the order in which they are listed (this can be modified). I.e. if on searching through the list, beginning with the first element or entry that appears, you find an element that fits with your search, the search will not continue and the action indicated in the said element will be applied.

IPSec makes use of the generic and extended access control lists defined in the root menu of the device configuration ***Config*> FEATURE ACCESS-LISTS.** The lists created in this menu must be assigned to the IPSec protocol through the ***IPSec config***>**ASSIGN-ACCESS-LIST** command**.** The order in which these lists are assigned determines the query order applied to the processed packets.

A generic and extended access control list is made up of a series of *entries* which define the properties that a packet must have in order to consider that it pertains to this entry and consequently to this list. Subsequently, this generic access control list is assigned to a protocol.

The first step consists in creating the access control list through the **ACCESS-LIST #** command. E.g., **ACCESS-LIST 100** accesses the ***Extended Access List 100*>** menu. Here you can register entries through the command **ENTRY # subcommand**.

Subsequently, the access control lists are made up of entries that admit the following subcommands:

Command	Operation
PERMIT	Type of action (IPSec processing in cases where the list is assigned to this protocol).
DENY	Type of action: does not carry out any process.
SOURCE ADDRESS	Defines the List entry source IP address selector.
SOURCE PORT-RANGE	Defines the entry source port selector.
DESTINATION ADDRESS	Defines the entry destination IP address selector.
DESTINATION PORT-RANGE	Defines the entry destination port selector.
PROTOCOL-RANGE	Defines the entry protocol selector.
DSCP	Diff Serv codepoint.
CONNECTION	Selector identifier for the connection between interfaces.

And the special commands:

Command	Function
LIST	To list the entries.
MOVE-ENTRY	To change the order of the entries.
NO	To delete an entry.

As an example we are going to display all the formats of all the subcommands together with an example of each in a possible configuration.

"ENTRY [ID] PERMIT"

Identifies the entry as a permitted type. In cases of IPSec this indicates that IPSec must be carried out. Therefore the entry in the access control list with this action specifies who the *Tunnel clients* will be i.e. defines the traffic to be transmitted through the Tunnel. The ID field is the integer which identifies the entry or element in the access control list.

Example:

```
Extended Access List 100>ENTRY 10 permit
```

"ENTRY [ID] DENY"

Identifies the entry as a non-permitted type. In cases of IPSec, this indicates that IPSec should not be carried out.

Example:

```
Extended Access List 100>ENTRY 10 deny
```

"ENTRY [ID] SOURCE ADDRESS [IP ADD] [MASK]"

To establish the IP source address selector for a possible packet. The range of addresses chosen is indicated in the form of a subnet mask. Once more, the ID field is the integer that identifies the element or entry in the access control list.

This address may be unnumbered i.e. you can set an address associated to an interface which is unknown at the time of configuring the device as, for example, it will be assigned by another mechanism such as PPP.

Example 1:

```
Extended Access List 100>ENTRY 10 source address 192.168.4.5 255.255.255.255
```

Example 2:

```
Extended Access List 100>ENTRY 10 source address 192.168.4.0 255.255.255.0
```

In Example 1, there is only one IP source address, and in Example 2 the source address for the entire subnet is 192.168.4.0 with a 255.255.255.0 mask. Please note that on using the same ID (10), the new information is added to or substitutes that already existing for this element. In this way the final entry is modified as shown in the following example.

As already said, you can choose not to introduce all the parameters for a command or subcommand or request help ('**?**'), and the router itself will progressively request these. In the following example, you can see how this works in the case of introducing the same data as that displayed in the previous example (Example 2):

```
Extended Access List 100>ENTRY 10 source address
Source IP address [0.0.0.0]? 192.168.4.0
Source IP mask [0.0.0.0]? 255.255.255.0
```

"ENTRY [ID] SOURCE PORT-RANGE [LOW] [HIGH]"

Establishes the selector for the Source Port. You can also select a range using the LOW and HIGH fields as port identifiers or a single port by setting both to the same value.

Example:

```
Extended Access List 100>ENTRY 10 source port-range 21 25
```

"ENTRY [ID] DESTINATION ADDRESS [IP ADD] [MASK]"

This command is similar to the one which establishes the source IP address selector of a possible packet. However this one is used to establish the selector for the destination IP address.

Example:

```
Extended Access List 100>ENTRY 10 destination address 192.168.10.0 255.255.255.0
```

"ENTRY [ID] DESTINATION PORT-RANGE [LOW] [HIGH]"

Establishes the selector for the Destination Port. In the same way, you can select a range by using the LOW and HIGH fields as port identifiers or a single port by setting both to the same value.

Example:

```
Extended Access List 100>ENTRY 10 destination port-range 1000 2000
```

If, once entered, you wish to eliminate the destination port control (or the source port control), as originally found, simply introduce the complete range. In this case:

```
Extended Access List 100>ENTRY 10 destination port-range 0 65535
```

On specifying the complete range, by default the corresponding selector does not appear.

"ENTRY [ID] PROTOCOL-RANGE [LOW] [HIGH]"

To establish the selector for the protocol or the protocol range of the packet. The LOW field is the protocol identifier in the lowest limit of the range. The HIGH field is the identifier in the highest limit. In cases where you do not want a range, simply set both to the same value.

Example:

```
Extended Access List 100>ENTRY 10 protocol-range 1 9
```

"ENTRY [ID] CONNECTION [ID CONN]"

Permits you to establish the identifier of the connection between interfaces for an LCA entry. This connection identifies the logical interface through which the packet is routed; this is configured in the IP rules. On establishing this relation, IPSec can associate traffic not only by the packet source, destination address etc., but also by the specific connection interface. The ID field is the integer that identifies the entry or element in the access control list.

Example:

Supposing that the following rule defined in IP exists:

ID	Local Address --> Remote Address	Timeout	Firewall	NAPT
1	**172.24.70.1 --> 172.24.70.2**	**0**	**NO**	**NO**

This identifies a specific connection between a router's local address and an end (the rest of the parameters are not considered). We therefore define an entry in the LCA, with the identifier of this connection (**1**) as selector:

```
Extended Access List 100>ENTRY 10 connection 1
```

Leaving the connection without specifying it or setting it to zero means that the connection will not be considered on checking the LCA.

A question mark will appear beside the connection (e.g. **Conn:1?**) should this not exist, together with a warning message.

Through this, all the selectors for an element in the access list are configured. If you do not configure one of these, this will not be taken into account when checking the packet against the control list.

Therefore what is left to define is the action to execute over a packet that coincides with this selection and also modification, if priority for this entry over the rest in the list is considered necessary. In order to do this, use the following subcommands:

"MOVE-ENTRY [ID_TO_MOVE][ID_BEFORE]"

Modifies the priority of an entry, placing the "ID_TO_MOVE" element in front of the "ID_BEFORE" element in the access control list, thus giving priority to the "ID_TO_MOVE" element versus "ID_BEFORE".

Example:

In order to display this, we will assume that we have to introduce a second entry:

```
Extended Access List 100, assigned to IPSec

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

The aim of this second entry is to permit certain transparent traffic to pass between two network hosts, 192.168.4.0/24 and 192.168.10.0/24, however the previous entry makes this ineffective. In order to avoid this situation, the entry order must be modified:

```
Extended Access List 100>MOVE 11 10
```

The order of the list and priority is now:

```
Extended Access List 100, assigned to IPSec

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

If you send a packet between hosts' 192.168.4.8 and 192.168.10.27 (with the adequate protocol etc), this will coincide with LCA entry with identifier 11, the first on the list, therefore the packet can transparently pass. Regarding traffic between the rest of the network hosts, 192.168.4.0/24 and 192.168.10.0/24, on checking the list, coincidence with the first entry will not be found. Consequently this will pass to the second entry (identifier 10). In cases where the packet coincides with the protocol, source port etc., this will be processed via IPSec Tunnel.

"LIST ALL-ENTRIES"

Displays all the access control list elements.

Example:

```
Extended Access List 100>LIST ALL-ENTRIES

Extended Access List 100, assigned to IPSec

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

You can achieve the same result if you execute the "LIST ACCESS-LISTS ALL-ENTRIES" command found in the IPSec config> menu.

```
IPSec config>LIST ACCESS-LISTS ALL-ENTRIES

Extended Access List 100, assigned to IPSec

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

"LIST ADDRESS-FILTER-ENTRIES [IP ADD] [MASK]"

Displays the access control list elements with source or destination IP address that is included within the [IP ADD] and the [MASK] defined range.

Example:

```
Extended Access List 100>LIST ADDRESS-FILTER-ENTRIES 192.168.4.8 255.255.255.255

Extended Access List 100, assigned to IPSec

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

You can achieve the same result if you execute the "LIST ACCESS-LISTS ADDRESS-FILTER-ENTRIES" command found in the IPSec config> menu.

```
IPSec config>LIST ACCESS-LISTS ADDRESS-FILTER-ENTRIES 192.168.4.8 255.255.255.255

Extended Access List 100, assigned to IPSec

11    DENY    SRC=192.168.4.8/32  DES=192.168.10.27/32  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

"LIST ENTRY [ID]"

Displays the access control list identifier [ID] entry.

Example:

```
Extended Access List 100>LIST ENTRY 10

Extended Access List 100, assigned to IPSec

10    PERMIT  SRC=192.168.4.0/24  DES=192.168.10.0/24  Conn:0
        PROT=1-9  SPORT=21-25  DPORT=1000-2000
```

"NO ENTRY [ID]"

Command used to delete an identifier entry [ID] from the access list.

Example:

```
Extended Access List 100>NO ENTRY 10
```

b) Configuring the Templates (security parameters)

The Templates are IPSec security policies that can be associated to one or various elements in the Access Control List. Only the generic lists that have been previously assigned to IPSec can be associated to a Template.

In each Template, the addresses of the two ends of the Tunnel you wish to establish are defined (corresponding to the security routers); IPSec Tunnels key management authentication or encryption algorithms and the **manual** (manual IPSec) or **dynamic mode** (IKE IPSec) as well as a Template identifier (ID) number.

Each **mode** has a series of commands associated, some are common to both and others being specific to each, although when you are listing these in the Template the meanings of the configured mode will be shown.

First of all the **manual IPSec** configuration will be described and subsequently **IKE IPSec** will be displayed.

- *Manual Templates*

In the IPSec manual, "manual-keying", the keys used in encryption processing and/or authentication for each SA, are introduced by the user. The user must introduce the same security parameters (keys, encoded algorithms and authentication) for both ends of the Tunnel in order to carry out secure communication.

The following subcommands are available within the TEMPLATE command in order to configure the manual Templates:

Command	Operation
DEFAULT	Sets the default values for a Template.
MANUAL	Creates a static Template with a security service (ESP or AH).
SOURCE-ADDRESS	Introduces the address of the Tunnel source end in the Template.
DESTINATION-ADDRESS	Introduces the address of the Tunnel destination end in the Template.
SPI	Introduces the security configuration identifier number (SA) defined by the Template.
KEY	Introduces a DES key into Template.
TKEY	Introduces a Triple DES key into Template.
MD5KEY	Introduces a MD5 key into Template.
SHA1KEY	Introduces a SHA1 key in the Template.
DF-BIT	Indicates the process that must be given to the DF bit in the IPSec packets.
MTU-THRESHOLD	Indicates the minimum MTU threshold to use in the PMTU procedure.
MTU-DEFAULT	Indicates the initial value given to the MTU through the IPSec tunnel.

The first thing to define in a Template (manual or dynamic) is the security service you wish to use, ESP or AH. The ESP service (Encapsulating Security Payload) is a confidential service that encrypts data with an option to authenticate these. The AH service (Authentication Header) only permits authentication:

"TEMPLATE [ID] DEFAULT]"

Sets the default values for a Template.

Example:

```
IPSec config>TEMPLATE 4 default
```

"TEMPLATE [ID] MANUAL ESP [ENCRYPT] [AUTHEN]"

This command defines a Manual Template with ESP security service.

The possible encryption algorithms in a manual template are "DES" (Data Encryption Standard), "TDES" (Triple Data Encryption Standard)". You cannot select "AES" (Advanced Encryption Standard) in a manual template.

You can choose the "MD5", or "SHA1" or "NONE" authentication algorithms

The “ID” field is the Template identification number.

Example:

```
IPSec config>TEMPLATE 4 manual esp des md5
```

“TEMPLATE [ID] MANUAL AH [AUTHEN]”

This defines a manual Template with AH security service.

The possible authentication algorithms are “MD5” or “SHA1”.

The “ID” field identifies the Template.

Example:

```
IPSec config>TEMPLATE 5 manual ah sha1
```

Once the security service has been defined, you need to enter the IP addresses for the secure Tunnel ends, the SA identifier created from the Template (SPI) and the keys to be used with the chosen encryption and authentication algorithms.

“TEMPLATE [ID] SOURCE-ADDRESS [IP ADD]”

This introduces the Tunnel’s local IP address for the Template identified by [ID].

Example:

```
IPSec config>TEMPLATE 4 source-address 192.100.1.2
```

“TEMPLATE [ID] DESTINATION-ADDRESS [IP ADD/DOMAIN NAME]”

This introduces the IP address or the domain name of the other remote end of the Tunnel.

Example:

```
IPSec config>TEMPLATE 4 destination-address 192.100.1.1
```

In this particular case, IP address “192.100.1.1” is used as destination.

Example:

```
IPSec config>TEMPLATE 4 destination-address atlas.teldat.es
```

In this particular case, the domain name “atlas.teldat.es” is used as destination. You must bear in mind that you need to have a DNS server configured so this is able to resolve the domain name.

“TEMPLATE [ID] SPI [INTEGER > 256]”

Permits you to introduce the “Security Parameter Index” for the Template identified by [ID]. This number is an integer, [INTEGER], that must be higher than 256. The **SPI** must be the same at both ends, identifying a Template with respect to other Templates with the same Tunnel destination address and with the same security service (ESP or AH).

Example:

```
IPSec config>TEMPLATE 4 spi 280
```

You cannot define two policies that have identical values for the three said parameters: Tunnel destination address, security service and SPI.

“TEMPLATE [ID] KEY [8 bytes key]”

In order to introduce the key in cases where you have selected DES as the encryption algorithm. “8 bytes Key” represents the Template encryption DES key (you can introduce this in Hexadecimal, beginning with 0x, or in ASCII).

Example:

```
IPSec config>TEMPLATE 4 key 0x0123456789ABCDEF
```

Please note, if you decide to introduce the key in hexadecimal, you must introduce double the amount of characters (between 0-9 and A-F), as two hexadecimal characters define one byte.

"TEMPLATE [ID] TKEY [24 bytes key]"

In cases where you have selected Triple DES as encryption algorithm. "24 bytes Key" contains the Triple DES key (you can introduce this in Hexadecimal, beginning with 0x, or in ASCII).

Example:

```
IPSec config>TEMPLATE 4 tkey 0123456789abcdefghijklmn
```

"TEMPLATE [ID] MD5KEY [16 bytes key]"

If you have chosen MD5 for authentication, you need to provide a "16 bytes Key" (you can introduce this in Hexadecimal, beginning with 0x, or in ASCII).

Example:

```
IPSec config>TEMPLATE 4 md5key teldatsateldatsa
```

"TEMPLATE [ID] SHA1KEY [20 bytes key]"

In cases of selecting SHA1 for authentication, you must enter a "20 bytes Key" (you can introduce this in Hexadecimal, beginning with 0x, or in ASCII).

Example:

```
IPSec config>TEMPLATE 4 sha1key teldatsateldatsa1234
```

"TEMPLATE [ID] DF-BIT [SET, CLEAR, COPY]"

When encapsulating a packet in IPSec, the router executes a procedure so the Path Maximum Transfer Unit Discovery (PMTUD) algorithm continues to function in the hosts protected by the tunnel. This algorithm makes use of the DF bit (Don't Fragment) in the IP header and the ICMP packets. Through this option, you can specify which policy the router should follow in the processing of this bit in the IPSec packets: always mark (all the IPSec packets exit with the DF bit set as TRUE), always eliminate (bit set as FALSE, this does not process the ICMP/PMTUD packets and consequently the router acts as a "Black Hole" to all effects of this algorithm) or copies the packet being protected (normal ICMP/PMTUD) process and the router default option). For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>TEMPLATE 4 df-bit ?
set     set the DF bit on the IPSec packets
clear   clear the DF bit on the IPSec packets
copy    copy the DF bit from the inner header
IPSec config>TEMPLATE 4 df-bit clear
```

"TEMPLATE [ID] MTU-THRESHOLD [INTEGER 256..2000]"

Specifies the minimum MTU (Maximum Transfer Unit) that must be indicated to the hosts protected by the router as a consequence of ICMP/PMTU message processing. Default value is 576 bytes as, in the majority of the applications; this is a compromise value between the behavior of the network executing fragmentation or with an excessively low MTU. In either case however, this depends on the type of traffic circulating over the network. For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>TEMPLATE 4 mtu-threshold 576
```

"TEMPLATE [ID] MTU-DEFAULT [INTEGER 256..2000, DISABLED]"

Specifies the initial MTU (Maximum Transfer Unit) through the path defined by the IPSec Tunnel which should communicate to the hosts protected by the router. By default this value is disabled and should only be assigned if you have prior knowledge of the MTU through the path. For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>TEMPLATE 4 mtu-default ?
<256..2000>    set starting value for path MTU
disabled       disables starting value for path MTU
IPSec config>TEMPLATE 4 mtu-default 512
```

Once all the corresponding parameters and keys are defined, you need to introduce these in the other router through which you are going to establish the Tunnel. The final step is the association (mapping) between the LCA entries and the Templates i.e. the creation of the **SPDs** entries. This will be explained after configuring the dynamic Templates.

You can view or delete configured Templates through the same **LIST** and **NO** commands used for the access lists:

Command	Operation
LIST TEMPLATE	Displays the elements from the Templates list.
NO TEMPLATE	Deletes elements from the Templates list.

"LIST TEMPLATE ALL"

Displays all the elements in the Templates list.

Example:

```
IPSec config>LIST TEMPLATE ALL
TEMPLATES
4 manual  ESP-DES  ESP-MD5  SRC=192.100.1.2 DES=192.100.1.1 SPI=280

5 manual  AH-SHA1  SRC=192.100.1.2 DES=192.100.1.10  SPI=280
```

"LIST TEMPLATE ADDRESS-FILTER [IP ADD] [MASK]"

Displays the elements in the Templates list with Tunnel source or destination IP address that is included within the range defined by [IP ADD] and [MASK].

Example:

```
IPSec config>LIST TEMPLATE ADDRESS-FILTER 192.100.1.10  255.255.255.255
TEMPLATES
5 manual  AH-SHA1  SRC=192.100.1.2 DES=192.100.1.10  SPI=280
```

"NO TEMPLATE [ID]"

Deletes the element from the Templates list identified by [ID].

Example:

```
IPSec config>NO TEMPLATE 5
IPSec config>LIST TEMPLATE ALL
```

```
TEMPLATES
4 manual  ESP-DES  ESP-MD5  SRC=192.100.1.2 DES=192.100.1.1 SPI=280
```

- *Dynamic Templates (IPSec IKE)*

The IKE IPSec (dynamic IPSec) configuration requires two types of Templates: those known as **dynamic Templates**, which are the equivalent to the Templates configured in manual mode, and the **ISAKMP Templates**. At this point you need to negotiate the algorithms and the keys between the Tunnel ends in order to establish a communication SA. This is carried out in two phases:

- In the first phase, certain security parameters that protect the negotiation are agreed as well as authenticating both ends. These parameters are defined in the ISAKMP Templates.
- The second phase consists of the SA negotiation for the Tunnel. This is based in dynamic Templates.

As regards the TEMPLATE subcommands to create these Templates, some are common and others are only applicable to some of the other two types.

Command	Operation
DYNAMIC	Creates a dynamic Template with a security service (ESP or AH).
ISAKMP	Creates an ISAKMP Template with some security parameters.
SOURCE-ADDRESS	Introduces the address of the Tunnel source end in the Template.
DESTINATION-ADDRESS	Introduces the address of the Tunnel destination end in the Template.
DISCOVER	Indicates that TED must be used to search for the remote end of the Tunnel.
NO DISCOVER	Deactivates the TED protocol.
BACKUP-DESTINATION	Adds a backup destination address.
ANTIREPLAY	Activates the Anti-Replay service in the Template.
NO ANTIREPLAY	Deactivates the Anti-Replay service in the Template.
PADDING-CHECK	Checks that the IPSec header padding field takes the value indicated in the RFC.
NO PADDING-CHECK	The value of the IPSec header padding field is ignored.
UDP-ENCAPSULATION	To encapsulate IPSec packets in UDP packets.
NO UDP-ENCAPSULATION	To disable the option of encapsulating IPSec packets in UDP packets.
UDP-IKE	To encapsulate the IPSec IKE packets in UDP packets.
NO UDP-IKE	To disable the option of encapsulating the IPSec IKE packets in UDP packets.
AGGRESSIVE	Configures the sending of the encryption/clear from the third IKE message in aggressive mode.
ENCAP	Configures the Tunnel or Transport operation mode.

LIFE	Introduces the SAs life span created from the Template.
IKE	Configures parameters relative to the IPSec IKE mode.
KEEPALIVE	Enables or disables the available keepalive services.
NO	Deletes a backup address or disables an option.
FAST-FORWARDER	Forces the use of fast-forwarding IPSec packets.
NO FAST-FORWARDER	Disables the use of IPSec packet fast-forwarding.
INVALID-SPI-RECOVERY	Enables the invalid SPI reception notification regardless of the existence of the ISAKMP SA with the remote end.
NO INVALID-SPI-RECOVERY	Disables the "invalid-spi-recover" function.
DF-BIT	Indicates the process that must be given to the DF bit in the IPSec packets.
MTU-THRESHOLD	Indicates the minimum MTU threshold to use in the PMTU procedure.
MTU-DEFAULT	Indicates the initial value given to the MTU through the IPSec tunnel.
TCP-MSS-ADJUST	Adjusts the value of the MSS field for the TCP packets in transit.
NO TCP-MSS-ADJUST	Deactivates the adjustment of the value of the MSS field for the TCP packets in transit.
RRI-ENABLED	Enables the RRI (Reverse Route Injection).
NO RRI-ENABLED	Disables the RRI.
MAPPED-TO-IFC	Maps the template to an interface.
NO MAPPED-TO-IFC	Does not map the template to an interface.
ASSIGNED-ADDRESS-GOES-TO-IFC	The address received during the ISAKMP configuration is established in this interface.
NO ASSIGNED-ADDRESS-GOES-TO-IFC	The received address is not established in any interface.
SET-LABEL	The packets processed by IPSEC are marked with this label.
NO SET-LABEL	This deactivates the marking of packets processed by IPSec.
UNIQUE	Only one similar tunnel per entry on the access list.
NO UNIQUE	Does not restrict similar tunnels per entry on the access list.
PKT-SRC-CLIENT-SRC	Uses the original packet's original IP address as original client.
NO PKT-SRC-CLIENT-SRC	Does not use the original packet's original IP address as original client.
SEND-ORIGINAL-PKT	Sends the original packet that provoked tunnel creation after it had been established.
NO SEND-ORIGINAL-PKT	Does not send the original packet after the tunnel was established.

REPLACE-DESTINATION	Replaces the destination of the encapsulated packets with the tunnel destination.
NO REPLACE-DESTINATION	Deactivates the destination replacement of the encapsulated packets.
PREFRAGMENTATION	Enables the data prefragmentation process.
NO PREFRAGMENTATION	Disables the data prefragmentation process.
RRI-NEXTHOP	Configures the next hop that the RRI must use.
GDOI GROUP	Configures the dynamic template as a client in a GDOI group.
NO GDOI GROUP	Deconfigures the dynamic template as a client in a GDOI group.
VRF	Assigns the template in a VRF.
NO VRF	Does not assign the template to any VRF.
FAULT-TOLERANT	Tunnels opened with this template participate in the Fault Tolerant IPSec Recovery system.

- *ISAKMP Template Parameters*

The section will begin by describing the ISAKMP, as this is the first step in the negotiations.

The first thing to establish is the security parameters for the ISAKMP Template, under which the connection SA negotiation is carried out. As regards the ISAKMP Template, this also gives rise to a negotiation SA, or ISAKMP SA:

"TEMPLATE [ID] ISAKMP [ENCRYPT] [AUTHEN]"

The Template ISAKMP is created based on encryption and authentication algorithms. For encryption, the options are DES and Triple DES (TDES), AES128, AES192 and AES256, and as authentication MD5 and SHA1. The difference between the three types of AES encryption is the length of the used key (128, 192 and 256 bits, respectively). Despite the similarity, this is not the ESP service and the selection of an authentication algorithm is compulsory.

Example:

```
IPSec config>TEMPLATE 2 isakmp tdes sha1
```

Now you need to specify the address of the Tunnel end. The ISAKMP Templates do not require the source address.

"TEMPLATE [ID] DESTINATION [IP ADD/DOMAIN NAME]"

Example:

```
IPSec config>TEMPLATE 2 destination-address 192.100.1.1
```

In this particular case, IP address "192.100.1.1" is used as destination.

Example:

```
IPSec config>TEMPLATE 4 destination-address atlas.teldat.es
```

In this particular case, the domain name "atlas.teldat.es" is used as destination. You must bear in mind that you need to have a DNS server configured so this is able to resolve the domain name.

We also have the option to specify any address (0.0.0.0) as a remote address and use the TED protocol to dynamically discover the Tunnel remote end address or wait for the remote end to open the Tunnel.

"TEMPLATE [ID] DISCOVER"

With this option enabled in the template and selecting address 0.0.0.0 as destination, we can use a TED negotiation to establish the remote end of the Tunnel. Before configuring the router to discover remote addresses however, there are certain restrictions imposed on the TED protocol which should be borne in mind (see chapter 1).

Example:

```
IPSec config>TEMPLATE 2 discover
```

"TEMPLATE [ID] NO DISCOVER"

This deactivates the use of the TED protocol. Either because we are going to specify the remote end Tunnel address or wait for this latter to open the Tunnel or because we use the advanced option "PKT-DEST-ISAKMP-DEST" (this is explained further on in the manual).

Example:

```
IPSec config>TEMPLATE 2 no discover
```

"TEMPLATE [ID] BACKUP-DESTINATION [IP ADD]"

Adds a backup destination IP address.

It's possible to establish up to three backup destination addresses in the ISAKMP Templates, so that in cases where the Tunnel cannot be established with the main address, the backup addresses are used.

While the device is connected to the Backup address, the main address is polled to see if the session can be established with it. In this case, the session is established with the main address and the Backup session that was established is closed.

The address polling period has to be that calculated with the following formula:

Main Address Polling Period: ADVANCED DPD IDLE-PERIOD + ADVANCED DPD PACKETS * ADVANCED DPD INTERVAL (seconds).

Which with the default values results in:

Main Address Polling Period: 60 + 5 * 3 = 75 seconds.

Example:

```
IPSec config>TEMPLATE 2 backup-destination 192.100.1.2
```

"TEMPLATE [ID] NO BACKUP-DESTINATION [IP ADD]"

Deletes a backup destination IP address.

Example:

```
IPSec config>TEMPLATE 2 no backup-destination 192.100.1.2
```

Finally, there are various optional parameters with default values. However these can be modified if necessary:

"TEMPLATE [ID] UDP-ENCAPSULATION"

This command indicates if the IPSec packets should be encapsulated in UDP packets. This is usually used to cross Firewalls or devices executing NAPT without needing to change the configuration. This makes sense in cases of ISAKMP Templates.

Example:

```
IPSec config>TEMPLATE 2 udp-encapsulation
```

"TEMPLATE [ID] NO UDP-ENCAPSULATION"

This command indicates that IPSec packets are not encapsulated in UDP packets i.e. normal operation. This makes sense for ISAKMP Templates.

Example:

```
IPSec config>TEMPLATE 2 no udp-encapsulation
```

"TEMPLATE [ID] UDP-IKE"

This command indicates that the IPSec IKE packets must be encapsulated in UDP packets. This is usually used to cross Firewalls or devices executing NAPT, without having to change the configuration. This makes sense in cases of ISAKMP Templates.

Example:

```
IPSec config>TEMPLATE 2 udp-ike
```

"TEMPLATE [ID] NO UDP-IKE"

This command indicates that the negotiation IPSec packets should not be encapsulated in UDP packets, even though this encapsulation is being carried out with the data packets.

Example:

```
IPSec config>TEMPLATE 2 no udp-ike
```

"TEMPLATE [ID] AGGRESSIVE CIPHER/CLEAR"

This command indicates if the IKE negotiation third message in aggressive mode should be encrypted or not.

Example:

```
IPSec config>TEMPLATE 2 aggressive clear
```

"TEMPLATE [ID] ENCAP TUNNEL/TRANSPORT"

This command indicates if encapsulation is going to be carried out in tunnel or transport mode.

Example:

```
IPSec config>TEMPLATE 2 encap transport
```

"TEMPLATE [ID] LIFE DURATION SECONDS [VALUE]"

Permits you to introduce the lifetime of the SA negotiation, the default value is 3600 seconds (1 hour).

Example:

```
IPSec config>TEMPLATE 2 life duration seconds 1000
```

"TEMPLATE [ID] IKE MODE AGGRESSIVE/MAIN"

Phase 1 of the ISAKMP/IKE exchange can be carried out in two ways: Aggressive Mode and Main Mode. The first mode is faster than the second, but at the cost of a diminution of parameters to be negotiated.

Example:

```
IPSec config>TEMPLATE 2 ike mode aggressive
```

"TEMPLATE [ID] IKE METHOD PRESHARED/RSA"

Establishes the authentication method used by the device. In principal, only the Pre-shared key method is available.

Example:

```
IPSec config>TEMPLATE 2 ike method preshared
```

"TEMPLATE [ID] IKE IDTYPE IP/FQDN/UFQDN/KEYID/ASN-DN"

Phase 1 of the ISAKMP /IKE exchange can be carried out by using different types of identifiers:

(See IKE-ID command)

IP: indicates that the own IP address will be used to identify the device.

FQDN ("Fully Qualified Domain Name"): used to identify a text string equivalent to the TCP/IP address for a network interface. E.g. if you have configured the host name "atlas1" and domain name "teldat.es" in the device, the device fqdn used for identification in IPSec will be "atlas1.teldat.es". This method can only be used in AGGRESSIVE mode.

If you do not correctly configure a domain name, the device will only use the host name followed by a period (".") which indicates the root domain. This should therefore be taken into account for the remote end.

UFQDN ("User - Fully Qualified Domain Name"): includes a specific user within the machine with an SMTP mail address format (user@host.domain.xx). In cases where there are no users in the device, the same as that shown above is sent. This method can only be used in AGGRESSIVE mode.

KEYID: carries out identification through a binary stream used to pass specific information from the device manufacturer. Teldat uses the hostname configured in the device in this case, without taking into account domain or subdomain names. This method can only be used in AGGRESSIVE mode.

ASN-DN: specifies the "Distinguished Name" (DN) binary DER codification for the main certificate from those being exchanged to establish the SA, as defined in the ASN.1 X.500 standard.

The remote device will use the received identifier and will search in its key table (Pre-shared Keys) associated to devices (IP addresses or Hostnames) created with the KEY IP/HOSTNAME command (this will be seen further on).

Example:

```
IPSec config>TEMPLATE 2 ike idtype ip
```

"TEMPLATE [ID] IKE ID [NAME/Dir IP]"

Establishes the identifier used for the ISAKMP/IKE exchange in phase 1.

If the template has IKE IDTYPE IP configured, this parameter specifies an IP address. Contrariwise, this is a string of characters.

If this parameter is not established, the identifier used in the IKE phase 1 will be that previously specified in the IKE IDTYPE command description.

Example:

```
Office IPSec config>TEMPLATE 2 ike idtype key
```

The identifier "Office" is used in this case.

Example:

```
Office IPSec config>TEMPLATE 2 ike idtype key
Office IPSec config>TEMPLATE 2 ike id MyOffice
```

The identifier "MyOffice" is used in this case.

Example:

If the device IP has IP address 10.0.0.1

```
Office IPSec config>TEMPLATE 2 ike idtype ip
```

The identifier used in this case is IP "10.0.0.1".

Example:

If the device IP has IP address 10.0.0.1

```
Office IPSec config>TEMPLATE 2 ike idtype ip
Office IPSec config>TEMPLATE 2 ike id 1.1.1.1
```

The identifier used in this case is IP "1.1.1.1".

"TEMPLATE [ID] IKE GROUP ONE/TWO/FIVE/FIFTEEN"

Establishes the type of Oakley group. The greater the index for the configured group, the longer the negotiation takes as it requires more processing. Group 1 is used by default.

Example:

```
IPSec config>TEMPLATE 2 ike group one
```

"TEMPLATE [ID] IKE FRAGMENTATION [DISABLE/FORCE]"

Disables/forces fragmentation of IKE packets before they are sent. The behavior is as follows:

The device fragments IKE negotiation packets 5 and 6 with the RSA and XAUTH-INIT-RSA method provided that:

- The remote end indicates that it supports this feature by sending the corresponding payload vendor and the DISABLE option is not configured for this command.
- That the FORCE option is configured for this command.

Example:

```
IPSec config>template 4 ike fragmentation force
```

"TEMPLATE [ID] IKE NO FRAGMENTATION"

The device fragments IKE negotiation packets 5 and 6 with the RSA and XAUTH-INIT-RSA method provided that the remote end indicates that it supports this feature by sending the corresponding *payload vendor*. This is the default behavior.

Example:

```
IPSec config>template 4 ike no fragmentation
```

"TEMPLATE [ID] IKE LIFETIME-NEGOTIATION/ NO IKE LIFETIME-NEGOTIATION"

This enables the sending of a lifetime proposal in the negotiation. I.e. the device proposes a lifetime which is negotiated. By default this option is enabled.

In cases where "ike no lifetime-negotiation" is configured, the device does not send a lifetime proposal. This is useful when:

- The device is operating with another device that does not admit negotiation if there is a lifetime proposal.

- When you want the remote end to set the lifetime.

Warning: there are some devices that do not allow the lifetime proposal parameter to be disabled and reject negotiation should it be disabled.

Example:

```
IPSec config>TEMPLATE 2 lifetime-negotiation
```

"TEMPLATE [ID] IKE PKT-DEST-ISAKMP-DEST"

The packet destination indicates the Tunnel's remote address. In this way you do not need to configure the template destination address as it is given by the packet destination.

Example:

```
IPSec config>template 4 ike pkt-dest-isakmp-dest
```

"TEMPLATE [ID] IKE EARLY-RETRY"

If a response hasn't been received, IKE negotiation is re-tried after ¼ of the PURGE-TIMEOUT time has lapsed. This is configured through the ADVANCED PURGE-TIMEOUT command.

This option is useful in WWAN environments where the first packets are often lost and if the traffic is very scattered so the packets aren't ever progressed.

Example:

```
IPSec config>template 4 ike early-retry
```

"TEMPLATE [ID] IKE NATT-VERSION"

Through this parameter, the remote end is informed of the type of NAT-Traversal which it wants to negotiate. In cases where the remote end initiates negotiation, the device adapts to what this proposes provided it is within the supported versions. The list of supported versions is as follows:

- RFC: rfc 3947
- DRAFT-V3: Natt version draft-3
- DRAFT-V2-N: Natt version draft-2-n
- DRAFT-V2: Natt version draft-2
- NONE: Disables this functionality. I.e. does not inform the remote end that NAT-Traversal is supported.

Example:

```
IPSec config>TEMPLATE 2 ike natt-version
```

"TEMPLATE [ID] SEND-ORIGINAL-PKT"

The creation of the IPSec tunnels is provoked when a packet, that needs to be encrypted when the corresponding tunnel still doesn't exist, arrives or is generated. Normally this packet, the original packet, isn't sent through the tunnel when it has already been established. By configuring this command, the said packet is saved and is sent after the tunnel has established. By default, this command is not configured.

Successive packets received during establishment are also stored for subsequent sending, up to a maximum of 8.

Example:

```
IPSec config>template 2 send-original-pkt
```

"TEMPLATE [ID] NO SEND-ORIGINAL-PKT"

When this command is configured, the original packet is not sent after the tunnel has been established; it only saves and sends the said packet in very specific cases where it is normally executed in the absence of the SEND-ORIGINAL-PKT command.

Example:

```
IPSec config>template 2 no send-original-pkt
```

Through this, all the parameters relative to the ISAKMP Templates are configured. When the router wishes to establish a security Tunnel, it first sends its appropriate ISAKMP Template proposals to the other end (depending on the destination IP address) and both have to reach an agreement on which Template is to be used.

Once the SA negotiation is established, the agreement must take into account the **dynamic Template** in order to create the connection SA.

- *Dynamic Template Parameters*

"TEMPLATE [ID] DYNAMIC ESP [ENCRYPT] [AUTHEN]"

A dynamic Template is defined with ESP security service, selecting encryption between DES and TDES, AES128, AES192 and AES256, and authentication between MD5, SHA1 or NONE. The difference between the three types of AES encryption is the length of the used key (128, 192 and 256 bits, respectively).

Example:

```
IPSec config>TEMPLATE 4 dynamic esp tdes sha1
```

"TEMPLATE [ID] DYNAMIC AH [AUTHEN]"

A dynamic Template is defined with AH security service, choosing between MD5 and SHA1.

Example:

```
IPSec config>TEMPLATE 3 dynamic ah md5
```

"TEMPLATE [ID] SOURCE-ADDRESS [IP ADD]"

To introduce the local IP address of the Tunnel. Please note that is only necessary to define this for the dynamic Templates.

This address may be unnumbered i.e. you can set an address associated to an interface which is unknown at the time of configuring the device as, for example, it will be assigned by another mechanism such as PPP.

If this is set to 0.0.0.0, i.e. not configured, the output interface address is taken as source address.

Example:

```
IPSec config>TEMPLATE 4 source-address 192.100.1.2
```

"TEMPLATE [ID] DESTINATION-ADDRESS [IP ADD/DOMAIN NAME]"

This introduces the address of the remote end of the Tunnel.

Example:

```
IPSec config>TEMPLATE 4 destination-address 192.100.1.1
```

In this particular case, IP address "192.100.1.1" is used as destination.

Example:

```
IPSec config>TEMPLATE 4 destination-address atlas.teldat.es
```

In this particular case, the domain name "atlas.teldat.es" is used as destination. You must bear in mind that you need to have a DNS server configured so this is able to resolve the domain name.

If the remote Tunnel address is 0.0.0.0, this is considered unknown and is not a significant parameter for selecting the dynamic Template during negotiation. Given that the destination address is unknown, only the remote end can begin IKE negotiation.

The following subcommands refer to the established default values; however it might be appropriate to modify these depending on the circumstances.

"TEMPLATE [ID] ANTIREPLAY":

This command enables the Anti-Replay service. This is a security method to avoid attacks based on packet retransmission.

Example:

```
IPSec config>TEMPLATE 3 antireplay
```

"TEMPLATE [ID] NO ANTIREPLAY"

Disables the Anti-Replay service.

Example:

```
IPSec config>TEMPLATE 3 no antireplay
```

"TEMPLATE [ID] PADDING-CHECK"

The original IPSec RFC permitted you to fill out the IPSec header padding field with any random value. The current RFC however specifies a determined value for the said field. So that the router can operate with devices which comply with the original RFC, you can configure a parameter indicating if a check should be carried out on whether the padding field takes the value defined in the RFC or if this data should be ignored.

Example:

```
IPSec config>TEMPLATE 3 padding-check
```

"TEMPLATE [ID] NO PADDING-CHECK"

The IPSec header padding field will not be checked.

Example:

```
IPSec config>TEMPLATE 3 no padding-check
```

"TEMPLATE [ID] LIFE TYPE SECONDS/KBYTES/BOTH"

Permits you to introduce the type of life duration for the communication SA based on the dynamic Template. In the dynamic Templates, the lifetime can be represented as a time limit ("SECONDS"), in the same way as for the ISAKMP Templates, or also as a quantity limit of transmitted bytes (KBYTES") through the SA generated with this Template.

The third option ("BOTH") establishes both limits at the same time. In this case the SA will delete when one of the two limits expire.

Example:

```
IPSec config>TEMPLATE 4 life type both
```

"TEMPLATE [ID] LIFE DURATION SECONDS/KBYTES [VALUE]"

The chosen life duration is shown in the VALUE field. In cases where you have selected BOTH in the previous subcommand, you will have to enter the subcommand twice in order to give both types of values (seconds and kilobytes).

Example:

```
IPSec config>TEMPLATE 4 life duration seconds 20000
IPSec config>TEMPLATE 4 life duration kbytes 1000
```

"TEMPLATE [ID] IKE PFS"

This enables the Perfect Forward Secrecy service. This increases the security of the created SAs, making for a better management of the used keys.

Example:

```
IPSec config>TEMPLATE 4 ike pfs
```

"TEMPLATE [ID] IKE NO PFS"

This disables the Perfect Forward Secrecy service.

Example:

```
IPSec config>TEMPLATE 4 ike no pfs
```

"TEMPLATE [ID] KEEPALIVE KEEPALIVE"

Enables the Keep Alive service for maintenance of the SAs.

Example:

```
IPSec config>TEMPLATE 4 keepalive keepalive
```

"TEMPLATE [ID] KEEPALIVE NO KEEPALIVE"

Disables the Keep Alive service for maintenance of the SAs.

Example:

```
IPSec config>TEMPLATE 4 keepalive no keepalive
```

"TEMPLATE [ID] KEEPALIVE DPD"

Enables the DPD service (Dead Peer Detection) for maintenance of the SAs. This makes sense in cases of ISAKMP Templates.

Example:

```
IPSec config>TEMPLATE 2 keepalive dpd
```

"TEMPLATE [ID] KEEPALIVE NO DPD"

Disables the DPD service (Dead Peer Detection) for maintenance of the SAs. This makes sense in cases of ISAKMP Templates.

Example:

```
IPSec config>TEMPLATE 2 keepalive no dpd
```

"TEMPLATE [ID] FAST-FORWARDER"

Forces the use of the packet routing fast forwarding with the aim of gaining speed. This makes sense in cases of Dynamic Templates provided that there is not going to be any subsequent manipulation (NAT for example) of the IPSec packets whether this takes place before or after encapsulating them.

Example:

```
IPSec config>TEMPLATE 2 fast-forwarder
```

"TEMPLATE [ID] NO FAST-FORWARDER"

Deactivates the use of IPSec packet routing fast forwarder.

Example:

```
IPSec config>TEMPLATE 2 no fast-forwarder
```

"TEMPLATE [ID] INVALID-SPI-RECOVERY"

Enables the notification function for the reception of a packet with invalid SIP regardless of the existence of an ISAKMP with the remote end. In cases where an invalid SIP is received when there isn't an ISAKMP SA created with the other end, a new ISAKMP SA is opened where it reports that an invalid SPI has been received so the remote end deletes that SA with the said SPI.

Regardless of the configuration of this parameter, if there is an ISAKMP SA with the remote end, when an invalid SPI is received the receiver sends a report.

This function is disabled by default.

Example:

```
IPSec config>TEMPLATE 2 invalid-spi-recovery
```

"TEMPLATE [ID] NO INVALID-SPI-RECOVERY"

Disables the "INVALID-SPI-RECOVERY" function.

Example:

```
IPSec config>TEMPLATE 2 no invalid-spi-recovery
```

"TEMPLATE [ID] GDOI GROUP [VALUE]"

Configures the dynamic template as client in a GDOI group. The value configured is the GDOI group ID. The GDOI server address is that configured as destination in the template.

Example:

```
IPSec config>TEMPLATE 2 gdoi group 1
```

"TEMPLATE [ID] NO GDOI GROUP"

Deactivates the use of the GDOI protocol in the dynamic template.

Example:

```
IPSec config>TEMPLATE 2 no gdoi group
```

"TEMPLATE [ID] DF-BIT SET/CLEAR/COPY"

When encapsulating a packet in IPSec, the router executes a procedure so the Path Maximum Transfer Unit Discovery (PMTUD) algorithm continues to function in the hosts protected by the tunnel. This algorithm makes use of the DF bit (Don't Fragment) in the IP header and the ICMP packets. Through this option, you can specify which policy the router should follow in the processing of this bit in the IPSec packets: always mark (all the IPSec packets exit with the DF bit set as TRUE), always eliminate (bit set as FALSE, this does not process the ICMP/PMTUD packets and consequently the router acts as a "Black Hole" to all effects of this algorithm) or copies the packet being protected (normal

ICMP/PMTUD process and the router default option). For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>template 4 df-bit ?
set     set the DF bit on the IPSec packets
clear   clear the DF bit on the IPSec packets
copy    copy the DF bit from the inner header
IPSec config>template 4 df-bit clear
```

"TEMPLATE [ID] MTU-THRESHOLD [INTEGER 256..2000]"

Specifies the minimum MTU (Maximum Transfer Unit) that must be indicated to the hosts protected by the router as a consequence of ICMP/PMTU message processing. Default value is 576 bytes as, in the majority of the applications; this is a compromise value between the behavior of the network executing fragmentation or with an excessively low MTU. In either case however, this depends on the type of traffic circulating over the network. For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>template 4 mtu-threshold 576
```

"TEMPLATE [ID] MTU-DEFAULT [INTEGER 256..2000, DISABLED]"

Specifies the initial MTU (Maximum Transfer Unit) through the path defined by the IPSec Tunnel which should communicate to the hosts protected by the router. By default this value is disabled and should only be assigned if you have prior knowledge of the MTU through the path. For further information, please see RFC 1191, Section 6.

Example:

```
IPSec config>template 4 mtu-default ?
<256..2000>    set starting value for path MTU
disabled       disables starting value for path MTU
IPSec config>template 4 mtu-default 512
```

"TEMPLATE [ID] TCP-MSS-ADJUST [(INTEGER 536..65535) / CLAMPING [HEADER_LENGTH]]"

Specifies the MSS option adjust value for the TCP packets transmitted by the security association, SA, created from this template.

Through the CLAMPING option, you indicate a value to subtract from the SA MTU size. The SA MTU value depends on the lowest value of the following possibilities:

- Value configured with the DEFAULT-MTU option.
- Value learnt by PMTUD.
- MTU value for the mapped interface with the MAPPED-TO-IFC option.

By default the HEADER_LENGTH is 40 bytes.

Example:

```
IPSec config>template 4 tcp-mss-adjust ?
  <536..65535>    Adjust the mss of transit packets
  clampling       Automatically adjust the mss
IPSec config>template 4 tcp-mss-adjust clampling ?
  <cr>             Typical TCP/IP header length (40 bytes)
  header-length    Especify TCP/IP header length
```

When an integer is specified, this indicates the value that the TCP.SYN packet MSS option changes to, provided this is lower than the value already in the packet.

Example 1:

```
IPSec config>template 4 tcp-mss-adjust 1100
```

With this option, you can ensure that the MSS for the TCP connections established through the SAs created from this template is not bigger than 1100 bytes.

Example 2:

```
IPSec config>template 4 tcp-mss-adjust clampling
```

With this option, you can ensure that the MSS for the TCP connections established through the SAs created from this template is not bigger than the size of the MTU less 40 bytes (default value).

Example 3:

```
IPSec config>template 4 tcp-mss-adjust clampling header-length 60
```

With this option, you can ensure that the MSS for the TCP connections established through the SAs created from this template is not bigger than the size of the MTU less 60 bytes.

"TEMPLATE [ID] NO TCP-MSS-ADJUST"

Disables the adjust for the MSS option for the TCP packets transmitted by the SA security association, created from this template. This is the default behavior.

Example:

```
IPSec config>template 4 no tcp-mss-adjust
```

"TEMPLATE [ID] RRI-ENABLED"

Enables RRI in the dynamic template so when the tunnel opens using this template a static route is created in the routing table. This will have the network (or subnet or host) indicated by the negotiated remote clients as its destination and as the next hop, a value that depends on the option selected with the RRI-NEXTHOP command (this will be seen further on). This route will survive as long as the tunnel remains open and can be broadcast by the traditional dynamic routing algorithms (RIP, OSPF, etc.) in the normal way.

Example:

```
IPSec config>template 4 rri-enabled
```

"TEMPLATE [ID] NO RRI-ENABLED"

Disables the use of RRI in the dynamic template.

Example:

```
IPSec config>template 4 no rri-enabled
```

"TEMPLATE [ID] RRI-NEXTHOP SOURCE/DESTINATION/USER-DEFINED[IP ADD]"

Configures the next hop that the router must use when establishing static routes for RRI. There are three possibilities for this:

"SOURCE" (default value) where the local tunnel end address is used as the next hop for the static route. This is specifically for when the tunnel source interface is point-to-point or when all the traffic using the route is going to be encrypted and the remote end is different from the negotiated client.

"DESTINATION" where the remote end address is used. This is specifically for when the tunnel end is in a network directly connected or when an address can be directly resolved through route recursion without depending on a default route.

"USER-DEFINED [IP ADD]" where the user must specify the IP for the next hop to hand. This is for specific cases which do not fit into any of the other scenarios.

Example:

```
IPSec config>template 4 rri-nexthop ?
  source          tunnel source address
  destination     tunnel destination address
  user-defined    user-defined next hop
IPSec config>template 4 rri-nexthop user-defined ?
  <a.b.c.d>    Ipv4 format
IPSec config>template 4 rri-nexthop user-defined 10.10.10.1
```

"TEMPLATE [ID] MAPPED-TO-IFC"

Associates the dynamic template to an interface. In this way the device knows that it can only apply the template if traffic is running over this interface.

Example:

```
IPSec config>template 4 mapped-to-ifc ppp1
```

The use of this command has many implications, therefore we recommend that you thoroughly understand how it is used before configuring it. To clarify the concepts, you will see some examples below together with their explanations.

Example 1:

```
feature access
   access-list 100
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.0.28 255.255.255.255
      entry 1 destination address 10.127.1.29 255.255.255.255
;
   exit
exit
;
protocol ip
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 100
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 destination-address 192.168.169.29
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address serial0/0
      template 2 destination-address 192.168.169.29

      map-template 100 2
```

In **Example 1**, all traffic with source 10.127.0.28 and destination 10.127.1.29 is protected by IPSec independently of the interface it is running over. I.e.

- Packets with source 10.127.0.28 and destination 10.127.1.29 which leave the device are **encapsulated** in an IPSec tunnel.

- Packets with source 10.127.1.29 and destination 10.127.0.28 entering the device must do this **encapsulated** in an IPSec tunnel. Contrariwise, the packet will be dropped.

If you map template 2 to serial0/0 interface, we'll be left with **Example 2**.

Example 2:

```
feature access
   access-list 100
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.0.28 255.255.255.255
      entry 1 destination address 10.127.1.29 255.255.255.255
;
   exit
exit
;
protocol ip
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 100
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 destination-address 192.168.169.29
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address serial0/0
      template 2 destination-address 192.168.169.29
      template 2 mapped-to-ifc serial0/0

      map-template 100 2
```

In Example2, **only** traffic with source 10.127.0.28 and destination 10.127.1.29 **which is sent over the serial0/0 interface** is protected by IPSec. I.e.

- Packets with source 10.127.0.28 and destination 10.127.1.29 which leave the device through the **serial0/0** interface are **encapsulated** in an IPSec tunnel.
- Packets with source 10.127.0.28 and destination 10.127.1.29 which leave the device through a **different** interface from the serial0/0 do this in **clear**.
- Packets with source 10.127.1.29 and destination 10.127.0.28 which enter the device through the **serial0/0** interface must do so **encapsulated** in an IPSec tunnel. Contrariwise, the packet is dropped.
- Packets with source 10.127.1.29 and destination 10.127.0.28 which enter the device through a **different** interface from the serial0/0 **are sent normally**, although it hasn't been encapsulated in IPSec.

If we now add a template which is not mapped to any interface, we get **Example 3**.

Example 3:

```
feature access
   access-list 100
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.0.28 255.255.255.255
      entry 1 destination address 10.127.1.29 255.255.255.255
;
   exit
;
```

```
protocol ip
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 100
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 destination-address 192.168.169.29
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address serial0/0
      template 2 destination-address 192.168.169.29
      template 2 mapped-to-ifc serial0/0

      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address serial0/0
      template 3 destination-address 192.168.169.29

      map-template 100 2
      map-template 100 3
```

In **Example 3**, **only** traffic with source 10.127.0.28 and destination 10.127.1.29 **which is sent over the serial0/0 interface** is protected by IPSec in tdes and md5 mode **as well as the rest of the traffic**. I.e.

- Packets with source 10.127.0.28 and destination 10.127.1.29 which leave the device through the **serial0/0** interface are **encapsulated** in an IPSec tunnel.
- Packets with source 10.127.0.28 and destination 10.127.1.29 which leave the device through a **different** interface from the serial0/0 are **encapsulated** in an IPSec tunnel.
- Packets with source 10.127.1.29 and destination 10.127.0.28 which enter the device through the **serial0/0** interface must do so **encapsulated** in an IPSec tunnel. Contrariwise, the packet is dropped.
- Packets with source 10.127.1.29 and destination 10.127.0.28 which enter the device through a **different** interface from the serial0/0 must do so **encapsulated** in an IPSec tunnel. Contrariwise, the packet is dropped.

Finally, we have added a configuration example, Example4, which allows you to use two different tunnels to send the same traffic. One tunnel or the other is used depending on what the IP routes indicate. This configuration is usually used when you have a device with two interfaces where one backs up the other, and the rest of the interfaces are protected from unsafe access, e.g. the local LAN or something similar.

Example 4:

```
feature access
   access-list 100
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.0.28 255.255.255.255
      entry 1 destination address 10.127.1.29 255.255.255.255
;
   exit
exit
;
protocol ip
   ipsec
```

```
; -- IPSec user configuration --
     enable
      assign-access-list 100
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 destination-address 192.168.169.29
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address serial0/0
      template 2 destination-address 192.168.169.29
      template 2 mapped-to-ifc serial0/0

      template 3 default
      template 3 isakmp tdes md5
      template 3 destination-address 1.1.1.2

      template 4 default
      template 4 dynamic esp tdes sha
      template 4 source-address ppp1
      template 4 destination-address 1.1.1.2
      template 4 mapped-to-ifc ppp1

      map-template 100 2
      map-template 100 4
```

In **Example 4**, if the IP routes indicate that the path to reach address 10.127.1.29 is through the serial0/0 interface, traffic with source 10.127.0.28 and destination 10.127.1.29 is protected by IPSec, in tdes and md5 mode, via the tunnel, with source being the serial interface IP address and destination address 192.168.169.29. If the IP routes change state and indicate that in order to reach address 10.127.1.29 you need to go over ppp1 interface, this traffic is protected by IPSec, in tdes and sha mode, via the tunnel with source being the IP address assigned to the ppp1 and destination 1.1.1.2.

It's obvious that both tunnels can remain established at the same time and that the traffic can be sent through one and return via the other if the IP routes indicate this.

IMPORTANT. If the template is mapped to an interface and in addition it has the *fast-forwarding* functionality enabled, when the packet is encapsulated by IPSec, it is immediately sent to the interface. I.e. it does not rejoin the queue in the forwarder in order to be routed, which is what normally happens. This therefore, does not follow the schema explained in the section on *Packet Processing with IPSec*.

With this configuration, in addition to the packets moving faster as they are sent over a faster path, you can also achieve a special treatment for the packets once they've been encapsulated in IPSec. E.g. you can get the packets to exit without executing NAT; the outgoing route is the one that has packets before being encapsulated, etc.

See the *fast-forwarder* command for further information.

"TEMPLATE [ID] NO MAPPED-TO-IFC"

Does not associate the dynamic template to any interface.

Example:

```
IPSec config>template 4 no mapped-to-ifc
```

"TEMPLATE [ID] ASSIGNED-ADDRESS-GOES-TO-IFC"

The address received during the ISAKMP configuration is established as the main address in the interface configured in this option.

This behavior is similar to that described in the section on "ASSIGNED IP ADDRESS DESTINATION" using the "ADVANCED ADDRESS-ASSIGNED-TO-IFC" command; but in this case, it only takes effect if this template is selected during the negotiation.
I.e. unlike the "ADVANCED ADDRESS-ASSIGNED-TO-IFC" command, by using this template option, you can select what interface the address will be established in, in cases where there is more than one alternative of possible interfaces.

Example:

```
IPSec config>template 2 assigned-address-goes-to-ifc loopback1
```

"TEMPLATE [ID] NO ASSIGNED-ADDRESS-GOES-TO-IFC"

The address received during the ISAKMP configuration only depends on the configuration of the ADVANCED ADDRESS-ASSIGNED-TO-IFC command.

Example:

```
IPSec config>template 2 no assigned-address-goes-to-ifc
```

"TEMPLATE [ID] SET-LABEL ENCODED/DECODED"

The packets encoded/decoded by IPSec are marked with the corresponding label that was configured with this option.
The IKE packets are only marked on output with that configured in the SET-LABEL ENCODED option from the ISAKMP template. The received IKE packets aren't marked.
If the ISKAMP template does not have the SET-LABEL ENCODED configured, but the matching DYNAMIC template does, the IKE packets are marked on output with the label configured in the DYNAMIC template.

Example:

```
IPSec config>template 2 set-label encoded 2
```

"TEMPLATE [ID] NO SET-LABEL ENCODED/DECODED"

This deactivates marking the packets with label.

Example:

```
IPSec config>template 2 no  set-label encoded
```

"TEMPLATE [ID] UNIQUE"

Determines that there cannot be more than one tunnel with similar characteristics associated to the same entry in an access control list. By default this option is disabled.

Example:

```
IPSec config>template 2 unique
```

Given that the same access list can be mapped to various templates, all templates that use it must have the same value configured in this option.

The process of creating a new IPSec tunnel goes through the installation for a new dynamic "DX" entry on the access list; the new entry is related to one of the static "E" entries configured in the list. The IPSec tunnel is exclusively associated to the said dynamic "DX" entry. When the tunnel has fully established and if the unique option is configured, it goes over the other dynamic entries that are related to the same "E" entry. If it finds a previous dynamic "DY" entry with characteristics similar to "DX", it eliminates the tunnel associated to the said "DY" entry which is subsequently deleted. The

characteristics compared between the "DX" and "DY" entries to determine similarity are: same source and destination, protocol, ports, VRF and action.

An example is given below to demonstrate this functionality:

Example 1:

```
feature access
   access-list 100
      entry 1 default
      entry 1 permit
      entry 1 source address 10.122.1.0 255.255.255.0
      entry 1 destination address 10.121.1.0 255.255.255.0
;
   exit
exit
;
protocol ip
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 100
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 destination-address 192.168.121.8
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address ethernet0/0.10
      template 2 destination-address 192.168.121.8
      template 2 mapped-to-ifc ethernet0/0.10
      template 2 unique
;
      template 3 default
      template 3 isakmp tdes md5
      template 3 destination-address 192.168.121.5
;
      template 4 default
      template 4 dynamic esp tdes md5
      template 4 source-address ethernet0/0.20
      template 4 destination-address 192.168.121.5
      template 4 mapped-to-ifc ethernet0/0.20
      template 4 unique
;
      map-template 100 2
      map-template 100 4
```

Complying with the configuration in **Example 1**, the device can establish a tunnel to encapsulate traffic between networks 10.121.1.0/24 and 10.122.1.0/24 where the outgoing interface is the ethernet0/0.10 or the ethernet0/0.20. Both tunnels cannot exist: if you create a tunnel whose local interface is one of these then the tunnel whose local interface is the other one is eliminated (should this case exist). The most recently created tunnel always prevails while the older one is eliminated.

"TEMPLATE [ID] NO UNIQUE"

Disables the restriction so there can only be one tunnel with similar characteristics associated to a same entry on the access control list.

Example:

```
IPSec config>template 2 no unique
```

"TEMPLATE [ID] PKT-SRC-CLIENT-SRC"

During the establishment of a locally originated tunnel, a check is executed to see if this command has been configured. If it has then it takes the packet's original IP which provokes the tunnel to be created

to use it as the local client for the new tunnel. Thanks to this, individual tunnels can be established for multiple clients without needing to specify an access control list for each one; they are simply included in bigger access control lists. Although each individual tunnel shares the same access control list, they can be independently managed. By default this command is disabled.

Example

```
IPSec config>template 4 pkt-src-client-src
```

"TEMPLATE [ID] NO PKT-SRC-CLIENT-SRC"

This disables the feature that takes the packet's original IP which provokes a tunnel to be created as client of the said tunnel.

Example:

```
IPSec config>template 4 no pkt-src-client-src
```

"TEMPLATE [ID] PREFRAGMENTATION"

Enables the packet fragmentation process before being encapsulated by the SA. The SA MTU value depends on the lowest value of the following possibilities:

- Value configured with the DEFAULT-MTU option.
- Value learnt by PMTUD.
- MTU value for the mapped interface with the MAPPED-TO-IFC option.

Example:

```
IPSec config>template 4 prefragmentation
```

If this option is configured, in cases where a packet is received that is larger than the SA MTU with the *don't fragment* bit in the IP header, the packet is dropped and a PMTUD packet is sent to the source, independently of the value configured with the TEMPLATE [ID] DF-BIT command.

The packet fragmentation process is incompatible with encapsulation in Transport mode, TEMPLATE [ID] ENCAP TRANSPORT.

"TEMPLATE [ID] NO PREFRAGMENTATION"

Disables packet fragmenting process before being encapsulated by the SA. This is the default behavior.

Example:

```
IPSec config>template 4 no prefragmentation
```

"TEMPLATE [ID] VRF"

Assigns the dynamic template to a VRF. Consequently the device knows to only apply the template if traffic is being transmitted by the said VRF.

Example:

```
IPSec config>template 2 vrf VRF1
```

In this command you also have to bear in mind if the template is associated to an interface (through the command), so if this is correct then the said interface must also be associated to the vrf which you want to assign the template, contrariwise an error message will appear. I.e. if for example the

ethernet0/0 interface is associated to a vrf called VRFE1, and the dynamic template is associated with the said interface, an error is produced if you try to assign the template to a different vrf.

Example:

```
IPSec config>template 2 vrf VRF2
CLI Error: The mapped interface is not associated to this vrf
CLI Error: Command error
IPSec config>
```

To further clarify, below you can see some examples that explain how these commands function:

In these examples, we start with a router where 2 VRFs, the VRF1 and VRF2, have been configured and from there 2 Ethernet subinterfaces have been created, each one associated to one of these VRFs, both with the same IP address.

```
   network ethernet0/0.10
; -- Ethernet Subinterface Configuration --
      ip vrf forwarding VRFE1
;
      ip address 192.168.212.201 255.255.254.0
;
;
;
      encapsulation dot1q 10
;
;
   exit
;
   network ethernet0/0.20
; -- Ethernet Subinterface Configuration --
      ip vrf forwarding VRFE2
;
      ip address 192.168.212.201 255.255.254.0
;
;
;
      encapsulation dot1q 20
;
;
   exit
;
```

Example 1:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      Exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
```

```
;
        template 2 default
        template 2 dynamic esp tdes sha1
        template 2 source-address 192.168.212.201
        template 2 encap transport
;
        map-template 100 2
```

In **Example 1**, all traffic from source 192.168.212.201 is protected by IPSec independently to the VRF it pertains to. I.e.

- Packets with source 192.168.212.201 leaving the device are **encapsulated** in an IPSec tunnel whether it's being transmitted from the subinterface ethernet0/0.10 or ethernet0/0.20.
- Packets with destination 192.168.212.201 entering the device must do so **encapsulated** in an IPSec tunnel. If they aren't, they are dropped.

If you assign the template to a VRF, we have **Example 2**:

Example 2:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
;
         template 2 default
         template 2 dynamic esp tdes sha1
         template 2 source-address 192.168.212.201
         template 2 encap transport
         template 2 vrf VRF1
;
         map-template 100 2
```

In **Example 2 only** traffic with source 192.168.212.201 **which is transmitted through vrf VRF1**, i.e. **through an interface associated to vrf VRF1**, is protected by IPSec. I.e.

- Packets with source 192.168.212.201 exiting the device through vrf **VRF1**, i.e. through the ethernet0/0.10 interface, are **encapsulated** in an IPSec tunnel.
- Packets with source 192.168.212.201 exiting the device through a **different** vrf to VRF1 do so **in plain**, i.e. they exit through the ethernet0/0.20 interface.
- Packets with destination 192.168.212.201 entering the device through vrf **VRF1**, i.e. through the ethernet0/0.10 interface, must do so **encapsulated** in an IPSec tunnel. If they are not, the packets are dropped.

- Packets with destination 192.168.212.201 exiting the device through a **different** vrf to VRF1 are **transmitted normally**, although they do not come encapsulated in IPSec, i.e. those that enter through the ethernet0/0.20 interface.

If we now add another dynamic template assigned to another VRF, we have Example 3:

Example 3:

```
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.201 255.255.255.255
;
      exit
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp des md5
         template 1 life duration seconds 1d
         template 1 keepalive dpd
;
         template 2 default
         template 2 dynamic esp tdes sha1
         template 2 source-address 192.168.212.201
         template 2 encap transport
         template 2 vrf VRF1
;
         template 3 default
         template 3 dynamic esp des md5
         template 3 source-address 192.168.212.201
         template 3 encap transport
         template 3 vrf VRF2
;
         map-template 100 2
         map-template 100 3
```

In **Example 3** traffic with source 192.168.212.201 **which is transmitted through vrf VRF1** is protected by IPSec, in **tdes** and **sha1** modes, while traffic with the same source but **being transmitted through vrf VRF2** is also protected by IPSec but in **des** and **md5** modes. I.e.

- Packets with source 192.168.212.201 exiting the device through vrf **VRF1**, are **encapsulated** in an IPSec tunnel in **tdes** and **sha1** modes, i.e. they exit through the ethernet0/0.10 interface.
- Packets with source 192.168.212.201 exiting the device through vrf **VRF2**, are **encapsulated** in an IPSec tunnel in **des** and **md5** modes, i.e. they exit through the ethernet0/0.20 interface.
- Packets with destination 192.168.212.201 entering the device through vrf **VRF1**, must do so **encapsulated** in an IPSec tunnel in **tdes** and **sha1** modes i.e. those that enter through the ethernet0/0.10 interface. If they are not, the packets are dropped.
- Packets with destination 192.168.212.201 entering the device through vrf **VRF2**, must do so **encapsulated** in an IPSec tunnel in **des** and **md5** modes i.e. those that enter through the ethernet0/0.20 interface. If they are not, the packets are dropped.

"TEMPLATE [ID] NO VRF"

Does not assign the template to any VRF.

Example:

```
IPSec config>template 2 no vrf
```

"TEMPLATE [ID] FAULT-TOLERANT"

Configures the dynamic template to participate in the Fault Tolerant IPSec Recovery system. The IPSec sessions established using this template automatically pass to the pair forming the Fault Tolerant IPSec Recovery system when this device drops.

Example:

```
IPSec config>TEMPLATE 2 fault-tolerant
```

"TEMPLATE [ID] NO FAULT-TOLERANT"

Deactivates the participation of the dynamic template in the Fault Tolerant IPSec Recovery system.

Example:

```
IPSec config>TEMPLATE 2 no fault-tolerant
```

"TEMPLATE [ID] REPLACE-DESTINATION"

Configures the dynamic template to replace the destination of the packets encrypted through IPSec for the IPSec tunnel destination. This only makes sense when the tunnel in configured in transport mode, where the IP header in the IPSec packet is not encapsulated.

Ejemplo:

```
IPSec config>TEMPLATE 2 replace-destination
```

"TEMPLATE [ID] NO REPLACE-DESTINATION"

Deactivates the substitution of the encapsulated packets destination.

Ejemplo:

```
IPSec config>TEMPLATE 2 no replace-destination
```

- *ADVANCED Command*

In relation to the connection SAs created starting from the dynamic Templates, there is a command in the IPSec configuration's main menu that permits you to configure certain advanced characteristics. This command is **ADVANCED** and provides access to several subcommands:

Command	Operation
DPD	Service to ensure the maintenance of an SA connection.
KEEP-ALIVE	Service to ensure the maintenance of an SA connection.
PURGE-TIMEOUT	Configuration of SA's timeout.
RENEGOTIATION-TIME	Service to carry out SA re-negotiation.
NO	Establishes the default values for the IPSec configuration advanced parameters.

"ADVANCED DPD"

DPD (Dead Peer Detection) is a service which detects when communication with the other end of the Tunnel is lost. In order to use this, an ID vendor from the DPD is sent in phase 1 of any negotiation. This service consists of the exchange of notifications (an R-U-THERE petition and an R-U-THERE-

ACK response) in phase 2 in the Tunnel when there is no data reception during a certain period of time. This is configurable as idle time.
If this is enabled in an ISAKMP Template, the router will send phase 2 DPD petitions in the Tunnels created from the said Template and will also respond to these notifications. In cases where this is not enabled, the router will not send petitions but will respond to any received.

Command	Operation
ALWAYS-SEND	Always sends the keepalive once the idle time has timed out.
ANTI-REPLAY	Enables the DPD packets anti-replay capacity.
IDLE-PERIOD	Idle period before sending DPD packets.
INTERVAL	Interval between DPD keepalives.
PACKETS	Maximum number of DPD packets without confirmation.
NO	Disables an option or establishes the default values for a parameter.

"ADVANCED DPD ALWAYS SEND"
Indicates that DPD exchanges must be carried out when the idle time times out.

"ADVANCED DPD NO ALWAYS SEND"
Indicates that you must wait for data after the idle time has timed out before executing the exchange.

"ADVANCED DPD ANTI-REPLAY"
Enables the anti-replay capacity for DPD packets.

"ADVANCED DPD NO ANTI-REPLAY"
Disables the anti-replay capacity for DPD packets.

"ADVANCED DPD IDLE-PERIOD [SECONDS]"
Idle time before carrying out DPD exchanges i.e. time without receiving data in the Tunnel. Default value is 60 seconds. This can be re-established by executing the "ADVANCED DPD NO IDLE-PERIOD" command.

"ADVANCED DPD INTERVAL [SECONDS]"
Wait interval (in seconds) between DPD petition transmissions when a response has not been received. The default value is 5 seconds which can be re-established by executing the "ADVANCED DPD NO INTERVAL" command.

"ADVANCED DPD PACKETS [MAX_PKTS]"
Maximum number of DPD petitions without receiving a response. The default value (3) can be re-established by executing the "ADVANCED DPD NO PACKETS" command.

Example:

```
IPSec config>ADVANCED DPD ALWAYS-SEND
IPSec config>ADVANCED DPD IDLE-PERIOD 60
IPSec config>ADVANCED DPD INTERVAL 5
IPSec config>ADVANCED DPD PACKETS 3
IPSec config>ADVANCED DPD ANTI-REPLAY
Keep Alive modified
Do not forget to enable DPD in the template configuration
```

As the final message indicates, you must individually enable the DPD service in each ISAKMP Template if you want it with "TEMPLATE [ID] KEEPALIVE DPD".

"ADVANCED KEEP-ALIVE"

Keep Alive is a service that deals with ensuring that the other end maintains its SA open, observing the time that this remains without showing signs of life. On introducing this command, the user is asked to define two parameters:

Command	Operation
PACKETS	Maximum number of packets without receiving a response.
TIMEOUT	Wait period (in seconds) after the last packet.
NO	Establishes the default value of any of the previous parameters.

Example:

```
IPSec config>ADVANCED KEEP-ALIVE PACKETS 4
Keep Alive modified
Do not forget to enable Keep Alive in the template configuration.
IPSec config>ADVANCED KEEP-ALIVE TIMEOUT 10
Keep Alive modified
Do not forget to enable Keep Alive in the template configuration.
```

As the final message indicates, you must individually enable the Keep Alive service in each dynamic Template if you want it with "TEMPLATE [ID] KEEPALIVE KEEPALIVE".

"ADVANCED PURGE-TIMEOUT [SECONDS]"

Permits you to configure the SAs timeout. This is for example, the time taken in deleting a negotiation SA when, during negotiation with a Tunnel, the destination does not respond. The "ADVANCED NO PURGE-TIMEOUT" command re-establishes the default value for this parameter (15 seconds).

Example:

```
IPSec config>ADVANCED PURGE-TIMEOUT 15
```

"ADVANCED RENEGOTIATION-TIME"

Renegotiation time is a limit that is established in relation to the end time of a connection SA lifespan. If between this limit and the end of the SA there is traffic, the router will automatically renegotiate a new SA before the current SA lifespan times out. This avoids the situation of losing traffic due to SA timeout.

This limit is interpreted as a percentage and is applied to each individual lifetime (only in seconds) for each SA, without allowing it to ever drop below one minute.

The default value for this parameter is 10 (10%) which can be re-established through the "ADVANCED NO RENEGOTIATION-TIME" command.

Example:

```
IPSec config>ADVANCED RENEGOTIATION-TIME 20
Check-out time (%) - from SA's end-lifetime - to renegotiate : 20
```

The last line is one of confirmation and describes the following behavior: when an SA has 20% of its time left until it finalizes, the router begins to check if there is traffic up until the end-lifetime. If there is traffic then the router renegotiates a new SA when it has one minute left.

Other parameters which are configurable from the ADVANCED submenu from the IPSec configuration main menu are as follows:

Command	Operation
EXPONENTATION-DEVICE	Service ensuring the maintenance of an SA connection.
LQUEUE	Length of the cipher queue.
NO LQUEUE	Establishes the default value for the cipher queue length.
PKT-DEST-ISAKMP-DEST	Packet destination indicates the remote Tunnel address.
NAT-T-PORT	Port used in UDP encapsulation (NAT-T Translation).
NO NAT-T-PORT	Establishes the default value for the previous command.
NAT-LOCAL-ADDRESS	Local addresses for rules that will be changed.
NO NAT-LOCAL-ADDRESS	Deletes all the local addresses for the rules that will be changed.
ADDRESS-ASSIGNED-TO-IFC	Interfaces that the IP addresses obtained in the IKE negotiation take as destination.
NO ADDRESS-ASSIGNED-TO-IFC	Deletes the interfaces that the IP addresses obtained in the IKE negotiation take as destination.
RRI-FLASH	Forces the routes injected by RRI to be broadcast as quickly as possible through the dynamic routing algorithms.
NO RRI-FLASH	Deactivates the fast sending of routes injected by RRI.
CONNEVENT-PERIOD	Sets the periodic notification interval for the open connections.
NO CONNEVENT-PERIOD	Sets the default value for the above command.
NUMBER-OF-IPSEC-HEADERS	Establishes the maximum number of IPSec headers that a packet can have.
NO NUMBER-OF-IPSEC-HEADERS	Establishes the default value for the above command.

"ADVANCED EXPONENTATION-DEVICE"

This command provides access to two other commands: HARDWARE and SOFTWARE. These permit you to configure the way in which operations are carried out for cipher packets processing. If you select the HARDWARE option, cipher will be carried out at the HARDWARE level (cipher card). The SOFTWARE option implies that the operations will be carried out by using the software code.

Example:

```
IPSec config>ADVANCED EXPONENTIATION-DEVICE ?
HARDWARE      A hardware device will be used to carry out cipher operations
SOFTWARE      Software will be used to carry out cipher operations
IPSec config>ADVANCED EXPONENTIATION-DEVICE HARDWARE
```

"ADVANCED LQUEUE"

Configures the length of the cipher queue.

The use of the Fault Tolerant IPSec Recovery can provoke a high number of petitions in the encryption queue. If you are using this subsystem, you need to increase this value to, at least, the number of IPSec session to establish simultaneously or to the number of input buffers through the interface that establishes the sessions. The higher of the two.

Example:

```
IPSec config>ADVANCED LQUEUE
Size of the cypher queque:[50]? 25
IPSec config>
```

"ADVANCED NO LQUEUE"

Sets the cipher queue length to its default value: 50.

Example:

```
IPSec config>ADVANCED NO LQUEUE
IPSec config>
```

"PKT-DEST-ISAKMP-DEST"

Indicates that as IPSec Tunnel remote address, we will use the address the packets have as destination which induces the Tunnel to open.

Example:

```
IPSec config>ADVANCED PKT-DEST-ISAKMP-DEST
```

"NAT-T-PORT"

Configures the port used in UDP encapsulation (NAT-T Translation).

Default value is 4500.

Example:

```
IPSec config>ADVANCED nat-t-port 10000
IPSec config>
```

"NO NAT-T-PORT"

Sets the port used in UDP encapsulation (NAT-T Translation) to its default value.

Default value is 4500.

Example:

```
IPSec config>ADVANCED no nat-t-port 10000
IPSec config>
```

"NAT-LOCAL-ADDRESS"

Establishes the local address for the rules which will be changed with the ISAKMP configuration protocol (see the section on ISAKMP Configuration).

Example:

```
IPSec config>ADVANCED nat-local-address ppp1
IPSec config>
```

"NO NAT-LOCAL-ADDRESS"

Deletes the local address for a rule which will be changed with the ISAKMP configuration protocol (see the section on ISAKMP Configuration).

Example:

```
IPSec config>ADVANCED no nat-local-address ppp1
IPSec config>
```

"ADDRESS-ASSIGNED-TO-IFC"

Establishes the interfaces that the IP addresses, obtained through the ISAKMP configuration protocol, take as destination (please see the section on Configuring ISAKMP).

Example:

```
IPSec config>ADVANCED address-assigned-to-ifc loopback1
IPSec config>
```

"NO ADDRESS-ASSIGNED-TO-IFC"

Deletes the interfaces that the IP addresses, obtained through the ISAKMP configuration protocol, take as destination (please see the section on Configuring ISAKMP).

Example:

```
IPSec config>ADVANCED no address-assigned-to-ifc
IPSec config>
```

"RRI-FLASH"

When this command is enabled, routes injected by RRI are broadcast through the dynamic routing algorithms as soon (provided this is possible) as they enter the device's routing table.

Example:

```
IPSec config>advanced rri-flash
IPSec config>
```

"NO RRI-FLASH"

Disables immediate broadcasting of the RRI routes. These will be sent by the habitual timers of the used dynamic routing algorithm (device default behavior).

Example:

```
IPSec config>advanced no rri-flash
IPSec config>
```

"CONNEVENT-PERIOD"

A temporary value must be entered to enable the periodic CONNEVENT event execution which uses this value as a period. A conn event is produced for each established connection with a maximum generation rate of 40 to 50 events per second; the aim is to concentrate these at the beginning of each interval even though the process is extended if the number of opened tunnels is high. This period must be sufficiently long so all the open connections can be notified, contrariwise the monitoring console, through the **list statistics**, displays the number of connections that could not be notified during the previous period.

Example:

```
IPSec config>advanced connevent-period 5m
IPSec config>
```

"NO CONNEVENT-PERIOD"

Disables the periodic notification for the open connections, which uses the CONNEVENT event, setting the period value to its default value: 0 seconds.

Example:

```
IPSec config>advanced no connevent-period
IPSec config>
```

"NUMBER-OF-IPSEC-HEADERS"

Establishes the maximum number of IPSec headers that a packet can have, i.e. the number of successive encapsulated IPSecs that can be carried out. The default value for this command is 1.

Example:

```
IPSec config>advanced number-of-ipsec-headers 2
IPSec config>
```

"NO NUMBER-OF-IPSEC-HEADERS"

Sets the number of IPSec headers that a packet can have to its default value: 1.

Example:

```
IPSec config>advanced no number-of-ipsec-headers
IPSec config>
```

- *KEY PRESHARED Command*

This step finalizes the configuration of the ISAKMP and dynamic Templates required in order to carry out IPSec IKE. However there is a further parameter left to introduce to make these operational. This deals with the Pre-Shared Key that both security routers must have in order to mutually authenticate. This key is introduced from the main IPSec menu:

"KEY PRESHARED IP/HOSTNAME [ADDRESS/NAME] CIPHERED/PLAIN [KEY]"

This permits you to introduce the Pre-shared key associated to the remote IP address or device name depending how the Tunnel was configured when the "TEMPLATE IKE IDTYPE" command was used for the ISAKMP Templates.

Please note that this key however is not associated to a Template but to a remote IP address or host. Consequently, this does not require an [ID] identifier as in the rest of the commands.

The Pre-shared key can be introduced in plain (subcommand PLAIN) or ciphered (subcommand CIPHERED). If this is manually configured from the console, you normally introduce the key in plain. If you use the configuration saved in text mode however (precedent from the "SHOW CONFIG" command) the key will be ciphered. In cases where it is plain, the key can have a length between 1 and 32 bytes. This can be introduced in hexadecimal, beginning with 0x or in ASCII. Please note that if you introduce this in hexadecimal, you must introduce double the characters (between 0-9 and A-F). If the key is ciphered then it is always displayed in hexadecimal.

Example 1:

```
IPSec config>KEY PRESHARED IP 192.100.1.1 plain 1234567890
IPSec config>KEY PRESHARED HOSTNAME Router2 plain 1234567890teldat
IPSec config>KEY PRESHARED IP 192.100.1.1 plain 0x1234567890abcdef
```

The Pre-shared key admits networks with mask 0, 8, 16 and 24 bits in IP addresses.

Example 2:

```
IPSec config>KEY PRESHARED IP 192.100.1.0 plain 1234567890
```

This key is assigned to all the network 192.100.1.0 255.255.255.0

The Pre-shared key admits the wildcard character (asterisk) at the end of the hostname.

Example 3:

```
IPSec config>KEY PRESHARED HOSTNAME Router* plain 1234567890teldat
```

This key is assigned to Router1, RouterTeldat, Router, Router_234...

In cases where intersections exist, the most restrictive is always taken.

Example 4:

```
IPSec config>KEY PRESHARED HOSTNAME Router* plain 1234567890teldat
IPSec config>KEY PRESHARED HOSTNAME Router plain 1111111
```

If the hostname is Router, key 1111111 will be used.

```
IPSec config>KEY PRESHARED IP 192.100.1.0 plain 1234567890
IPSec config>KEY PRESHARED IP 192.100.1.163 plain aaaa
```

If the IP is 192.100.1.163 key aaaa will be used.

You can view the configured Pre-shared keys by using the "LIST KEY PRESHARED" command. The keys are not printed as such in the console but it is possible find out what IP addresses or hostnames have a Pre-shared key associated:

```
IPSec config>LIST KEY PRESHARED
5 key entries
   192.100.1.1 ************
   Router2 ************
   192.100.1.0 ************
   Router* ************
   Router ************
```

If you wish to delete a key associated to an IP address or hostname, simple execute the "NO KEY PRESHARED IP/HOSTNAME [ADDRESS/NAME]" command:

```
IPSec config>NO KEY PRESHARED IP 192.100.1.0
```

c) Creating the SPD

Finally, once the Access Control List and the Templates have been defined you have to create a policy database or SPD. Each input from this database is made up of an element from the Access Control List and an associated Template. The association is known as **mapping** and the command and its use for mapping the entries is shown as follows:

Command	Operation
ASSIGN-ACCESS-LIST	Assigns an access control list to the IPSec protocol
ASSOCIATE-KEY	Associates a key with an access control list.
ASSOCIATE-DEST-MASK	Associates a destination mask to an access control list.
MAP-TEMPLATE	Associates access control list elements with Templates.

"ASSIGN-ACCESS-LIST [LCA entry ID]"

Assigns a generic and extended access control list to the IPSec protocol.

Example:

```
IPSec config>ASSIGN-ACCESS-LIST 100
```

"ASSOCIATE-KEY IP/HOSTNAME [ACCESS_LIST] [ADDRESS/NAME KEY]"

One of the parameters negotiated during the opening of an IPSec Tunnel is the access control i.e. the *Tunnel clients*. In principal, the knowledge of a Pre-shared key permits the remote device to open a Tunnel to the local device with client independence. However occasionally this is not convenient and you need to provide certain controls for the devices which recognize one key and other controls to those that recognize a different key.

In the **Example 1** shown below, the following statements can be made:

- Only devices which know the key associated to the hostname *teldat_router* will be able to open a Tunnel accessing the whole of the 192.60.64.0/24 network.

- Devices which only know the key associated to *router* **will not** be able to open a Tunnel accessing the whole of the 192.60.64.0/24 network.
- As the access control list 101 does not have a key associated, devices which know the key associated to *router* and that associated to *teldat_router* will be able to open a Tunnel accessing host 192.60.64.1

Example 1:

```
Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=192.60.64.1/32  DES=0.0.0.0/16  Conn:0


Extended Access List 100, assigned to IPSec

10    PERMIT  SRC=192.60.64.0/24  DES=0.0.0.0/16  Conn:0
IPSec config> LIST KEY PRESHARED
2 key entries
   teldat_router ************
   router ************
IPSec config> ASSOCIATE-KEY HOSTNAME 100 teldat_router
```

In the **Example 2** shown below, the following statements can be made:

- Only those devices that have a key associated to an IP address, which begins with 10, can open a Tunnel for production. In this case a production Tunnel can only be opened for a device that has the key for IP address 10.127.0.28. This behavior is controlled by the *associate-key ip 100 10.0.0.0* command.
- All devices that have the key associated to an IP address, which begins with 10 or the generic key, introduced by the *key preshared ip 0.0.0.0* command, can open a Management Tunnel with the *10.127.1.57* management device.

Example2:

```
feature access-lists
; -- Access Lists user configuration --
   access-list 100
      description "Control de Acceso para Produccion"
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.1.0 255.255.255.0
      entry 1 destination address 10.127.0.0 255.255.0.0
;
   exit
;
   access-list 101
      description "Control de Acceso para Gestion"
;
      entry 1 default
      entry 1 permit
      entry 1 source address 10.127.1.57 255.255.255.255
      entry 1 destination address 10.127.0.0 255.255.0.0
;
   exit
;
exit
p ip
; (...)
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
```

```
      assign-access-list 100
;
; (...)
      associate-key ip 100 10.0.0.0
      key preshared ip 0.0.0.0 ciphered 0x37349246263B0066
      key preshared ip 10.127.0.28 ciphered 0x7CC9756395EFB97F
   exit
```

"ASSOCIATE-DEST-MASK [ACCESS_LIST] [MASK]"

One of the parameters negotiated when opening an IPSec Tunnel is the access control, i.e. *Tunnel clients*. In a star configuration for example, the tunnel clients usually have different network subnets assigned to them. E.g. supposing we have a network where each remote end has IP addresses with C mask assigned from the whole 192.168.0.0/16 network. I.e. we'd have the *A* end with the 192.168.1.0/24 group of addresses, the *B* end with 192.168.2.0/24 group of addresses, and so on.

In this case, in the remote central, you simply need to configure a single access control list which has 192.168.0.0/16 as its destination address. The problem with this configuration is that this does not limit a remote end from opening a tunnel with a wider addressing than wanted, i.e. 255 addresses. With the associate-dest-mask parameter equal to 255.255.255.0 you can apply the required limit. I.e. by using this parameter you can achieve a very simple configuration in the central side which is protected from negotiations that try and propose a wider range of addresses than permitted, without having to configure a different access list for each of them.

In the below example, we can make the following assertions:

- The remote devices can only open the Tunnel with a subset of the 192.168.0.0/16 network and that this subset has a mask with more bits set to 1 than 255.255.255.0. This latter is that configured in the associate-dest-mask.
- A remote device that tries to open a Tunnel with a mask with fewer bits set to 1 than that configured in the associate-dest-mask will have its proposal rejected with an *invalid-id* message.

Example:

```
Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=192.60.64.1/32  DES=192.168.0.0/16  Conn:0


IPSec config> ASSOCIATE-DEST-MASK 101 255.255.255.0
```

"MAP [LCA entry ID] [Template ID]"

This command associates an element from the access control list with a Template, creating an SPD element.

Example:

```
IPSec config>MAP-TEMPLATE 100 4
```

When mapping is carried out, you can sometimes see some automatic entries not introduced by the user in the list of entries in the access control list found in the IPSec monitoring menu. These are distinguished by the words DYNAMIC ENTRY. These automatic entries are necessary so that both ends of the Tunnel can communicate control packets.

```
Extended Access List 101, assigned to IPSec

 ACCESS LIST ENTRIES
0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=17  SPORT=500
        DYNAMIC ENTRY
        Hits: 0

0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=17  SPORT=4500
        DYNAMIC ENTRY
        Hits: 0

0     DENY    SRC=0.0.0.0/0  DES=192.60.64.1/32  Conn:0
        PROT=50-51
        DYNAMIC ENTRY
        Hits: 0

0     PERMIT  SRC=192.60.64.2/32  DES=192.60.64.1/32  Conn:0
        DYNAMIC ENTRY
        Hits: 0

1     PERMIT  SRC=0.0.0.6/32  DES=192.60.64.1/32  Conn:0
        Hits: 0
```

Mapping is the last step required in order to configure the complete IPSec security service. Before considering the configuration completed you can check what has been carried out, modifying any errors and even determine which events you wish to view in the trace monitoring:

Command	**Operation**
LIST ALL	Displays all the configuration.
SHOW CONFIG	Displays the configuration commands.
NO ASSIGN-ACCESS-LIST	Eliminates the assignation of an access control list to the IPSec protocol.
NO ASSOCIATE-KEY	Eliminates the association of a key to an access control list.
NO ASSOCIATE-DEST-MASK	Eliminates the association of a destination mask to an access control list.
NO MAP-TEMPLATE	Eliminates the association between LCA elements and Templates.
EVENT	Enables certain Events.
LIST ENABLED-EVENTS	Displays the filter configured for events monitoring (should there be one).
QOS-PRE-CLASSIFY	Classification of packets in their respective BRS classes.
NO QOS-PRE-CLASSIFY	Disables classification of packets in their respective BRS classes.

"LIST ALL"

Displays all of the configuration policies the SPD contains, i.e. the LCA elements and the list of Templates.

Example:

```
IPSec config>LIST ALL
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 1


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=0.0.0.6/32  DES=192.60.64.1/32  Conn:0

TEMPLATES
1 dynamic ESP-3DES ESP-MD5  SRC=0.0.0.6 DES=192.60.64.1
   LifeTime:0h3m0s 100000 kbytes
   PFS disabled

2 dynamic ESP-DES ESP-SHA1  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

3 dynamic AH-MD5  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

4 dynamic AH-SHA1  SRC=192.24.51.75 DES=192.24.51.74
   LifeTime:0h50m0s 100000 kbytes
   PFS disabled

20 isakmp 3DES MD5  DES=192.60.64.1
   LifeTime:0h4m0s
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1

4 key entries
   172.24.51.57 ************
   192.24.51.74 ************
   192.24.78.75 ************
   192.60.64.1 ************
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Use software exponentiation
IPSec config>
```

"SHOW CONFIG"

Displays the configuration commands. Please note that the values of the fields that coincide with the default value are not shown. In the example shown below, the result of the *SHOW CONFIG* command is displayed with the configuration of the example presented with the *LIST ALL* command.

Example:

```
IPSec config>SHOW CONFIG
; Showing System Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

      enable
      assign-access-list 101
;
      template 1 create
      template 1 dynamic esp tdes md5
      template 1 source-address 0.0.0.6
      template 1 destination-address 192.60.64.1
      template 1 life type both
      template 1 life duration seconds 180
      template 1 life duration kbytes 100000
;
      template 2 create
      template 2 dynamic esp des sha1
      template 2 source-address 192.24.51.75
      template 2 destination-address 192.24.51.74
      template 2 life type both
      template 2 life duration seconds 3000
      template 2 life duration kbytes 100000
;
      template 3 create
      template 3 dynamic ah md5
      template 3 source-address 192.24.51.75
      template 3 destination-address 192.24.51.74
      template 3 life type both
      template 3 life duration seconds 3000
      template 3 life duration kbytes 100000
;
      template 4 create
      template 4 dynamic ah sha1
      template 4 source-address 192.24.51.75
      template 4 destination-address 192.24.51.74
      template 4 life type both
      template 4 life duration seconds 3000
      template 4 life duration kbytes 100000
;
      template 20 create
      template 20 isakmp tdes md5
      template 20 destination-address 192.60.64.1
      template 20 life duration seconds 240
      template 20 ike ca THAWTECA.CER
      template 20 ike mode aggressive
      template 20 ike idtype fqdn
;
      map-template 101 1
      key preshared ip 172.24.51.57 holas
      key preshared ip 192.24.51.74 ciphered 0xF85C0CB62556C562120794C28EB9334
      key preshared ip 192.24.78.75 ciphered 0xF85C0CB62556C562120794C28EB9334
      key preshared ip 192.60.64.1 ciphered 0xF85C0CB62556C562120794C28EB9334
IPSec config>
```

"NO ASSIGN-ACCESS-LIST [LCA entry ID]"

Eliminates the assignation of an access control list to the IPSec protocol.

Example:

```
IPSec config>NO ASSIGN-ACCESS-LIST 100
```

"NO ASSOCIATE-KEY [LCA entry ID]"

Eliminates the association of a key to an access control list.

Example:

```
IPSec config>NO ASSOCIATE-KEY 100
```

"NO ASSOCIATE-DEST-MASK [LCA input ID]"

Eliminates the association of a destination mask to an access control list.

Example:

```
IPSec config>NO ASSOCIATE-DEST-MASK 100
```

"NO MAP-TEMPLATE [LCA entry ID] [Template ID]"

Eliminates the association or mapping of an LCA element with the Template.

Example:

```
IPSec config>NO MAP-TEMPLATE 10 4
```

Even though you disable the mapping, the automatic entry that was generated remains. I.e. this has to be deleted if you do not require it.

"EVENT ALL"

This permits you to view all the events. The said events have to be enabled in the events monitoring process (P 3) and can be viewed in P 2.

Example:

```
IPSec config>EVENT ALL
```

"EVENT ADDRESS-FILTER [IP ADD][MASK]"

Once enabled, this only permits you to view those events with a source address or destination that is included within the range defined by [IP ADD][MASK]. Please see the **list negotiation filter** monitoring command.

Example:

```
IPSec config>EVENT ADD 192.100.1.2 255.255.255.255
```

"LIST ENABLED-EVENTS"

Displays the filter configured for event monitoring (should there be one).

Example:

```
IPSec config>LIST ENABLED-EVENTS

 Address/Subnet enabled : 192.100.1.2 with MASK : 255.255.255.255
```

"QOS- PRE-CLASSIFY"

Permits you to enable the classification of packets in their respective BRS classes before being ciphered.

```
IPSec config>QOS-PRE-CLASSIFY
IPSec config>
```

To disable this option, simple execute the "NO QOS-PRE-CLASSIFY" command:

```
IPSec config>NO QOS-PRE-CLASSIFY
IPSec config>
```

If this mode is enables, the packets will be classified before being ciphered therefore distinct traffic classes can be prioritized within the same IPSec Tunnel. Classification only operates in those access controls which are associated to an IP rule, contrariwise you will not know which interface the packets are going to exit through before being ciphered and therefore the BRS associated to this interface cannot be applied. If this mode is disabled, all traffic coming from the IPSec Tunnel will be classified in the same BRS class, as the header that will be analyzed is the IPSec Tunnel header.

3.3. ISAKMP Configuration Mode

There is a method that permits you to configure the phase II parameters which are negotiated after finishing phase I. Through this method, you can reliably define the characteristics that the IPSec session negotiated in phase II will have in order to exchange data. When creating this documentation, the details of the properties and operation mode for this configuration mode can be found in the draft: *The ISAKMP Configuration Mode.*

This method is usually used in star configurations, where the central node assigns the addresses that each of the ends connecting to the VPN are going to have during the session, which will be the name servers, if using PFS or the port over which NAT-T will be carried out is going to be used.

You will find the following parameters within the TEMPLATE menu which permit you to configure this method:

Command	Operation
IKE METHOD	Incorporates the "xauth-init-preshared" option.
CONFIG	Permits you to define if the device will initiate the configuration method, wait for a proposal or if it will behave as indicated by the IKE method used.

"IKE METHOD XAUTH-INIT-PRESHARED"

Through this command you add a new functionality to the previously described IKE METED command. This functionality is known as *Extended Authentication Preshared* described in the *Extended Authentication within ISAKMP/Oakle* draft when creating this documentation. On activating this parameter this indicates if you wish to carry out a pre-shared authentication where you wish to execute an *ISAKMP Configuration* process, where the initiator device must authenticate with a remote server. This latter can assign, among other things, the IP address within the VPN.

Example:

```
IPSec config>TEMPLATE 4 ike method xauth-init-preshared
```

"CONFIG INITIATOR"

This command indicates that the device will initiate the configuration method, carrying out the initial proposals and requesting the necessary parameters.

Example:

```
IPSec config>TEMPLATE 4 config initiator
```

"CONFIG RESPONDER"

This command indicates that the device will wait for the remote end to initiate the configuration method.

Example:

```
IPSec config>TEMPLATE 4 config responder
```

"CONFIG NONE"

This command indicates that the device will behave as the initiator or responder depending on that indicated by the used IKE method.

Example:

```
IPSec config>TEMPLATE 4 config none
```

- *EXTENDED AUTHENTICATION*

Extended Authentication consists of authentication with a server device which assigns the parameters needed to establish a connection. This authentication is typically executed through a user and a password.

The commands described below permit you to associate a user and a password and an IP address or name.

Command	Operation
XAUTH-IP	Associates a user to an IP address.
XAUTH-HOSTNAME	Associates a user to a name.
XAUTH-USER	Specifies a user's properties.

"XAUTH-IP [IP address] USER [user name]"

"XAUTH-IP [IP address] PASSWORD [password]"

"XAUTH-IP [IP address] LOCAL-LAN-ACCESS [network]"

"XAUTH-IP [IP address] NO REQUEST IP-ADDRESS"

Through these commands you can define a user and a password that will be associated to the IP address which is introduced as a parameter.

In cases where this is the initiator this IP address will indicate the address with which the remote end identified itself. Generally, when phase I and the authentication have concluded, the initiator requests an IP address, but it won't do this if the *no request ip-address* is configured. Consequently, once authentication has been achieved, this immediately moves to phase II, using, as *clients*, those configured in the associated access list.

In cases where this is the responder this IP address will indicate the address which will be assigned to the end initiator in the ISAKMP Configuration method negotiation (when you wish to get this address from a pool, use the XAUTH-USER command instead of the former). In addition, the LOCAL-LAN-ACCESS command specifies the initiator's local network which remains outside of the IPSec tunnel. I.e. this is a network not protected by IPSec, on the initiator side. This command is usually used to provide the initiator with restricted access to its local network, which is normally a LAN, without passing through the IPSec policies.

Example:

```
IPSec config>xauth-ip 1.1.1.1 user router1
IPSec config>xauth-ip 1.1.1.1 password plain mykey
IPSec config>xauth-ip 1.1.1.1 local-lan-access 192.168.1.0 255.255.255.0
```

"XAUTH-HOSTNAME [hostname] USER [user name]"

"XAUTH-HOSTNAME [hostname] PASSWORD [password]"

Through these two commands you can define the user and password that will be associated to the name introduced as a parameter.

This name indicates the hostname through which the remote end identifies itself.

Example:

```
IPSec config>xauth-hostname remoterouter user router1
IPSec config>xauth-hostname remoterouter password plain mykey
```

"XAUTH-USER [user] POOL [pool name]"

"XAUTH-USER [user] PASSWORD [password]"

"XAUTH-USER [user] LOCAL-LAN-ACCESS [network]"

"XAUTH-IP [user] NO REQUEST IP-ADDRESS"

Through these commands, you can associate a user with a pool, a password and the local network which is outside of the tunnel.

In cases where this is the responder, when the user identifies through *user* and subsequently requests an IP address, this is obtained from the pool configured as *pool name* which must have been previously defined in the IP configuration. Please note that if you wish this address to always remain the same, use the XAUTH-IP command instead of the former. Additionally the LOCAL-LAN-ACCESS command, in the same way as the XAUTH-IP command, specifies the initiator's local network which is outside the IPSec tunnel.

Example:

```
IP config>pool remotevpn 172.24.100.80 172.24.100.95
IPSec config>xauth-user myuser default
IPSec config>xauth-user myuser pool remotevpn
IPSec config>xauth-user myuser password plain mykey
IPSec config>xauth-user myuser local-lan-access 192.168.1.0 255.255.255.0
```

In the above example, the dialogue follows this sequence:

1. When phase I concludes, the responder requests the user identification and password.
2. The initiator delivers its user and password.
3. The responder checks that they are correct and returns a positive response.
4. The initiator then requests an IP address, DNS server, local access network, NAT port plus other parameters.
5. The responder obtains an IP address from the remotevpn pool and delivers this to the initiator. It also delivers the local access network, the DNS server, NAT port and the rest of the requested parameters.
6. The initiator then begins phase II using the assigned IP.

Please note that in cases where the initiator has the *no request ip-address* configured steps 4 and 5 do not appear. Consequently, once authentication has been achieved, this immediately moves to phase II, using, as clients, those configured in the associated access list.

- *Configuration example: Teldat Router Server for VPN Clients*

Description

Let's assume we have a Teldat Router that closes VPN Clients connections tunnels. This router has an Ethernet interface and an ADSL interface.

The Ethernet network is 172.24.0.0/16 and the address is172.24.78.130.

The ADSL address is 80.1.1.123.

```
network ethernet0/0
; -- Internet protocol user configuration --
   ip address 172.24.78.130 255.255.0.0
   exit
network atm0/0.1
   ip address 80.1.1.123 255.255.255.255
```

The device has subnet 172.24.6.80 255.255.255.240 available for users which remotely connect. This subnet is configured in the *remotevpn* pool.

```
feature access-lists
; -- Access Lists user configuration --
   access-list 101
;
      entry 1 default
      entry 1 permit
      entry 1 destination address 172.24.6.80 255.255.255.240
;
      exit
exit
protocol ip
      route 172.24.6.80 255.255.255.240 80.1.1.123
      pool remotevpn 172.24.6.80 172.24.6.95
      proxy-arp
; -- Proxy ARP Configuration --
         enable
      exit
exit
network ethernet0/0
      ip proxy-arp ip-address 172.24.78.130 enable
```

The configuration for user *daisy* would be:

```
protocol ip
ipsec
      xauth-user daisy default
      xauth-user daisy pool remotevpn
      xauth-user daisy password plain adios
```

The group is called *migrupo* and the key would be *hola*.

```
protocol ip
ipsec
key preshared hostname migrupo plain hola
```

Complete Configuration

```
; Showing System Configuration ...
; ATLAS Router 2 8 Version 10.1.X
log-command-errors
no configuration
add device atm-subinterface atm0/0 1
network ethernet0/0
; -- Internet protocol user configuration --
   ip address 172.24.78.130 255.255.0.0
   ip proxy-arp ip-address 172.24.78.130 enable
   exit
network atm0/0.1
   ip address 80.1.1.123 255.255.255.255
         exit
feature access-lists
; -- Access Lists user configuration --
   access-list 101
;
      entry 1 default
      entry 1 permit
      entry 1 destination address 172.24.6.80 255.255.255.240
;
   exit
;
```

```
exit
;
;
protocol ip
; -- Internet protocol user configuration --
;
   route 0.0.0.0 0.0.0.0 80.1.1.123 1
   route 172.24.6.80 255.255.255.240 80.1.1.123
   pool remotevpn 172.24.6.80 172.24.6.95
   proxy-arp
; -- Proxy ARP Configuration --
   enable
   exit
;
   rule 1 default
   rule 1 local-ip 80.1.1.123
   rule 1 napt translation
   rule 1 napt firewall
   rule 1 napt timeout 30
;
   classless
;
;
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes md5
      template 1 source-address 80.1.1.123
      template 1 life duration seconds 86400
      template 1 ike mode aggressive
      template 1 ike method xauth-init-preshared
      template 1 ike group two
      template 1 keepalive dpd
;
      template 2 default
      template 2 dynamic esp tdes md5
      template 2 source-address 80.1.1.123
;
      map-template 101 2
      key preshared hostname migrupo plain hola
      advanced purge-timeout 30
;
;
      xauth-user pepito default
      xauth-user pepito pool remotevpn
      xauth-user pepito password plain adios
;
;
   exit
;
exit
;
feature dns
; -- DNS resolver user configuration --
   server 172.24.0.7
exit
;
dump-command-errors
end
```

- *Configuration for a VPN Client if this is a Teldat Router and not requesting IP address assignment*

This section shows the configuration that a router which is acting as the server client in the previous example would have, and additionally is not requesting an IP address.

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS150 Router 7 96 Version 10.7.8-Alfa

   log-command-errors
   no configuration
   set hostname migrupo
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.6.84 255.255.255.252
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.248.28 255.255.255.0
      ip address 172.24.6.85 255.255.255.252 secondary
;
;
;
;
;
   exit
;
;
;
;
;
;
   event
; -- ELS Config --
      enable trace subsystem IKE ALL
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.248.98
;
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes md5
         template 1 destination-address 80.1.1.123
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v2-n
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyid
         template 1 ike group two
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.248.28
         template 2 destination-address 80.1.1.123
;
         map-template 100 2
         key preshared ip 80.1.1.123 plain hola
;
;
```

```
         xauth-ip 80.1.1.123 default
         xauth-ip 80.1.1.123 user pepito
         xauth-ip 80.1.1.123 password plain adios
         xauth-ip 80.1.1.123 no request ip-address
;
      exit
;
   exit
;
;
;
   dump-command-errors
   end
```

- *ASSIGNED IP ADDRESS DESTINATION*

During the ISAKMP Configuration protocol, the device acting as client can receive an IP address. This assigned IP address is used in two different modes:

- As a NAT address in the NAPT rules.
- As an interface address.

As a NAT address in the NAPT rules

The assigned IP address becomes the NAT address used in the NAPT rules, whose local address or interface coincides with that configured in the nat-local-address command.

In this operating mode, you need to know which network the IP address, assigned by the service to configure the access list, pertains to. Additionally you need to use a fictitious address for this network in the NAPT rule so ISAKMP negotiation is triggered due to traffic.

Below we have a configuration example where we know that the IP address the server assigns pertains to network 172.24.0.0/16 and an address from this network, 172.24.78.1, will be used as the IP address for the NAPT rule. You need to bear in mind that you can set any IP address from this network in the NAPT rule as it's never going to be used. This address is changed during the ISAKMP Configuration protocol.

```
   log-command-errors
   no configuration
   set hostname teldat
   add device ppp 1
   set data-link at cellular0/0
   set data-link at cellular0/1
   set data-link at cellular1/0
   set data-link at cellular1/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.24.0.0 255.255.0.0
      exit
   exit
;
   global-profiles dial
; -- Dial Profiles Configuration --
      profile HSPA default
```

```
      profile HSPA inout
      profile HSPA 3gpp-apn movistar.es
;
      profile MOVISTAR default
      profile MOVISTAR dialout
      profile MOVISTAR idle-time 300
   exit
;
   global-profiles ppp
; -- PPP Profiles Configuration --
      lcp-options cellular1/1 default
      lcp-options cellular1/1 acfc
      lcp-options cellular1/1 pfc
      lcp-options cellular1/1 accm 0
   exit
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.6.3.186 255.255.255.0
;
   exit
;
   network cellular1/0
; -- Interface AT. Configuration  --
      pin plain 1111
      sim-select internal-socket-2
;
      network mode automatic
      network domain cs+ps
   exit
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
;
      ppp
; -- PPP Configuration --
         authentication sent-user MOVISTAR password keykey
         ipcp local address assigned
         no ipcp peer-route
         lcp echo-req off
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface cellular1/1 link
         base-interface cellular1/1 profile HSPA
;
      exit
;
   exit
;
   network loopback1
; -- Loopback interface configuration --
      ip address unnumbered
   exit
;
   event
; -- ELS Config --
      enable trace subsystem IKE ALL
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 ppp1
;
      rule 1 local-ip ppp1 remote-ip any
      rule 1 napt translation
      rule 1 napt ip 172.24.78.1
```

```
;
      Classless
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes md5
         template 1 destination-address 80.36.189.231
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v3
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyid
         template 1 ike group two
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address ppp1
         template 2 destination-address 80.36.189.231
;
         map-template 100 2
         key preshared ip 80.36.189.231 plain key1
         advanced nat-local-address ppp1

;
         xauth-ip 80.36.189.231 default
         xauth-ip 80.36.189.231 user anonymous
         xauth-ip 80.36.189.231 password plain pppp
;
      exit
;
   exit
   dump-command-errors
   end
```

Notes:

- All the traffic is protected by IPSec, with source addressing being assigned by the server.
- In this configuration, you cannot use the NAPT rules' firewalling command.
- The device doesn't respond to the assigned IP address. I.e. it cannot be managed (telnet, ftp, snmp, etc.), from the server network.
- You need to know the network the IP address assigned by the server, belongs to a priori.

If one of these conditions isn't fulfilled, you need to configure the address assignation mode to an interface. This is explained below.

As an interface address

The assigned IP address becomes the interface address configured through the "advanced address-assigned-to-ifc" command. Normally this interface is loopback.

A typical configuration example is as follows:

```
   log-command-errors
   no configuration
   set hostname teldat
   add device ppp 1
```

```
   add device loopback 1
   set data-link at cellular0/0
   set data-link at cellular0/1
   set data-link at cellular1/0
   set data-link at cellular1/1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address interface loopback1
      exit
;
   exit
;
   global-profiles dial
; -- Dial Profiles Configuration --
      profile HSPA default
      profile HSPA inout
      profile HSPA 3gpp-apn movistar.es
;
      profile MOVISTAR default
      profile MOVISTAR dialout
      profile MOVISTAR idle-time 300
;
   exit
;
   global-profiles ppp
; -- PPP Profiles Configuration --
      lcp-options cellular1/1 default
      lcp-options cellular1/1 acfc
      lcp-options cellular1/1 pfc
      lcp-options cellular1/1 accm 0
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.6.2.186 255.255.255.0
;
   exit
;
   network cellular1/0
; -- Interface AT. Configuration  --
      pin plain 1111
      sim-select internal-socket-2
;
      network mode automatic
      network domain cs+ps
   exit
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address unnumbered
      ppp
; -- PPP Configuration --
         authentication sent-user MOVISTAR password keykey
         ipcp local address assigned
         no ipcp peer-route
         lcp echo-req off
      exit
;
      base-interface
; -- Base Interface Configuration --
         base-interface cellular1/1 link
         base-interface cellular1/1 profile HSPA
;
```

```
      exit
   exit
;
   network loopback1
; -- Loopback interface configuration --
      ip address unnumbered
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      route 80.36.189.231 255.255.255.255 ppp1
      route 0.0.0.0 0.0.0.0 loopback1
;
      rule 2 local-ip loopback1 remote-ip any
      rule 2 napt translation
      rule 2 napt firewall
;
      rule 1 local-ip ppp1 remote-ip any
      rule 1 napt translation
      rule 1 napt firewall
;
      classless
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes md5
         template 1 destination-address 80.36.189.231
         template 1 ike mode aggressive
         template 1 ike natt-version draft-v3
         template 1 config responder
         template 1 ike method xauth-init-preshared
         template 1 ike idtype keyed
         template 1 ike group two
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address ppp1
         template 2 destination-address 80.36.189.231
;
         map-template 100 2
         key preshared ip 80.36.189.231 plain key1
         advanced address-assigned-to-ifc loopback1

;
         xauth-ip 80.36.189.231 default
         xauth-ip 80.36.189.231 user anonymous
         xauth-ip 80.36.189.231 password plain pppp
;
      exit
;
   exit
   dump-command-errors
   end
```

The series of events are as follows:

- The device starts up.
- NAPT is executed if there is traffic as it has a default route for the loopback interface, as indicated in rule 2, and the loopback1 address is set as the source IP address.
- The traffic consequently coincides with entry 1 on the access control list 100 and begins negotiation with address 80.36.189.231; ppp1 has this address as a route and sets the ppp1 IP address as source.

- The ISAKMP Configuration session is established and the assigned IP address becomes the loopback1 address.
- The SAs are established and all the traffic goes through these given that any traffic has the loopback1 as the route, applying rule 2, setting the new loopback1 address as source which coincides with entry 1 in the access control list 100.

Notes:

- All the traffic is protected by IPSec, with the source address being assigned by the server.
- You can access the device from the server by accessing the assigned IP.
- If you don't need firewalling in ppp1, rule 1 isn't required.

To ensure that only traffic with destination 172.24.0.0/16 passes through IPSec, the rest goes through ppp1, you need to change the routes in the following way:

```
route 172.24.0.0 255.255.0.0 loopback1
route 0.0.0.0 0.0.0.0 ppp1
```

I.e. traffic that you want protected is defined in the routes. All traffic whose route destination is loopback1 is protected by IPSec.

3.4. GDOI GROUP [id]

The GDOI GROUP <id> command permits you to configure a GDOI server where the clients register to download the policies and the encryption keys. On entering the gdoi group <id> command you can access the GDOI server configuration menu which contains the following commands:

Command	Function
ADDRESS IPV4 <ip>	Configures the server's local IP.
IDENTITY NUMBER <id>	Configures the group identifier.
REKEY ADDRESS IPV4 <ip>	Configures the multicast IP for the rekey messages.
REKEY LIFETIME SECONDS <sec>	Configures the lifetime for the rekey SAs.
REKEY ALGORITHM <alg>	Configures the encryption algorithm used in the rekey message, with the option of choosing between des, 3des or aes.
REKEY AUTHENTICATION RSA	Configures the RSA key used for authentication in the rekey messages.
REKEY RETRANSMIT <s> <n>	Configures the time between rekey message retransmissions and the number of retransmissions.
REKEY TRANSPORT UNICAST	Rekey messages are sent to the clients' IP.
REKEY TRANSPORT MULTICAST	Rekey messages are sent to a multicast IP.
SA IPSEC <id>	Permits entrance to an SA configuration menu.

You can access the configuration submenu for a specific SA through the SA IPSEC <sa-id> command.

Example:

```
IPSec GDOI config>sa ipsec 1
GDOI SA config>
```

The said submenu contains the following commands:

Command	Function
LIFETIME <time>	Configures the lifetime of the SAs created by the clients.
MATCH ADDRESS IPV4 <acclst>	Configures the access list used by the clients registered in this server.
REPLAY COUNTER	Activates replay in sequence.
REPLAY NONE	Deactivates anti-replay.
REPLAY TIME	Activates anti-replay through timestamp. This is the default option.
TRANSFORM-SET <ciph> <auth>	Defines the encryption and authentication algorithm to be used by the SAs created by the clients.

Example:
GDOI server with identifier 2.
The rekey is configured in multicast mode using address 239.0.0.2, and encryption for the rekey packets is aes 256 bits. To authenticate the rekey messages RSA MYKEY is used, which must be previously generated through the KEY RSA GENERATE command. The aes keys used are refreshed every 10 minutes, sending 3 message retransmissions with a 10 second lapse between each.
Regarding the SAs: an SA has been configured associated to access list number 100, which must be previously configured (please see manual Dm 752-I Access Control). The traffic pertaining to the said SA is encrypted using triple des and authenticated through sha. The keys are valid for 5 minutes, when this times out the server sends a rekey message to refresh them.

```
gdoi group 2
; -- GDOI user configuration --
      identity number 2
      rekey address ipv4 239.0.0.2
      rekey algorithm aes-256
      rekey authentication rsa MYKEY
      rekey lifetime seconds 10m
      rekey retransmit 10s number 3
      sa ipsec 1
          lifetime 5m
          match address ipv4 100
          transform-set tdes sha1
      exit
;
exit
```

3.5. FAULT-TOLERANT

The FAULT-TOLERANT command accesses the IPSecFT protocol configuration submenu.

The commands in this submenu depend on its operating mode, i.e. this doesn't offer the same commands if it is operating as master or as slave. The commands for both cases are given below:

Commands for slave	Function
ENABLE	Enables the IPSecFT protocol.
LIST	Lists the IPSecFT protocol configuration.
LISTEN-PORT	Listening port expecting the incoming IPSecFT connections.
MODE	Changes the operating mode between master and slave.
NO	Sets a command to its default value.
EXIT	Exits the IPSecFT configuration menu.

Commands for master	Function
ENABLE	Enables the IPSecFT protocol.
INHERIT-CONDITION	Selects the condition for sending the database sessions in IPSecFT to IPSec.
LIST	Lists the IPSecFT configuration.
LISTEN-PORT	Listening port expecting the incoming IPSecFT connections.
MODE	Changes the operating mode between master and slave.
NO	Sets a command to its default value.
SLAVE-ADDRESS	IP address to connect to the slave.
SLAVE-PORT	Port to connect to the slave.
SOURCE-ADDRESS	Source address to use when sending IPSecFT packets.
TIMERS	Configures the IPSecFT wait times.
EXIT	Exits the IPSecFT configuration menu.

"ENABLE"

Enables the IPSecFT protocol.

The IPSecFT protocol cannot be enabled without having previously configured the *source-address* command and the *slave-address* command. Contrariwise you receive an error.

Each time the mode is changed using the *mode* command, the protocol deactivates.

Example:

```
IPSecFT config>enable
CLI Error: Source address is not configured. Unable to enable
CLI Error: Command error
```

This example tries to enable IPSecFT without having configured the source to use for the IPSecFT packets.

Example:

```
IPSecFT config>enable
CLI Error: Slave address is not configured. Unable to enable
CLI Error: Command error
```

This example tries to enable IPSecFT without having configured the slave IP address in the example.

Example:

```
IPSecFT config>enable
```

The IPSecFT protocol is enabled in the example.

"INHERIT-CONDITION VRRP"

Selects the condition used to order the IPSec to establish the sessions that the IPSecFT has in its database. Please remember that these sessions in the database are the result of exchanging data in the IPSecFT protocol between the master/slave pair.

Example:

```
IPSecFT config>inherit-condition vrrp
```

This example selects which VRRP determines when the IPSec should be ordered to establish the IPSecFT database sessions.

"LIST"

Lists the protocol configuration. Depending on the operating mode, more or less information is shown.

Example:

```
IPSecFT config>list
Fault tolerant configuration:
        Enable:                 TRUE
        Mode:                   Master
        Slave server address:   1.1.1.1
        Slave server port:      52912
        Source address:         ethernet0/1
        Listen port:            52912
        Inactivity timeout:     500 milliseconds
        Keepalive period:       100 milliseconds
        Inherit condition:      VRRP
```

This example lists the configuration in master mode.

Example:

```
IPSecFT config>list
Fault tolerant configuration:
        Enable:                 FALSE
        Mode:                   Slave
        Listen port:            52912
```

This example lists the configuration in slave mode.

"LISTEN-PORT [PORT]"

Listening port for the incoming IPSecFT connections. When operating in master mode, the value for this command is transmitted to the pair operating in slave mode so that the connection with this port initiates.

Default port value is 52912.

Example:

```
IPSecFT config>listen-port 5645
```

In this example the listen port is set to 5645

"MODE [MASTER/SLAVE]"

This command selects the IPSecFT protocol operating mode. The possible operating modes are master or slave.

In the device operating as master, all the protocol parameters are configured. This is transmitted to the slave when it connects to it.

In the device operating as slave, only the listen port for the IPSecFT connections is configured. When the first is accepted, the parameters to initiate a new IPSecFT connection in the opposite direction are received. The slave device tries to establish the IPSecFT sessions with the parameters received in the last connection.

Example:

```
IPSecFT config>mode master
```

In this example, the device is configured as master.

"SLAVE-ADDRESS [IP-ADDRESS]"

This command configures the IP address that connects with the slave.

Example:

```
IPSecFT config>slave-address 1.1.1.1
```

In this example the slave IP address is configured as 1.1.1.1.

"SLAVE-PORT [PORT]"

This command configures the port to connect to the slave.

The default port value is 52912.

Example:

```
IPSecFT config>slave-port 4658
```

In this example the slave port is configured as 4658.

"SOURCE-ADDRESS [IP-ADDRESS/INTERFACE]"

This command configures in the master the source address to use in the IPSecFT packets or the interface that the IPSecFT packets are going to be transmitted through. In the slave, the source address used is the destination address of the first packet in the IPSecFT session establishment.

Through the IPSecFT session, an exchange of data between the two devices making up the Fault Tolerant IPSec Recovery system is executed. This said IPSecFT session must be established in a controlled way and over the path the user considers to be the most appropriate. By using the source-address command you can ensure that the session is established using the selected source.

Example:

```
IPSecFT config>source-address ethernet0/0
```

In this example, the IPSecFT packets are configured to transmit through the ethernet0/0 interface.

"TIMERS KEEPALIVE-PERIOD [KEEPALIVE] INACTIVITY-TIMEOUT [INACTIVITY]"

This command configures the times the IPSecFT protocol waits before taking decisions.

The *keepalive* value refers to the time the protocol waits before executing the next action, i.e. the time it waits before sending monitoring packets, the time waited before polling the VRRP or the time waited before changing states.

The inactivity value configures the maximum time permitted without receiving packets from the other end before considering the IPSecFT session as down.

Low values in this command can provoke high CPU usage.

The default value for the *keepalive* is 100 milliseconds and for the *inactivity* 500 milliseconds.

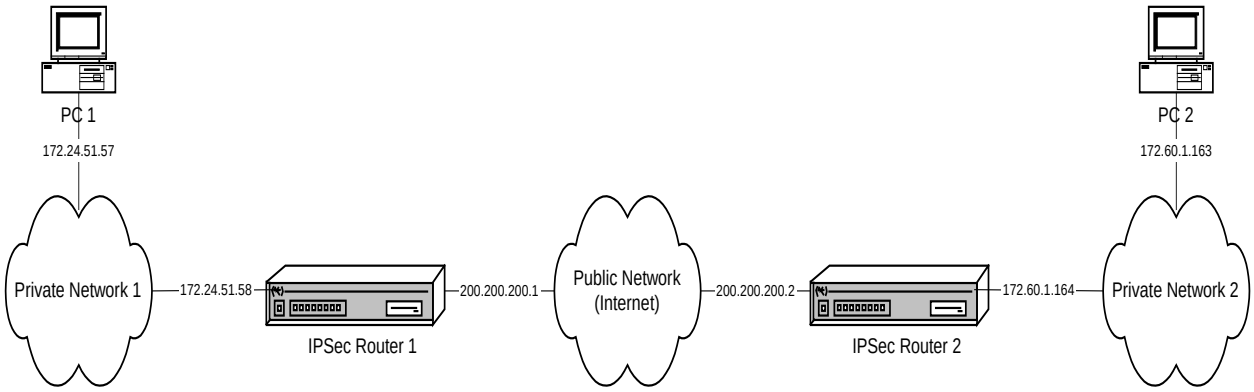
Example:

```
timers keepalive-period 200 inactivity-timeout 1000
```

In this example, the keepalive time is configured to 200 milliseconds and the inactivity time to 1 second.

4. Examples

4.1. Example 1: Manual Mode



This is the process of creating a new virtual private network (VPN) between Host A and Host B. The rest of the traffic between private networks will be allowed to pass in normal mode. Create an IPSec Tunnel with Triple DES encryption and SHA1 authentication in order to comply with the security requirements.

- *Creating the access control lists*

As already mentioned, the Tunnel clients are host A and host B.

Router 1:

```
Config>FEATURE ACCESS-LISTS

-- Access Lists user configuration --
Access Lists config>access-list ?
  <1..99>       Standard Access List number (1-99)
  <100..199>    Extended Access List number (100-199)
Access Lists config>access-list 101


Extended Access List 101>entry 1 ?
  default           Sets default values to an existing or a new entry
  permit            Configures type of entry or access control as permit
  deny              Configures type of entry or access control as deny
  source            Source menu: subnet or port
  destination       Destination menu: subnet or port
  protocol          Protocol
  protocol-range    Protocol range
  dscp              IP type-of-service byte value
  connection        IP connection identifier (rule)
  description       Sets a description for the current entry
  no                Negates a command or sets its defaults
Extended Access List 101>entry 1 source ?
  address       IP address and mask of the source subnet
  port-range    Source port range
Extended Access List 101>entry 1 source address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
Extended Access List 101>entry 1 source address 172.24.51.57 ?
  <a.b.c.d>    Ipv4 format
Extended Access List 101>entry 1 source address 172.24.51.57 255.255.255.255
Extended Access List 101>entry 1 destination ?
```

```
  address       IP address and mask of the destination subnet
  port-range    Destination port range
Extended Access List 101>entry 1 destination address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
Extended Access List 101>entry 1 destination address 172.60.1.163 ?
  <a.b.c.d>    Ipv4 format
Extended Access List 101>entry 1 destination address 172.60.1.163 255.255.255.255
Extended Access List 101>
```

The configured access list is as follows:

```
Extended Access List 101>LIST ALL-ENTRIES

Extended Access List 101, assigned to no protocol

1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0

Extended Access List 101>
```

Through the "SHOW CONFIG" command the configuration can be displayed and used in the future by introducing this command in the console as shown below:

```
Access Lists config>SHOW CONFIG
; Showing System Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

   access-list 101
;
      entry 1 permit
      entry 1 source address 172.24.51.57 255.255.255.255
      entry 1 destination address 172.60.1.163 255.255.255.255
;
   exit
;
Access Lists config>
```

I.e. you could have configured the required entry in the access list in the following way:

```
Access Lists config>
   access-list 101
      entry 1 permit
      entry 1 source address 172.24.51.57 255.255.255.255
      entry 1 destination address 172.60.1.163 255.255.255.255
```

Please note that in the Router 2 example the source and destination addresses, as regards the Router 1 example, have been interchanged.

Router 2:

```
Access Lists config>
   access-list 101
      entry 1 permit
      entry 1 source address 172.60.1.163 255.255.255.255
      entry 1 destination address 172.24.51.57 255.255.255.255
```

- *Creating Templates*

Subsequently the security patterns or Templates are created:

Router 1:

The first step is to enable IPSec.

```
Config>PROTOCOL IP

-- Internet protocol user configuration --
IP config> IPSEC

-- IPSec user configuration --
IPSec config> ENABLE
IPSec config>
```

Next you need to configure the required Template:

```
IPSec config>TEMPLATE 2
  default                sets default values to a template or creates a new
                         one
  dynamic                dynamic template
  manual                 manual template
  isakmp                 isakmp template
  source-address         tunnel's local IP address
  destination-address    Address of the other remote end of the tunnel
  discover               use TED to discover the remote end of the tunnel
  backup-destination     backup destination IP address
  spi                    Security Parameter Index
  key                    template encryption DES key
  tkey                   triple DES key
  md5key                 MD5 key
  sha1key                SHA1 key
  antireplay             activates the Anti-Replay service
  padding-check          enables padding check
  udp-encapsulation      enables UDP encapsulation
  udp-ike                enables IKE UDP encapsulation
  life                   introduces the SAs life span created from the
                         template
  ike                    configures parameters relative to the IPSec IKE mode
  keepalive              enables the available keepalive services
  encap                  type of encapsulation for packets
  config                 isakmp configuration
  aggressive             aggresive configuration mode ciphered/clear
  napt-id-skipped        ipsec must not mark packets for napt
  fast-forwarder         force fast-forwarding of packets
  no                     deletes a backup destination or disables an option
IPSec config>TEMPLATE 2 ?
  esp      ESP security service (Encapsulating Security Payload)
  ah       AH security service (Authentication Header)
IPSec config>TEMPLATE 2 esp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
IPSec config>TEMPLATE 2 esp tdes ?
  md5       authentication algorithm MD5
  sha1      authentication algorithm SHA1
  none      no authentication algorithm
IPSec config>TEMPLATE 2 esp tdes sha1
IPSec config>TEMPLATE 2 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config>TEMPLATE 2 source-address 200.200.200.1
IPSec config>TEMPLATE 2 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config>TEMPLATE 2 destination-address 200.200.200.2
IPSec config>TEMPLATE 2 spi ?
  <257..65535>    Enter SPI (SPI > 256):
IPSec config>TEMPLATE 2 spi 280
IPSec config>TEMPLATE 2 tkey h53s45ef46agv4646n2j8qpo

IPSec config>TEMPLATE 2 sha1key b74hd748ghzm67k6m6d1
```

The Template configuration is established as shown below:

```
IPSec config>LIST TEMPLATE ALL
TEMPLATES
2 manual ESP-3DES ESP-SHA1  SRC=200.200.200.1 DES=200.200.200.2 SPI=280

IPSec config>
```

Through the "SHOW CONFIG" command you obtain the following:

```
IPSec config>SHOW CONFIG
; Showing System Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

      enable
;
      template 2 default
      template 2 manual esp tdes sha1
      template 2 source-address 200.200.200.1
      template 2 destination-address 200.200.200.2
      template 2 spi 280
      template 2 tkey h53s45ef46agv4646n2j8qpo
      template 2 sha1key b74hd748ghzm67k6m6d1
;
IPSec config>
```

I.e. The Template could also have been configured like this:

```
IPSec config>
      enable
      template 2 default
      template 2 manual esp tdes sha1
      template 2 source-address 200.200.200.1
      template 2 destination-address 200.200.200.2
      template 2 spi 280
      template 2 tkey h53s45ef46agv4646n2j8qpo
      template 2 sha1key b74hd748ghzm67k6m6d1
```

Router 2:

```
IPSec config>
      enable
      template 2 default
      template 2 manual esp tdes sha1
      template 2 source-address 200.200.200.2
      template 2 destination-address 200.200.200.1
      template 2 spi 280
      template 2 tkey h53s45ef46agv4646n2j8qpo
      template 2 sha1key b74hd748ghzm67k6m6d1
```

Please note that in the Router 2 example the source and destination addresses, as regards the Router 1 example, have been interchanged.

The SPI must be the same in both Routers.

- *Creating the SPDs*

In order to complete the Security Policies database (***SPD***), it is necessary to "map" the elements from the Access Control list to the chosen Templates.

Router 1:

```
IPSec config>assign-access-list ?
  <1..65535>    Enter extended access list id
IPSec config>assign-access-list 101
IPSec config>map-template ?
  <100..65535>    Enter extended access list id
IPSec config>map-template 101 ?
  <1..65535>    Enter template id(1-65534)
IPSec config>map-template 101 2
IPSec config>
```

Or:

```
IPSec config>
      assign-access-list 101
      map-template 101 2
```

The IPSec configuration is established as follows:

```
IPSec config>LIST ALL
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 2


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0

TEMPLATES
2 manual ESP-3DES ESP-SHA1  SRC=200.200.200.1 DES=200.200.200.2 SPI=280

0 key entries
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Use software exponentiation

IPSec config>
```

Through the “SHOW CONFIG” command you obtain the following:

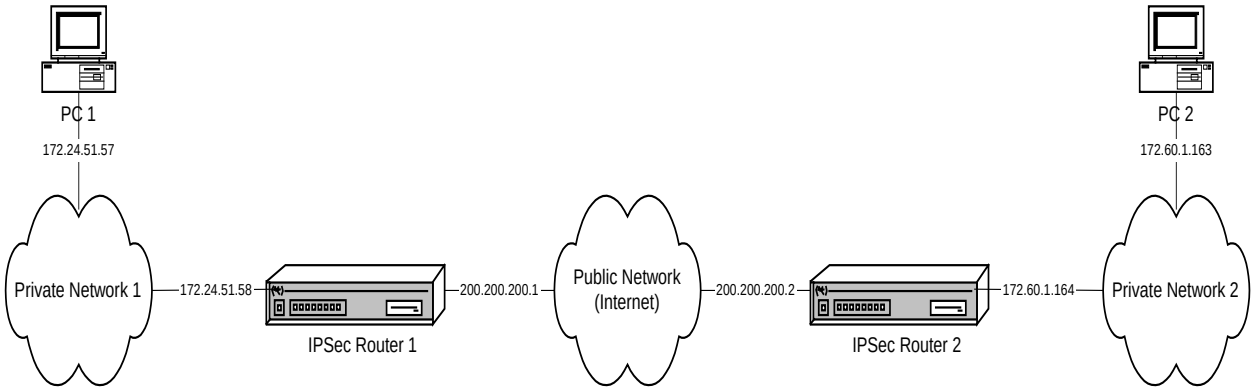
```
IPSec config>SHOW CONFIG
; Showing System Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

      enable
      assign-access-list 101
;
      template 2 default
      template 2 manual esp tdes sha1
      template 2 source-address 200.200.200.1
      template 2 destination-address 200.200.200.2
      template 2 spi 280
      template 2 tkey h53s45ef46agv4646n2j8qpo
      template 2 sha1key b74hd748ghzm67k6m6d1
;
      map-template 101 2
IPSec config>
```

Router 2:

```
IPSec config>
      assign-access-list 101
      map-template 101 2
```

Now any communication between hosts A and B is securely carried out regarding the said communication. However, the complete security of the communications system, based as well in the devices, introduced keys, modification permissions, etc., is the responsibility of the user.

4.2. Example 2: Dynamic mode (IPSEC IKE Main Mode)



The scenario for this example is the same as for the previous one. However the Tunnel is now going to be established based on dynamic Templates so that the communications, keys etc are automatically negotiated using the Main mode.

- *Creating the access control lists*

There is no further modification in this configuration with regard to example 1.

- *Creating Templates*

At this point you need to create the ISAKMP and dynamic Templates. The final command is important to introduce the Pre-shared key which must be the same in both devices. By default, the negotiation mode is Main Mode where the identities of the end routers for the Tunnel are masked.

Although the same lifetimes have also been introduced, these parameters can be different and be negotiated.

Router 1:

```
IPSec config>ENABLE
IPSec config>TEMPLATE 1 ?
  default               sets default values to a template or creates a new
                        one
  dynamic               dynamic template
  manual                manual template
  isakmp                isakmp template
  source-address        tunnel's local IP address
  destination-address   Address of the other remote end of the tunnel
  discover              use TED to discover the remote end of the tunnel
  backup-destination    backup destination IP address
  spi                   Security Parameter Index
  key                   template encryption DES key
  tkey                  triple DES key
  md5key                MD5 key
  sha1key               SHA1 key
  antireplay            activates the Anti-Replay service
  padding-check         enables padding check
  udp-encapsulation     enables UDP encapsulation
  udp-ike               enables IKE UDP encapsulation
  life                  introduces the SAs life span created from the
                        template
  ike                   configures parameters relative to the IPSec IKE mode
  keepalive             enables the available keepalive services
  encap                 type of encapsulation for packets
  config                isakmp configuration
  aggressive            aggresive configuration mode ciphered/clear
  napt-id-skipped       ipsec must not mark packets for napt
  fast-forwarder        force fast-forwarding of packets
  no                    deletes a backup destination or disables an option
IPSec config>TEMPLATE 1 isakmp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
IPSec config>TEMPLATE 1 isakmp tdes ?
  md5     authentication algorithm MD5
  sha1    authentication algorithm SHA1
IPSec config>TEMPLATE 1 isakmp tdes sha1
IPSec config> TEMPLATE 1 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config> TEMPLATE 1 destination-address 200.200.200.2
IPSec config> TEMPLATE 1 life ?
  type        type of life duration for the SA
  duration    life duration
IPSec config> TEMPLATE 1 life duration ?
  seconds    lifetime in seconds
  kbytes     lifetime in Kbytes
IPSec config> TEMPLATE 1 life duration seconds ?
  <0s..3550w>    Time value
IPSec config> TEMPLATE 1 life duration seconds 43200
IPSec config> TEMPLATE 3 dynamic ?
  esp      ESP security service (Encapsulating Security Payload)
  ah       AH security service (Authentication Header)
IPSec config> TEMPLATE 3 dynamic esp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    encryption algorithm AES using 128-bit key (Advanced Encryption
```

```
            Standard)
  aes192    encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
IPSec config> TEMPLATE 3 dynamic esp tdes ?
md5       authentication algorithm MD5
sha1      authentication algorithm SHA1
none      no authentication algorithm
IPSec config> TEMPLATE 3 dynamic esp tdes md5
IPSec config> TEMPLATE 3 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
IPSec config> TEMPLATE 3 source-address 200.200.200.1
IPSec config> TEMPLATE 3 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
IPSec config> TEMPLATE 3 destination-address 200.200.200.2
IPSec config> TEMPLATE 3 life ?
  type          type of life duration for the SA
  duration      life duration
IPSec config> TEMPLATE 3 life type ?
  seconds       lifetime in seconds
  kbytes        lifetime in kbytes
  both          lifetime in seconds and kbytes
IPSec config> TEMPLATE 3 life type both
IPSec config> TEMPLATE 3 life duration ?
  seconds       lifetime in seconds
  kbytes        lifetime in kbytes
IPSec config> TEMPLATE 3 life duration seconds ?
  <0s..3550w>    Time value
IPSec config> TEMPLATE 3 life duration seconds 14400
IPSec config> TEMPLATE 3 life duration ?
  seconds       lifetime in seconds
  kbytes        lifetime in kbytes
IPSec config> TEMPLATE 3 life duration kbytes ?
  <0..4294967295>    kbytes
IPSec config> TEMPLATE 3 life duration kbytes 0
IPSec config> KEY PRESHARED IP 200.200.200.2 plain 1234567890123456

IPSec config>
```

You could have also used the configuration in text mode (taken from that obtained through the "SHOW CONFIG" command).

```
IPSec config>
      enable
      template 1 default
      template 1 isakmp tdes sha1
      template 1 destination-address 200.200.200.2
      template 1 life duration seconds 43200
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 200.200.200.1
      template 3 destination-address 200.200.200.2
      template 3 life type both
      template 3 life duration seconds 14400
      key preshared ip 200.200.200.2 plain 1234567890123456
```

Router 2:

```
IPSec config>
      enable
      template 1 default
      template 1 isakmp tdes sha1
      template 1 destination-address 200.200.200.1
```

```
      template 1 life duration seconds 43200
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 200.200.200.2
      template 3 destination-address 200.200.200.1
      template 3 life type both
      template 3 life duration seconds 14400
      key preshared ip 200.200.200.1 plain 1234567890123456
```

- *Creating the SPD´s*

Finally, you need to establish the *SPDs*:

Router 1:

```
IPSec config>assign-access-list ?
  <1..65535>    Enter extended access list id
IPSec config>assign-access-list 101
IPSec config>map-template ?
  <100..65535>    Enter extended access list id
IPSec config>map-template 101 ?
  <1..65535>    Enter template id(1-65534)
IPSec config>map-template 101 3
IPSec config>
```

Or:

```
IPSec config>
      assign-access-list 101
      map-template 101 3
```

The IPSec final configuration is established as shown below:

```
IPSec config>LIST ALL
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 3


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=172.24.51.57/32  DES=172.60.1.163/32  Conn:0

TEMPLATES
1 isakmp 3DES SHA1  DES=200.200.200.2
   LifeTime:12h0m0s
   IKE MAIN
   PRESHARED
   addr4 ID TYPE
   OAKLEY GROUP 1

3 dynamic ESP-3DES ESP-MD5  SRC=200.200.200.1 DES=200.200.200.2
   LifeTime:4h0m0s 0 kbytes
   PFS disabled

1 key entries
   200.200.200.2 ************
0 rsakey entries
```

```
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Use software exponentiation

IPSec config>
```

With the "SHOW CONFIG" command:

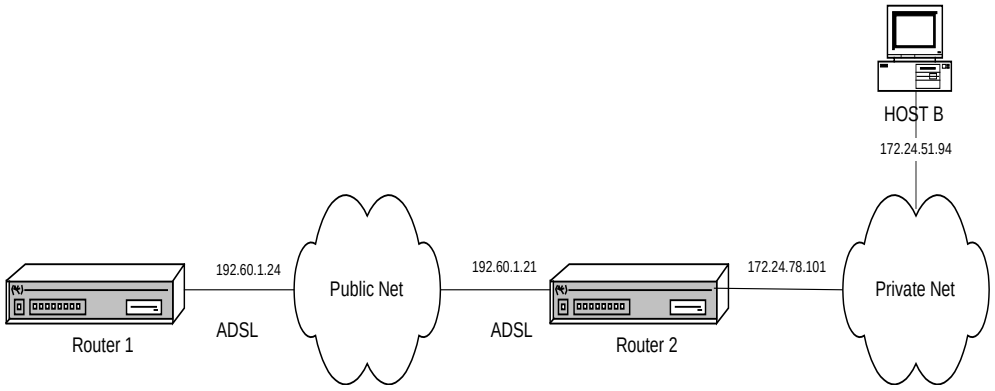
```
IPSec config>SHOW CONFIG
; Showing System Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes sha1
      template 1 destination-address 200.200.200.2
      template 1 life duration seconds 43200
;
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 200.200.200.1
      template 3 destination-address 200.200.200.2
      template 3 life type both
      template 3 life duration seconds 14400
      template 3 life duration kbytes 0
;
      map-template 101 3
      key preshared ip 200.200.200.2 ciphered 0xE21C47018BC8B868FB72F48DC4363FC0
CFABF60C9FFE0286
IPSec config>
```

Router 2:

```
IPSec config>
      assign-access-list 101
      map-template 101 3
```

Communication between hosts A and B is now securely carried out, with the Pre-shared key as the only key to protect in this case.

4.3. Example 3: Dynamic mode (IPSEC IKE Aggressive mode) with one Tunnel end having an unknown address



This scenario reflects how to connect two routers through a virtual private network (VPN) using an ADSL line as the connection means. You create an IPSec Tunnel based on the dynamic Templates, with DES encryption and MD5 authentication as security requirements for the ISAKMP negotiation and ESP service with DES encryption and SHA1 authentication in the SA negotiation of the Tunnel. The Tunnel will be based on dynamic Templates so that the communications, keys, etc., are automatically negotiated using Aggressive mode.

The Aggressive mode has the advantage that Router 2 does not need to know the IP address of the other end of the Tunnel. This signifies that this configuration is perfectly adequate for many devices to connect to a single Router 2 by simply knowing the *hostname* and the common key between them. Router 1 must know the IP address of the router through which the Tunnel is going to be established, as it is this router that initiates the negotiation and must know which IP address it needs to connect to.

Firstly, we are going to give an in-depth explanation on how to configure Router 1. Once this has been configured, we will configure Router 2, going into detail on those parameters which differ from the Router 1 configuration.

a) Configuring the Router 1

- *Configuring the hostname, IP addresses and rules*

As previously indicated, authentication is carried out through the *hostname* rather than the IP addresses. Therefore, the first thing you have to configure is the name you are going to give to the device.

```
Teldat              (c)1996-2002

Router model C5i IPSec 1 17 CPU MPC860      S/N: 391/02415
1 LAN, 1 WAN Line, 1 ISDN Line, 1 ADSL Line


*PROCESS 4


Config>SET HOSTNAME GAS1
```

Subsequently, you need to assign the IP address for the ADSL interface. You also need to add a static route indicating that all the packets you are going to send to the private network are transmitted using the other end of the IPSec Tunnel as the link port.

You can also specify a connection identifier for the traffic between the routers. This is only necessary if you wish to treat the packets differently in different connections.

```
GAS1 Config>LIST DEVICES

 Interface       Con    Type of interface             CSR     CSR2  int
 ethernet0/0     LAN1   Quicc Ethernet           fa200a00 fa203c00   5e
 serial0/0       WAN1   X25                      fa200a20 fa203d00   5d
 atm0/0          ADSL1  Async Transfer Mode      fa200a60 fa203f00   55
 bri0/0          ISDN1  ISDN Basic Rate Int      fa200a40 fa203e00   5c
 x25-node        ---    Router->Node                    0             0
 ppp1            ---    Generic PPP                     0             0
 ppp2            ---    Generic PPP                     0             0
Config>

GAS1 Config>PROTOCOL IP
```

- *Creating the access control lists*

Once you have configured all the IP's own parameters, you need to configure the IPSec itself.

The first thing that you must configure is the access control lists. To do this, you need to access the generic lists configuration menu, select a number from the list corresponding to an extended list (between 100 and 199), indicate an entry ID within the list, in this case 1 and give the required value to the following parameters:

- The source IP address, this will be the one previously configured in the ADSL interface.
- The destination IP, this is the device with which you are going to establish an IPSec Tunnel, in our case this deals with a Router 2.
- The connection: you have to indicate the connection ID assigned to the Tunnel's traffic. This ID is displayed through the **LIST RULE** command. In this particular example, it is not necessary to assign the connection as no distinction is made when dealing with the packets according to the connection.
- The action to be taken in the packets, in this case, IPSec procedure (PERMIT).

```
GAS1 Config>FEATURE ACCESS-LISTS

-- Access Lists user configuration --
GAS1 Access Lists config>access-list ?
  <1..99>       Standard Access List number (1-99)
  <100..199>    Extended Access List number (100-199)
GAS1 Access Lists config>access-list 102


GAS1 Extended Access List 102>entry 1 ?
  default           Sets default values to an existing or a new entry
  permit            Configures type of entry or access control as permit
  deny              Configures type of entry or access control as deny
  source            Source menu: subnet or port
  destination       Destination menu: subnet or port
  protocol          Protocol
  protocol-range    Protocol range
  dscp              IP type-of-service byte value
  connection        IP connection identifier (rule)
  description       Sets a description for the current entry
  no                Negates a command or sets its defaults
GAS1 Extended Access List 102>entry 1 source ?
  address       IP address and mask of the source subnet
  port-range    Source port range
GAS1 Extended Access List 102>entry 1 source address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
```

```
GAS1 Extended Access List 102>entry 1 source address 192.60.1.24 ?
  <a.b.c.d>    Ipv4 format
GAS1 Extended Access List 102>entry 1 source address 192.60.1.24 255.255.255.255
GAS1 Extended Access List 102>entry 1 destination ?
  address       IP address and mask of the destination subnet
  port-range    Destination port range
GAS1 Extended Access List 102>entry 1 destination address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
GAS1 Extended Access List 102>entry 1 destination address 172.24.0.0 ?
  <a.b.c.d>    Ipv4 format
GAS1 Extended Access List 102>entry 1 destination address 172.24.0.0 255.255.0.0
GAS1 Extended Access List 102>entry 1 permit
GAS1 Extended Access List 102>
```

Or:

```
GAS1 Access Lists config>
   access-list 102
      entry 1 permit
      entry 1 source address 192.60.1.24 255.255.255.255
      entry 1 destination address 172.24.0.0 255.255.0.0
```

- *Creating Templates*

Now you need to create the ISAKMP and dynamic Templates. The last command is important to introduce the Pre-Shared key that must be the same in both devices. The difference between this example and the previous one is that here the negotiation mode is Aggressive Mode, where the identities of the Tunnel's end routers are not masked and the IP address of the other end of the Tunnel is unknown.

Although you have also introduced the same lifetimes, these parameters may be different and be negotiated in such a way that the negotiation result will be the smallest configured at the Tunnel ends.

When creating the ISAKMP Template, you need to indicate the encryption type (DES) and the authentication (MD5) which are going to be used, as indicated in the initial security specifications.

On creating the Template, you need to indicate the ID number that will be used in the rest of the configuration for this Template. You also need to indicate the Tunnel's destination IP which you are going to connect to and additionally Aggressive mode will be used, as the authentication executed sends the hostname rather than the IP address. This is extremely useful when you do not know the IP address of the other end of the Tunnel a priori, as in the case of Router 2 in this example, where it does not need to know the IP address of the Routers to be connected to it. The IPSec Tunnel can be created by simply knowing the hostname.

Through the **TEMPLATE 1 IKE IDTYPE FQDN** command, you indicate that the authentication uses the hostname instead of the IP address which is the default option.

```
GAS1 Config>PROTOCOL IP

-- Internet protocol user configuration --
GAS1 IP config>IPSEC

-- IPSec user configuration --
GAS1 IPSec config>ENABLE
GAS1 IPSec config>TEMPLATE 1 ?
  default               sets default values to a template or creates a new
                        one
  dynamic               dynamic template
  manual                manual template
  isakmp                isakmp template
  source-address        tunnel's local IP address
```

```
  destination-address    Address of the other remote end of the tunnel
  discover               use TED to discover the remote end of the tunnel
  backup-destination     backup destination IP address
  spi                    Security Parameter Index
  key                    template encryption DES key
  tkey                   triple DES key
  md5key                 MD5 key
  sha1key                SHA1 key
  antireplay             activates the Anti-Replay service
  padding-check          enables padding check
  udp-encapsulation      enables UDP encapsulation
  udp-ike                enables IKE UDP encapsulation
  life                   introduces the SAs life span created from the
                         template
  ike                    configures parameters relative to the IPSec IKE mode
  keepalive              enables the available keepalive services
  encap                  type of encapsulation for packets
  config                 isakmp configuration
  aggressive             aggresive configuration mode ciphered/clear
  napt-id-skipped        ipsec must not mark packets for napt
  fast-forwarder         force fast-forwarding of packets
  no                     deletes a backup destination or disables an option
GAS1 IPSec config>TEMPLATE 1 isakmp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
GAS1 IPSec config>TEMPLATE 1 isakmp des ?
  md5     authentication algorithm MD5
  sha1    authentication algorithm SHA1
GAS1 IPSec config>TEMPLATE 1 isakmp des md5
GAS1 IPSec config>TEMPLATE 1 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
GAS1 IPSec config>TEMPLATE 1 destination-address 192.60.1.21
GAS1 IPSec config>TEMPLATE 1 ike ?
  ca                     CA
  mode                   mode in which phase I of the ISAKMP/IKE exchange is
                         carried out
  method                 establishes the authentication method used by the
                         device
  pfs                    enables the Perfect Forward Secrecy service
  idtype                 types of identifiers used during phase 1 of the
                         ISAKMP/IKE exchange
  crl                    CRL
  group                  group
  jfe                    JFE
  lifetime-negotiation   enables lifetime negotiation
  no                     disables an IKE option
GAS1 IPSec config>TEMPLATE 1 ike mode ?
  aggressive    aggressive mode
  main          main mode
GAS1 IPSec config>TEMPLATE 1 ike mode aggressive
GAS1 IPSec config>TEMPLATE 1 ike idtype ?
  ip        IP Address
  fqdn      FQDN
  ufqdn     UFQDN
  keyid     keyid
  asn-dn    asn-dn
GAS1 IPSec config>TEMPLATE 1 ike idtype fqdn
GAS1 IPSec config>
```

Or in a more condensed form if you use the configuration in text mode:

```
GAS1 IPSec config>
    enable
    template 1 default
    template 1 isakmp des md5
    template 1 destination-address 192.60.1.21
    template 1 ike mode aggressive
    template 1 ike idtype fqdn
```

Once the ISAKMP Template has been created, you need to create the DYNAMIC Template.

Firstly, you indicate the type of service, ESP or AH. The ESP service provides confidentiality, authentication of the source address in each IP packet, integrity and protection against replays, while the AH service does not provide confidentiality. Subsequently you have to indicate that this is dealing with encryption (DES) and the type of authentication (SHA1), as indicated in the initial security specifications.

When indicating the Template ID, you must chose a different one from the above ISAKMP Template (1), as contrariwise the previous configuration will be overwritten with the DYNAMIC Template configuration. In the example, the ID is 2.

In the same way as in the ISAKMP Template, you have to indicate the destination address, however you also have to indicate what the source address is i.e. the address of your ADSL interface. In this Template we have also enabled the KEEPALIVE option thus ensuring that the other end maintains its SA open.

```
GAS1 IPSec config>TEMPLATE 2 dynamic ?
  esp    ESP security service (Encapsulating Security Payload)
  ah     AH security service (Authentication Header)
GAS1 IPSec config>TEMPLATE 2 dynamic esp ?
  des       encryption algorithm DES (Data Encryption Standard)
  tdes      encryption algorithm TDES (Triple Data Encryption Standard)
  aes128    encryption algorithm AES using 128-bit key (Advanced Encryption
            Standard)
  aes192    encryption algorithm AES using 192-bit key (Advanced Encryption
            Standard)
  aes256    encryption algorithm AES using 256-bit key (Advanced Encryption
            Standard)
GAS1 IPSec config>TEMPLATE 2 dynamic esp des ?
  md5     authentication algorithm MD5
  sha1    authentication algorithm SHA1
  none    no authentication algorithm
GAS1 IPSec config>TEMPLATE 2 dynamic esp des sha1
GAS1 IPSec config>TEMPLATE 2 source-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
GAS1 IPSec config>TEMPLATE 2 source-address 192.60.1.24
GAS1 IPSec config>TEMPLATE 2 destination-address ?
  <a.b.c.d>      Ipv4 format
  <word>         Text
GAS1 IPSec config>TEMPLATE 2 destination-address 192.60.1.21
GAS1 IPSec config>TEMPLATE 2 keepalive ?
  keepalive    enables the available keepalive services
  dpd          enables the DPD service (Dead Peer Detection)
  no           disables the available keepalive services
GAS1 IPSec config>TEMPLATE 2 keepalive keepalive
GAS1 IPSec config>
```

Or:

```
GAS1 IPSec config>
      template 2 default
      template 2 dynamic esp des sha1
      template 2 source-address 192.60.1.24
      template 2 destination-address 192.60.1.21
      template 2 keepalive keepalive
```

Lastly, you need to configure the Pre-shared key. This key is common to both ends of the Tunnel.

When introducing the key, you need to indicate this is dealing with a Pre-shared key. We are also going to introduce a name instead of an IP address as previously explained.

The name to be introduced corresponds to the **domain name** of the other end of the Tunnel.

In addition to the device hostname, it's possible to configure the device domain. This can be carried out in the following way:

```
GAS1 IP config>DNS-DOMAIN-NAME ?
  <word>    Text
GAS1 IP config>DNS-DOMAIN-NAME madrid.es
Domain name : madrid.es
Domain Name configured.
GAS1 IP config>
```

In this example, we have not used the domain name. Therefore, on displaying the domain name, this indicates that it is not configured and that the name to be used will be "**GAS1.**" This will be the name you need to configure when indicating the Pre-shared common keys at the other end of the Tunnel, i.e. in Router 2.

```
GAS1 IP config>LIST DNS-DOMAIN-NAME
No Domain Name configured.
Partial DNS name : GAS1.
```

In Router 1, you need to introduce the hostname to be used in the "**HOST.**" key as the domain in Router 2 has not been configured either. Only the device hostname as HOST has been configured.

```
GAS1 IPSec config>KEY PRESHARED HOSTNAME HOST. plain 1234567890123456
```

- *Creating SDPs*

Lastly, you need to establish the *SPD's* i.e. relating a control access to a created Template. In the below example, the configured generic list is 102, and the Template that this must be related to is dynamic i.e. ID 2.

```
GAS1 IPSec config>assign-access-list ?
  <1..65535>    Enter extended access list id
GAS1 IPSec config>assign-access-list 102
GAS1 IPSec config>map-template ?
  <100..65535>    Enter extended access list id
GAS1 IPSec config>map-template 102 ?
  <1..65535>    Enter template id(1-65534)
GAS1 IPSec config>map-template 102 2
GAS1 IPSec config>
```

In text mode:

```
GAS1 IPSec config>
      assign-access-list 102
      map-template 102 2
```

The IPSec configuration in Router 1 is established as follows:

```
GAS1 IPSec config>LIST ALL
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 102
  Templates: 2


Extended Access List 102, assigned to IPSec

1     PERMIT  SRC=192.60.1.24/32  DES=172.24.0.0/16  Conn:0

TEMPLATES
1 isakmp DES MD5  DES=192.60.1.21
   LifeTime:1h0m0s
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1

2 dynamic ESP-DES ESP-SHA1  SRC=192.60.1.24 DES=192.60.1.21
   LifeTime:1h0m0s
   PFS disabled
   Keep Alive enabled

1 key entries
   HOST. ************
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10

SA's purge timeout:  15

Use software exponentiation

GAS1 IPSec config>
```

Through the “SHOW CONFIG” command, you obtain the following:

```
GAS1 IPSec config>SHOW CONFIG
; Showing Menu and Submenus Configuration ...
; Router C5i IPSec 1 17 Version 10.0.0CAI

      enable
      assign-access-list 102
;
      template 1 default
      template 1 isakmp des md5
      template 1 destination-address 192.60.1.21
      template 1 ike mode aggressive
      template 1 ike idtype fqdn
;
      template 2 default
```

```
      template 2 dynamic esp des sha1
      template 2 source-address 192.60.1.24
      template 2 destination-address 192.60.1.21
      template 2 keepalive keepalive
;
      map-template 102 2
      key preshared hostname HOST. ciphered 0xE21C47018BC8B868FB72F48DC4363FC0CF
ABF60C9FFE0286
GAS1 IPSec config>
```

b) Configuring the Router 2

- *Configuring the hostname, IP addresses and rules*

Hostname and the IP protocol parameters configuration are similar to that executed for Router 1.

```
Teldat              (c)1996-2002

Router model Centrix SEC (c) 1 36 CPU MPC860      S/N: 359/00144
1 LAN

*PROCESS 4
User Configuration
Config>SET HOSTNAME HOST
```

On configuring the IP protocol, care must be taken when configuring the interface addresses as the ethernet0/0 interface connects the network card with the 172.24.0.0 LAN. You also need to assign the IP address to the ADSL interface where the IPSec Tunnel connection is carried out.

```
HOST IP config>address atm0/0 192.60.1.24 255.255.255.0
HOST IP config>address ethernet0/0 172.24.78.101 255.255.0.0
```

- *Creating the access control lists*

Once all the IP parameters have been configured, you need to configure the IPSec itself.

Configuring the access control lists is similar to the way this was carried out for Router 1. Care must be taken when configuring the source and destination IP addresses.

```
HOST Access Lists config>
   access-list 103
      entry 1 permit
      entry 1 source address 172.24.0.0 255.255.0.0
      entry 1 destination address 192.60.1.24 255.255.255.255
```

- *Creating Templates*

As done for Router 1, you need to create the ISAKMP and dynamic Templates with Aggressive Mode as the negotiation mode. The Pre-shared key must be the same as that configured in Router 1, however in this case indicating that the key corresponds to the *hostname* "**GAS1**".

When creating the ISAKMP Template, you need to indicate the encryption type (DES) and the authentication (MD5) which are going to be used, as indicated in the initial security specifications. This coincides with that previously configured in Router 1.

On creating the Template, you need to indicate the ID number that will be used in the rest of the configuration for this Template. You also need to indicate the Tunnel's destination IP which you are going to connect to; however as the IP address of the device which is going to connect to Router 2 is unknown and we only know the *hostname*, the **destination IP** address will be **0.0.0.0**. Additionally

you need to indicate you are going to use Aggressive mode and that the IDTYPE is FQDN so that the hostname is used in the authentication instead of the IP address which is the default option.

```
HOST IPSec config>
      enable
      template 1 default
      template 1 isakmp des md5
      template 1 destination-address 0.0.0.0
      template 1 ike mode aggressive
      template 1 ike idtype fqdn
```

Once the ISAKMP Template has been created, you need to create the DYNAMIC Template with ESP service, DES encryption and SHA1 authentication as done for Router 1. When indicating the Template ID, you must chose a different one from the above ISKMP Template (1), as contrariwise the previous configuration will be overwritten with the DYNAMIC Template configuration. In the example, the ID is 2.

In the same way as in the ISAKMP Template, you have to indicate the **destination address** (**0.0.0.0**), however you also have to indicate what the source address will be i.e. the address of your ADSL interface. The KEEPALIVE option is not enabled in this Template to free process time for Router 2 and it is the routers connecting to this that have to check that the SA is open.

```
HOST IPSec config>
      template 2 default
      template 2 dynamic esp des sha1
      template 2 source-address 192.60.1.21
      template 2 destination-address 0.0.0.0
      template 2 life duration seconds 1800
```

Lastly, you need to configure the Pre-shared key. This key is common to both ends of the Tunnel.

When introducing the key, you need to indicate this is dealing with a Pre-shared key. We are also going to introduce a name instead of an IP address as previously explained.

The name to be introduced corresponds to the **domain name** of the other end of the Tunnel as explained in the case of Router 1.

The name used in this example is "**GAS1.**" This is the Router 1 domain name.

```
HOST IPSec config> KEY PRESHARED HOSTNAME GAS1. plain 1234567890123456
HOST IPSec config>
```

If more routers apart from Router 1 are going to be connected to this Router, you must specify a hostname and the corresponding key for each of them.

- *Creating SPDs*

Lastly, you need to establish the *SPD's* i.e. relating a control access to a created Template. In the below example, the configured extended list that must be assigned to IPSec and associated with a Template is the 103, and the Template that must be related is the dynamic i.e. ID 2.

```
HOST IPSec config>
      assign-access-list 103
      map-template 103 2
```

Finally, you can free more Router 2 process time indicating the SA is not re-negotiated when this reaches the lifetime percentage specified and that the other end of the Tunnel (Router 1) will re-negotiate the SA.

```
HOST IPSec config>ADVANCED RENEGOTIATION-TIME 0
HOST IPSec config>
```

The resulting IPSec configuration is:

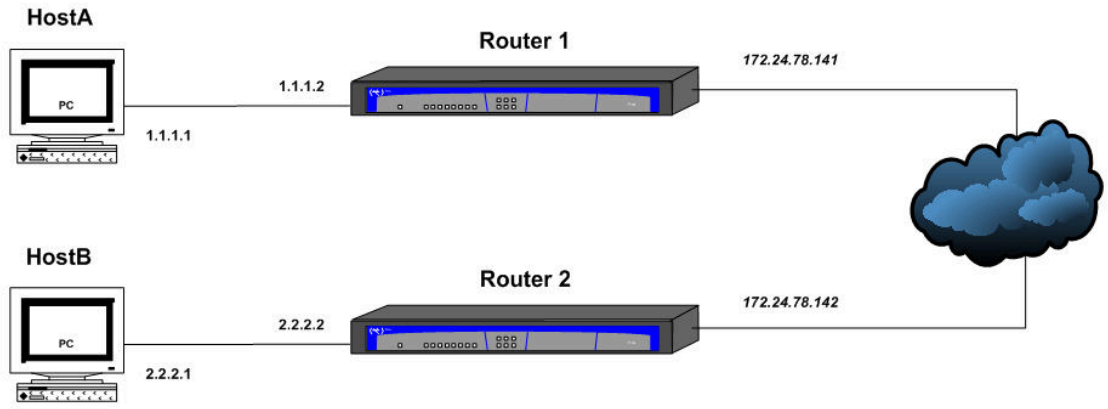
```
HOST IPSec config>LIST ALL
IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 103
  Templates: 2


Extended Access List 103, assigned to IPSec

1     PERMIT  SRC=172.24.0.0/16  DES=192.60.1.0/24  Conn:0

TEMPLATES
1 isakmp DES MD5  DES=0.0.0.0
   LifeTime:1h0m0s
   IKE AGGRESSIVE
   PRESHARED
   fqdn ID TYPE
   OAKLEY GROUP 1

2 dynamic ESP-DES ESP-SHA1  SRC=192.60.1.21 DES=0.0.0.0
   LifeTime:0h30m0s
   PFS disabled

1 key entries
   GAS1. ************
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 0

SA's purge timeout:  15

Use software exponentiation

HOST IPSec config>
```

The following is displayed on executing the “SHOW CONFIG” command:

```
HOST IPSec config>SHOW CONFIG
; Showing System Configuration ...
; Router CENTRIX SEC (c) 1 36 Version 10.0.0CAI

      enable
      assign-access-list 103
;
      template 1 default
      template 1 isakmp des md5
      template 1 ike mode aggressive
      template 1 ike idtype fqdn
;
      template 2 default
      template 2 dynamic esp des sha1
      template 2 source-address 192.60.1.21
      template 2 life duration seconds 1800
;
      map-template 103 2
      key preshared hostname GAS1. ciphered 0xE21C47018BC8B868FB72F48DC4363FC0CF
ABF60C9FFE0286
      advanced renegotiation-time 0
HOST IPSec config>
```

Communication between the routers will now be securely carried out, with the Pre-shared key as the only protected key in this case.

4.4. Example 4: Tunnel End-Point Discovery



This scenario shows how to use the TED functionality (Tunnel End-Point Discovery) in dynamic IPSec (IPSec IKE). To do this, two routers have been configured to open an IPSec tunnel between them as described in example 2, with the peculiarity that the IP address to use to open the tunnel (remote peer) has not been specified to either. When one of the hosts protected by a router wishes to communication with its remote end (e.g. Host A with B), the router uses the TED protocol to find its colleague and initiate ISAKMP negotiation.

a) Configuring Router 1

- *Configuring the hostname, addresses and IP rules*

The IP configuration for this example is very basic. We only need the addresses for the two networks connected by the router and a route to reach the network protected by the remote end:

```
*p 4


Config>p ip

-- Internet protocol user configuration --
IP config>address ethernet0/0 172.24.78.141 255.255.0.0
IP config>address ethernet0/0 1.1.1.2 255.255.255.0
IP config>route 2.2.2.0 255.255.255.0 172.24.78.142
IP config>exit
Config>
```

- *Creating the access control lists*

Once the IP parameters have been configured, you need to configure the access control lists. To do this, access the generic lists configuration menu, select a number from the list corresponding to an extended list (between 100 and 199, in our example this is 101), indicate an entry ID from the list (number 1 in this example) and set the required value for the following parameters:

- The source address of the packets you wish to "collide" with the access list; in this case the subnet with the clients going to be protected.
- The destination address of the packets to "collide", in this case the subnet with clients that protects the opposite router.
- The action to be executed in the packets, in this case IPSec process (PERMIT).

```
Config>feature access-lists

-- Access Lists user configuration --
Access Lists config>; -- Access Lists user configuration --
Access Lists config>access-list 101


Extended Access List 101>entry 1 default
Extended Access List 101>entry 1 permit
Extended Access List 101>entry 1 source address 1.1.1.0 255.255.255.0
Extended Access List 101>entry 1 destination address 2.2.2.0 255.255.255.240
Extended Access List 101>exit
Access Lists config>exit
Config>
```

- *Creating templates*

Now we need to create the ISAKMP and dynamic Templates. When creating the ISAKMP Template we use the "discover" option in order to specify that the TED must be used to discover which is the tunnel remote end. In the same way, on creating the dynamic Template, you should not indicate the tunnel remote address as this is still unknown and will be discovered through the TED process.

```
Config>protocol ip

-- Internet protocol user configuration --
IP config>ipsec

-- IPSec user configuration --
IPSec config>enable
IPSec config>;
IPSec config>template 1 default
IPSec config>template 1 isakmp tdes sha1
IPSec config>template 1 discover
IPSec config>template 1 life duration seconds 45m
IPSec config>;
IPSec config>template 3 default
IPSec config>template 3 dynamic esp tdes md5
IPSec config>template 3 source-address 172.24.78.141
IPSec config>template 3 life type both
```

```
IPSec config>template 3 life duration seconds 45m
IPSec config>;
IPSec config>key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
IPSec config>event address-filter 0.0.0.0 0.0.0.0
```

- *Creating the SPDs*

Finally the SPDs must be established, i.e. relate an access control to a created Template. In our example, the configured extended generic list must be assigned to IPSec and associated to a Template, 101, and the Template to be related is dynamic, i.e. ID 3.

```
IPSec config>assign-access-list 101
IPSec config>map-template 101 3
```

The IPSec configuration will be as follows:

```
IPSec config>list all


IPSec Access Control.
Access Control is: enabled
QOS Preclassify is: disabled

Access Lists assigned to IPSec:
  Extended Access List 101
  Templates: 3


Extended Access List 101, assigned to IPSec

1     PERMIT  SRC=1.1.1.0/24  DES=2.2.2.0/28  Conn:0

TEMPLATES
1 isakmp 3DES SHA1  DES=0.0.0.0
   LifeTime:0h45m0s
   IKE MAIN
   PRESHARED
   addr4 ID TYPE
   OAKLEY GROUP 1
   Tunnel End-point Discovery enabled

3 dynamic ESP-3DES ESP-MD5  SRC=172.24.78.141 DES=0.0.0.0
   LifeTime:0h45m0s 4608000 kbytes
   PFS disabled

1 key entries
   0.0.0.0 ************
0 rsakey entries
Id.           Date.         Len         CA.                   Cert sn.

Timer to check LDAP sessions not configured. Using default value: 30 seconds

KeepAlive Configuration:
   Maximum number of encoded packets without receiving an answer: 0.
   Timeout after last packet encoded: 0 seconds.

DPD Configuration:
Idle period(secs) before sending DPD keepalives: 60
Maximum number of DPD keepalives not acknowledged: 3
Period of time(secs) between DPD keepalives: 5
Always send keepalive after idle period expiration : ENABLED
Anti-replay : DISABLED

Check-out time (%) - from SA's end-lifetime - to renegotiate : 10
```

```
SA's purge timeout:  15

NAT Translation Port:  4500

Use hardware exponentiation (AVAILABLE)
```

The complete device configuration, obtained through "show config", is as follows:

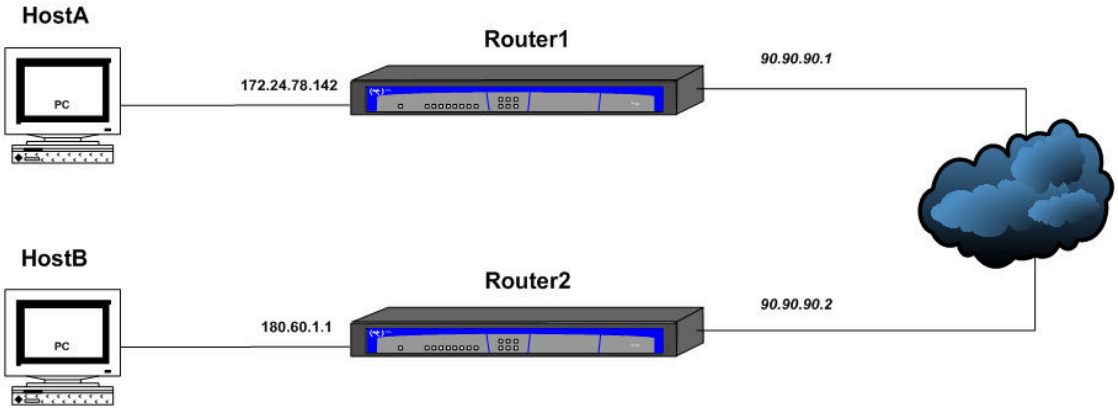
```
Config>show config
; Showing System Configuration ...
; ATLAS Router 2 8 Version 10.6.2-Alfa

log-command-errors
no configuration
set data-link x25 serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2
feature access-lists
; -- Access Lists user configuration --
   access-list 101
      entry 1 default
      entry 1 permit
      entry 1 source address 1.1.1.0 255.255.255.0
      entry 1 destination address 2.2.2.0 255.255.255.240
;
   exit
;
exit
;
;
protocol ip
; -- Internet protocol user configuration --
   address ethernet0/0 172.24.78.141 255.255.0.0
   address ethernet0/0 1.1.1.2 255.255.255.0
;
;
   route 2.2.2.0 255.255.255.0 172.24.78.142
;
;
;
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes sha1
      template 1 discover
      template 1 life duration seconds 45m
;
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 172.24.78.141
      template 3 life type both
      template 3 life duration seconds 45m
;
      map-template 101 3
      key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
      event address-filter 0.0.0.0 0.0.0.0
   exit
;
exit
;
dump-command-errors
end
; --- end ---
```

b) Configuring Router 2

Router 2 configuration is similar to Router 1 but differs as regards the source addresses and the access lists and templates destination. The configuration is as follows:

```
Config>show config
; Showing System Configuration ...
; ATLAS Router 2 8 Version 10.4.1-Alfa

log-command-errors
no configuration
feature access-lists
; -- Access Lists user configuration --
   access-list 101
      entry 1 default
      entry 1 permit
      entry 1 source address 2.2.2.0 255.255.255.240
      entry 1 destination address 1.1.1.0 255.255.255.0
   exit
exit
;
protocol ip
; -- Internet protocol user configuration --
   address ethernet0/0 172.24.78.142 255.255.0.0
   address ethernet0/0 2.2.2.2 255.255.255.0
;
   route 1.1.1.0 255.255.255.0 172.24.78.141
;
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes sha1
      template 1 discover
      template 1 life duration seconds 4m
;
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 172.24.78.142
      template 3 life type both
      template 3 life duration seconds 6m
;
      map-template 101 3
      key preshared ip 0.0.0.0 ciphered 0xD8599397F3F05E04A00A56234D376BCD
      event address-filter 0.0.0.0 0.0.0.0
   exit
exit
;
dump-command-errors
end
; --- end ---
```

4.5. Example 5: Permanent Tunnel



This scenario shows how to configure the devices so a permanent tunnel is created between them. For this example, we are going to use a configuration similar to the one seen in example 2 (dynamic IKE) and add the commands needed to ensure the tunnel remains open.

a) Configuring Router 1

- *Configuring IP, Lca, templates and SPDs*

We are going to start with a basic dynamic IPSec configuration with two routers protecting both subnets. The most important command here is the "advanced renegotiation-time100" which allows the tunnel to renegotiate even if there is no traffic.

```
; Showing System Configuration ...
; ATLAS Router 2 8 Version 10.6.2-Alfa

log-command-errors
no configuration
set data-link x25 serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2
feature access-lists
; -- Access Lists user configuration --
   access-list 101
      entry 1 default
      entry 1 permit
      entry 1 source address 172.24.0.0 255.255.0.0
      entry 1 destination address 180.60.0.0 255.255.0.0
;
   exit
;
exit
;
protocol ip
; -- Internet protocol user configuration --
   address ethernet0/0 172.24.78.142 255.255.0.0
   address ethernet0/0 90.90.90.1 255.255.255.0
;
;
   route 0.0.0.0 0.0.0.0 90.90.90.2
;
   classless
```

```
;
;
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes sha1
      template 1 destination-address 90.90.90.2
;
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 90.90.90.1
      template 3 destination-address 90.90.90.2
;
      map-template 101 3
      key preshared ip 90.90.90.2 ciphered 0xD8599397F3F05E04A00A56234D376BCD
      advanced renegotiation-time 100
   exit
;
exit
;
dump-command-errors
end
; --- end ---
```

Additionally we are going to use the proprietor TIDP protocol to ensure that there is traffic through the tunnel which will open in cases where the device reboots or when during negotiation there is no communication between the tunnel ends. What we do is to configure the device so it periodically sends a discovery packet whose source and destination IPs "collide" with the access list used in IPSec (A packet has been configured to be sent every two minutes in this example. This time is more than sufficient to allow the tunnel to open when the device restarts).

```
Config>feature ip-discovery

-- Teldat IP Discovery Protocol configuration --
TIDP config>discovery-station 1 ip 180.60.1.1
TIDP config>discovery-station 1 source ip 172.24.78.142
TIDP config>discovery-station 1 timer 2m
TIDP config>exit
Config>
```

b) Configuring Router 2

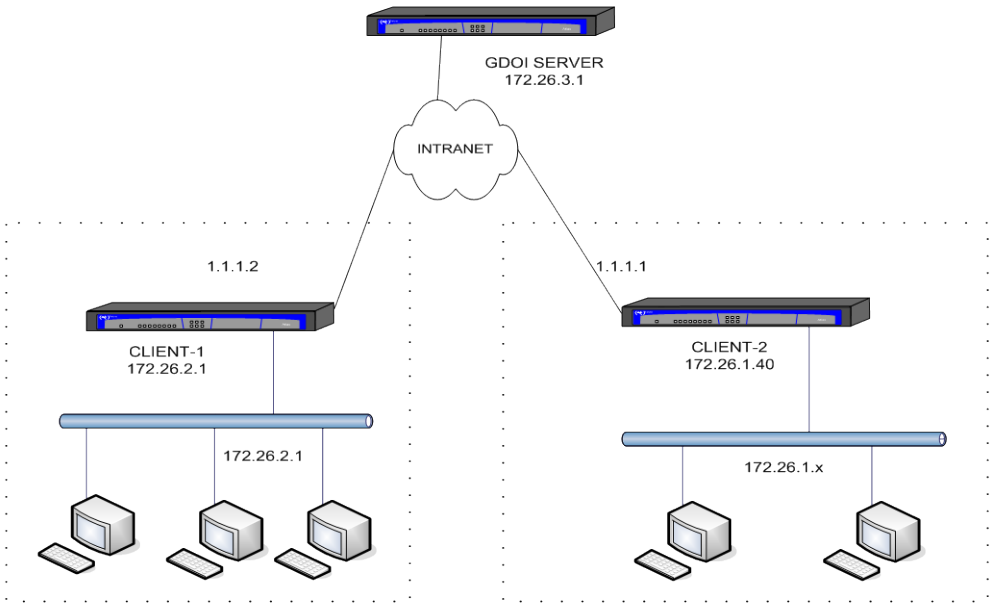
Router 2 configuration is similar to Router 1 but differs as regards the source addresses and the access lists and templates destination and does not require any renegotiation time configuration (it's the other end that opens the tunnel) or for TIDP. The configuration therefore is as follows:

```
; Showing System Configuration ...
; ATLAS Router 2 8 Version 10.6.3-Alfa

log-command-errors
no configuration
set data-link x25 serial0/0
set data-link x25 serial0/1
set data-link x25 serial0/2
feature access-lists
; -- Access Lists user configuration --
   access-list 101
      entry 1 default
      entry 1 permit
```

```
      entry 1 source address 180.60.0.0 255.255.0.0
      entry 1 destination address 172.24.0.0 255.255.0.0
;
   exit
;
exit
;
;
protocol ip
; -- Internet protocol user configuration --
   address ethernet0/0 90.90.90.2 255.255.255.0
   address ethernet0/0 180.60.1.1 255.255.0.0
;
;
   route 0.0.0.0 0.0.0.0 90.90.90.1
;
   classless
;
;
   ipsec
; -- IPSec user configuration --
      enable
      assign-access-list 101
;
      template 1 default
      template 1 isakmp tdes sha1
      template 1 destination-address 90.90.90.1
;
      template 3 default
      template 3 dynamic esp tdes md5
      template 3 source-address 90.90.90.2
      template 3 destination-address 90.90.90.1
;
      map-template 101 3
      key preshared ip 90.90.90.1 ciphered 0xD8599397F3F05E04A00A56234D376BCD
   exit
;
exit
;
dump-command-errors
end
; --- end ---
```

4.6. Example 6: GDOI

This scenario shows how to configure the devices to use the GDOI protocol to negotiate the IPSec keys and the encryption policies. We want to encrypt traffic between locations 172.26.1.x and 172.26.2.x using triple des and sha1. To do this, configure a server located at a central point in the network and two clients, located at each location.



a) Configuring the server

The server configuration consists of an access list, list number 100 which permits all traffic between the network 172.26.0.0. This list is assigned to IPSec and subsequently associated to the SA 1 of the GDOI group being used. This SA 1 specifies triple des encryption and sha1 authentication with the keys being refreshed every 5 minutes.

The rekey method used is unicast encryption with aes 256 bits, refreshing the keys every 10 minutes. For rekey message authentication, a public key known as MYKEY is used which needs to be previously generated:

```
IPSec config$key rsa generate MYKEY 512
```

You also need to configure an isakmp template for phase 1 negotiation with the GDOI clients, as well as a pre-share key which is needed for this negotiation.

The GDOI server configuration looks like this:

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS100 NOE Router 2 250 Version 10.8.0-Alfa

   log-command-errors
   no configuration
   add device x25 1
   set data-link frame-relay serial0/0
   set data-link sync serial0/1
   set data-link sync serial0/2
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 172.26.0.0 255.255.0.0
         entry 1 destination address 172.26.0.0 255.255.0.0
;
      exit
;
   exit
```

```
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.3.1 255.255.255.0
;
   exit
;
   protocol ip
; -- Internet protocol user configuration --
      internal-ip-address 172.26.3.1
;
      route 0.0.0.0 0.0.0.0 172.26.3.2
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes md5
;
         key preshared ip 172.26.2.1 ciphered 0x0DD598B4F74E201E
         key preshared ip 172.26.1.40 ciphered 0x0DD598B4F74E201E

         gdoi group 2
; -- GDOI user configuration --
            identity number 2
            rekey transport unicast
            rekey algorithm aes-256
            rekey authentication rsa MYKEY
            rekey lifetime seconds 10m
            rekey retransmit 10s number 3
            sa ipsec 1
               lifetime 5m
               match address ipv4 100
               transform-set tdes sha1
            exit
;
         exit
;
      exit
;
   exit
;
;
;
   dump-command-errors
   end
```

b) Configuring client 1

In order to provide IP connectivity, you need to configure the PPP interface addresses as well as the Ethernet address complying with the schema drawn up at the beginning of this example. The PPP interface IP is 1.1.1.2 and Ethernet is 172.26.2.1.

You need to configure an access list to associate it to the dynamic template. This access list consists of one or more deny entries to select a specific traffic pattern over which IPSec is not executed, even though the downloaded lists from the server indicate this, and a final permit entry. As in this example we want to encapsulate all the traffic between network 172.26.1.0/24 and 172.26.2.0/24 without exception, the access list only contains the permit entry.

You need to configure an isakmp template and a pre-share key to be used in the phase 1 negotiations with the server, as well as a dynamic template with the selected GDOI group, in this case 2. The destination IP for the isakmp template and for the dynamic must be the GDOI server's IP.

Client 1 configuration looks like this:

```
; Showing Menu and Submenus Configuration for access-level 15 ...
; ATLAS Router 2 156 Version 10.8.0-Alfa

   log-command-errors
   no configuration
   add device ppp 1
   set data-link sync serial0/0
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
;
      exit
;
   exit
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      speed 256000
   exit
;
;
;
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.2.1 255.255.255.0
;
;
;
;
;
;
   exit
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address 1.1.1.2 255.255.255.0
;
;
;
;
;
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 172.26.1.0 255.255.255.0 1.1.1.1
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 dynamic esp tdes sha1
         template 1 source-address 172.26.2.1
         template 1 destination-address 172.26.3.1
         template 1 mapped-to-ifc ppp1
         template 1 gdoi group 2
;
         template 2 default
         template 2 isakmp tdes md5
```

```
         template 2 destination-address 172.26.3.1
;
         map-template 100 1
         key preshared ip 172.26.3.1 ciphered 0x0DD598B4F74E201E
      exit
;
   exit
;
;
;
   dump-command-errors
end
```

c) Configuring client 2

Client 2 configuration is similar to client 1 except that the IP addresses for the PPP interface and for the Ethernet7 interface are different. The configuration looks like this:

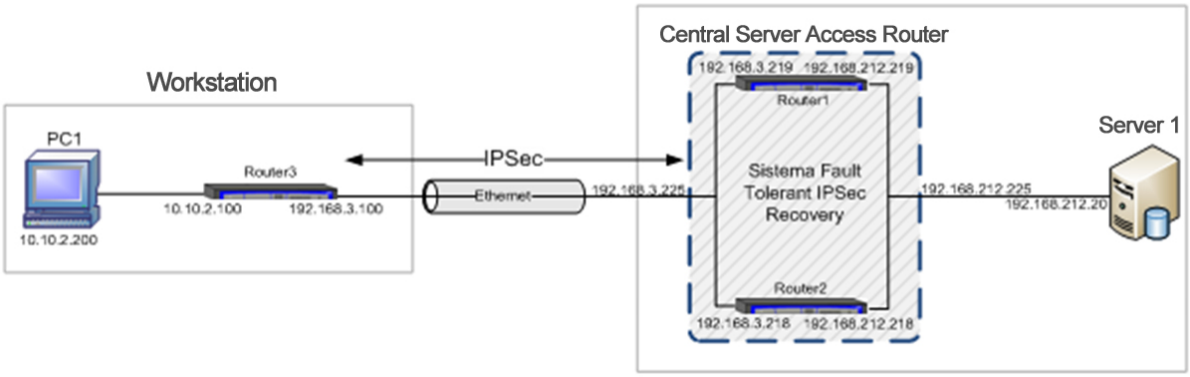
```
; Showing Menu and Submenus Configuration for access-level 15 ...
; Super Router * * Version 10.8.0-Alfa

   log-command-errors
   no configuration
   add device ppp 1
   set data-link sync serial0/0
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 172.26.1.40 255.255.255.0
;
;
;
;
;
   exit
;
;
;
   network serial0/0
; -- Interface Synchronous Serial Line. Configuration --
      speed 512000
   exit
;
;
;
   network ppp1
; -- Generic PPP User Configuration --
      ip address 1.1.1.1 255.255.255.0
;
;
;
;
;
      base-interface
; -- Base Interface Configuration --
         base-interface serial0/0 link
;
      exit
;
   exit
```

```
;
   protocol ip
; -- Internet protocol user configuration --
      internal-ip-address 172.26.1.40
;
      route 172.26.2.0 255.255.255.0 1.1.1.2
;
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 dynamic esp tdes sha1
         template 1 source-address 172.26.1.40
         template 1 destination-address 172.26.3.1
         template 1 mapped-to-ifc ppp1
         template 1 gdoi group 2
;
         template 2 default
         template 2 isakmp tdes md5
         template 2 destination-address 172.26.3.1
;
         template 3 default
         template 3 isakmp tdes md5
;
         map-template 100 1
         key preshared ip 172.26.3.1 ciphered 0x0DD598B4F74E201E
      exit
;
   exit
;
;
;
   dump-command-errors
   end
```

4.7. Example 7: Fault Tolerant IPSec Recovery

This scenario shows how to configure the Fault Tolerant IPSec Recovery system to protect the IPSec sessions when faced with failures in the tunnel terminator device.

We're assuming that a PC in a workstation, PC1, needs to connect to a central server, Server1. To do this we use a router, Router3, which establishes an IPSec session between the workstation and the access router to the central server. To make the connection more robust, we have implanted a Fault Tolerant IPSec Recovery system in the central server's access router, converting this router into two, Router1 and Router2.



a) Configuring the router in the workstation, Router3

The configuration of Router3 doesn't differ conceptually from that in example2, where the tunnel's end address is the IP address that the central server's access router offers 192.168.3.225 to the exterior. This is the final result.

```
   log-command-errors
   no configuration
   set hostname Router3
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 2 default
         entry 2 permit
         entry 2 source address 10.10.2.0 255.255.255.0
         entry 2 destination address 192.168.212.0 255.255.254.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.100 255.255.255.0
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 10.10.2.100 255.255.255.0
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 0.0.0.0 0.0.0.0 192.168.3.225
;
      classless
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes sha1
         template 1 destination-address 192.168.3.225
         template 1 keepalive dpd
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 destination-address 192.168.3.225
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
      exit
;
   exit
;
;
   dump-command-errors
   end
```

b) Configuring the access router to the central server, Router1 and Router2

The access router to the central server is made up of two routers, Router1 and Router2, which have the Fault Tolerant IPSec Recovery system implanted between them. This system requires various protocols to be configured: IPSec, IPSecFT and VRRP. These are further explained below.

- *Configuring IPSec*

The IPSec configuration for both routers must be identical and is similar to that shown in example 2. We are going to start therefore from a similar configuration in IPSec:

```
RouterX Config>show con
[…]
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
[…]
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      classless
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes sha1
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
;
      exit
;
   exit
[…]
RouterX Config>
```

However, there is a very important configuration parameter in the dynamic templates. This parameter indicates which templates form part of the Fault Tolerant IPSec Recovery system i.e. which IPSec sessions remain when a fault occurs in the device that they are established with. In this case we have a single dynamic template, template 2, which we are going to apply the fault tolerance.

```
RouterX IPSec config>template 2 fault-tolerant
RouterX IPSec config>
```

This is the resulting configuration both for Router1 as well as for Router2:

```
RouterX Config>show con
[…]
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
[…]
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      classless
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes sha1
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
;
      exit
;
   exit
[…]
RouterX Config>
```

- *Configuring IPSecFT*

IPSecFT is the protocol used to maintain an IPSec sessions database so when one of the devices goes down, the other can take over.

In order to configure IPSecFT, we need to first decide which device is going to act as master and which as slave. This choice makes no difference to the system operations so we are going to use Router1 for example as the master router.

IPSecFT in Router1

Beginning with Router1, the first thing we need to do is to access the IPSecFT submenu.

```
Router1 Config>protocol ip

-- Internet protocol user configuration --
Router1 IP config>ipsec

-- IPSec user configuration --
Router1 IPSec config>fault-tolerant

-- Fault tolerant IPSec recovery user configuration --
Router1 IPSecFT config>
```

This subsystem is by default in slave mode and disabled. You need to change the mode so it can act as master, and before enabling it, you must configure the necessary parameters so it operates correctly.

These parameters are the IP address for the slave and the source for the packets. In this case the slave address is 192.168.212.218 and the source is IP 192.168.212.219.

It's very important that neither the source address nor the destination in IPSecFT coincide with the virtual IP address managed in VRRP.

```
Router1 IPSecFT config>mode master
Router1 IPSecFT config>slave-address 192.168.212.218
Router1 IPSecFT config>source-address 192.168.212.219
Router1 IPSecFT config>enable
Router1 IPSecFT config>show menu
; Showing Menu Configuration for access-level 15 ...

            mode master
            slave-address 192.168.212.218
            source-address 192.168.212.219
            enable
Router1 IPSecFT config>
```

Other parameters such as the port are optional and in this example we are going to leave them with their default values.

IPSecFT in Router2

Going to Router2, we need to access the IPSecFT submenu.

```
Router2 Config>protocol ip

-- Internet protocol user configuration --
Router2 IP config>ipsec

-- IPSec user configuration --
Router2 IPSec config>fault-tolerant

-- Fault tolerant IPSec recovery user configuration --
Router2 IPSecFT config>
```

As this device is going to act in slave mode, we don't need to configure anything, the only parameter that we can configure is the listen port, and in this case we are going to leave its default value in the same way as we did in the master device. Here we simply need to enable the protocol.

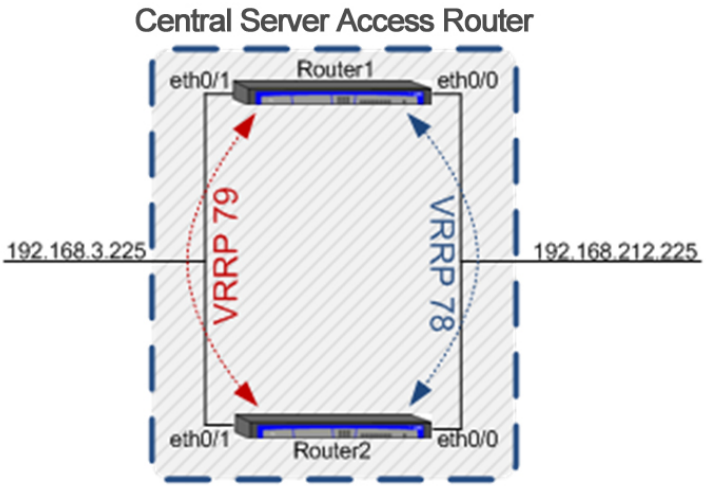
```
Router2 IPSecFT config>enable
Router2 IPSecFT config>show menu
; Showing Menu Configuration for access-level 15 ...

            enable
Router2 IPSecFT config>
```

With this configured, the protocol is capable of establishing the appropriate sessions between Router1 and Router2 and to maintain a database with the sessions that IPSec has established in each device.

- *Configuring VRRP*

Here you need to configure VRRP between the two devices that make up the access router to the main server so that this appears to be the only router to the exterior.

VVRP is the protocol that decides which device has the IP address that the IPSec sessions are established with at any point (VRRP with VRID 79) and additionally is the protocol that decides which device communicates with the central server (VRRP with VRID 78). I.e. in this example, we need to configure two VRRPs in each device, one for each interface.



The VRRP configuration is further explained in manual "Dm759-I VRRP Protocol", however there are certain specific requirements so it can be used with Fault Tolerant IPSec Recovery. These are explained below:

standby-delay

The first parameter is the *standby-delay*. This parameter is related to the time that the device takes in taking over the virtual IP address when it is ready for this. This parameter is important because the fact of taking over the virtual IP address is a process that must occur subsequent to establishing the IPSecFT session. This occurs in this order because it's the VRRP that tells the IPSecFT when it must take over the IPSec sessions and the IPSecFT is not prepared to do this until it is established.

For this example the *standby-delay* parameter is left with its default values: 10 seconds.

reload-delay

Another parameter to bear in mind is the *reload-delay*. This parameter indicates the time the protocol waits before initiating from device start-up. This parameter must take different values in both devices, but the same value for the interfaces within the same router; the aim is to prevent a device when initiating from taking over both virtual IP addresses at the same time as there is another device that in turn is doing the same.

In this example, we have configured the *reload-delay* parameter with a value of 40 seconds for the two interfaces in Router2 and with a value of 30 seconds (default value) for the two interfaces in Router1.

Preemption Mode

We suggest that the preempt mode is left disabled so reducing changes in the virtual IP address proprietor and as a consequence the number of times that the IPSec session change from one device to another.

```
ip vrrp XX no-preempt
```

VRRP Priorities

Another consideration is adjusting the VRRP priorities to a higher value. The reason for this is that the time the VRRP takes to react depends on this priority (the higher the priority, the shorter response time) and can reach 1 second. We suggest using values greater than 250.

Priority cost in tracking

Finally, we need to remember that both virtual IP addresses must pass from one device to another simultaneously. This is achieved by applying in the interfaces of the same device a track that reports when one of the interfaces stops operating and in this case, forces the rejection of any virtual IP address that it is managing. The way to do this is to apply a priority cost equal to the VRRP priority to the tracking, i.e. the priority less the priority cost is 0.

Once these considerations have been executed, we need to configure the two VRRPs in each device.

As already presented in the schema above, we are going to configure a VRRP in each device one with VRID 78 and the other with VRID 79. The 78 is configured in the ethernet0/0 interfaces in the devices and manage IP address 192.168.212.225, and the 79 configured in the ehternet0/1 interfaces managing IP address 192.168.3.225.

A priority of 254 is assigned to Router1 and 253 to Router2.

A track is configured in each interface in both devices that monitors the state of the other interface within the same device. In cases when one drops a value to its priority equal to its priority is subtracted so it passes to 0.

```
   network ethernet0/0
[…]
      ip vrrp VRID1 priority PRIO
      ip vrrp VRID1 track interface ethernet0/1 prio-cost PRIO
;
   exit
;
   network ethernet0/1
[…]
      ip vrrp VRID2 priority PRIO

      ip vrrp VRDI2 track interface ethernet0/0 prio-cost PRIO
```

In both devices an advertise-interval is configured for the VRRP messages equal to 100 milliseconds.

```
ip vrrp XX advertise-interval 100 msec
```

The configuration relative to the VRRP in each device is shown below.

VRRP in Router1

```
Router1 Config>show config
[…]
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.219 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 254
      ip vrrp 78 track interface ethernet0/1 prio-cost 254
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.219 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 254
      ip vrrp 79 track interface ethernet0/0 prio-cost 254
;
   exit
[…]
Router1 Config>
```

VRRP in Router2

```
Router2 Config>show config
[…]
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.218 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 253
      ip vrrp 78 reload-delay 40s
      ip vrrp 78 track interface ethernet0/1 prio-cost 253
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.218 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 253
      ip vrrp 79 reload-delay 40s
      ip vrrp 79 track interface ethernet0/0 prio-cost 253
;
   exit
[…]
Router2 Config>
```

- *Full configuration*

Once each section on the configuration has been broken down, the full configuration in Router1 and Router2 is shown:

Router1 Configuration

```
   log-command-errors
   no configuration
   set hostname Router1
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.219 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 254
      ip vrrp 78 track interface ethernet0/1 prio-cost 254
;
```

```
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.219 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 254
      ip vrrp 79 track interface ethernet0/0 prio-cost 254
;
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      classless
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes sha1
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         fault-tolerant
; -- Fault tolerant IPSec recovery user configuration --
            mode master
            slave-address 192.168.212.218
            source-address 192.168.212.219
            enable
         exit
;
      exit
;
   exit
;
;
   dump-command-errors
   end
```

Router2 Configuration

```
   log-command-errors
   no configuration
   set hostname Router2
   feature access-lists
; -- Access Lists user configuration --
      access-list 100
         entry 1 default
         entry 1 permit
         entry 1 source address 192.168.212.0 255.255.254.0
         entry 1 destination address 10.10.2.0 255.255.255.0
;
      exit
```

```
;
   exit
;
;
   network ethernet0/0
; -- Ethernet Interface User Configuration --
      ip address 192.168.212.218 255.255.254.0
;
      ip vrrp 78 ip 192.168.212.225
      ip vrrp 78 advertise-interval 100 msec
      ip vrrp 78 no-preempt
      ip vrrp 78 accept-vip-packets
      ip vrrp 78 priority 253
      ip vrrp 78 reload-delay 40s
      ip vrrp 78 track interface ethernet0/1 prio-cost 253
;
   exit
;
;
   network ethernet0/1
; -- Ethernet Interface User Configuration --
      ip address 192.168.3.218 255.255.255.0
;
      ip vrrp 79 ip 192.168.3.225
      ip vrrp 79 advertise-interval 100 msec
      ip vrrp 79 no-preempt
      ip vrrp 79 accept-vip-packets
      ip vrrp 79 priority 253
      ip vrrp 79 reload-delay 40s
      ip vrrp 79 track interface ethernet0/0 prio-cost 253
;
      input-buffers 1024
   exit
;
;
   protocol ip
; -- Internet protocol user configuration --
      route 10.10.2.0 255.255.255.0 192.168.3.100
;
      classless
      ipsec
; -- IPSec user configuration --
         enable
         assign-access-list 100
;
         template 1 default
         template 1 isakmp tdes sha1
         template 1 keepalive dpd
         template 1 send-original-pkt
;
         template 2 default
         template 2 dynamic esp tdes md5
         template 2 source-address 192.168.3.225
         template 2 fault-tolerant
;
         map-template 100 2
         key preshared ip 0.0.0.0 ciphered 0x12D942B46B48645B
         fault-tolerant
; -- Fault tolerant IPSec recovery user configuration --
            enable
         exit
;
      exit
;
   exit
;
;
   dump-command-errors
   end
```

5. Certificates

When the authentication methods based on RSA are applied, you need to use RSA asymmetric keys. These keys are usually used within the higher layer encapsulations known as *Certificates*. The Teldat Routers permit authentication based on RSA and require tools that are capable to manager Certificates for this.

The device will only use the certificates if they are correctly signed, are valid and are not revoked.

Even if all these requirements are not fulfilled, the device does allow the certificates to be downloaded and display their properties although they cannot be used.

Given that the fact the certificates must be within their period of validity (mandatory) in order to be used, it's vital that the time in the device is correctly synchronized and that time zone parameters and the summer time changes, if this is required, are correctly configured.

We are going to describe how to operate with Certificates in this section, i.e. how to load them, how to assign them to Templates, how to create them, etc.

5.1. CERT Menu

The CERT menu is located within the IPSec menu. In the CERT menu you will find the CERTIFICATE command which has the following options:

Command	Operation
LOAD	Loads a CERTIFICATE from a disk to RAM memory.
DISK_DELETE	Deletes a CERTIFICATE from a disk.
CONFIG_DELETE	Deletes a CERTIFICATE from the configuration.
PRINT	Displays the content of a CERTIFICATE on screen.
BASE64	Loads a CERTIFICATE from the console in base64 format.
NO	Disables or deletes an option.

"CERTIFICATE [CertFile] LOAD"

This command permits you to load a Certificate from a disk to the device RAM memory. Before executing an operation with a Certificate, Certificate must be loaded in the RAM through this command.

Example:

```
CERTIFICATES config>certificate router.cer load
```

"CERTIFICATE [CertFile] DISK_DELETE"

This command permits you to delete a Certificate from a disk. The certificates can be loaded from a file saved in the disk or from the router configuration using the FILE command.

Example:

```
CERTIFICATES config>certificate router.cer disk_delete
```

"CERTIFICATE [CertFile] CONFIG_DELETE"

Through this command you can delete a Certification from the configuration. The certificates can be loaded from a file saved in the disk or from the router configuration using the FILE command.

Example:

```
CERTIFICATES config>certificate router.cer config_delete
```

"CERTIFICATE [CertFile] PRINT"

This command permits you to print the content of a previously loaded Certificate.

Example:

```
CERTIFICATES config>certificate router.cer print
Version                   : V3
Serial Number             : 547E D185 0000 0000 1E6E
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ): SECTESTCA1
  OU (Organizational Unit): Microsoft, Interopability Testing Only
  O  (Organization Name  ): Microsoft, Interopability Testing Only
  L  (Locality           ): Redmond
  S  (State or Province  ): WA
  C  (Country Name       ): US
  E  (Email              ): testca@microsoft.com
Valid From                : Wed Jul 25 09:21:24 2001
Valid To                  : Thu Jul 25 09:31:24 2002
Subject:
  E  (Email              ): jiglesias@teldat.es
  CN (Common Name        ): router.teldat.es
  OU (Organizational Unit): ImasD
  O  (Organization Name  ): Teldat
  L  (Locality           ): Tres Cantos
  S  (State or Province  ): Madrid
  C  (Country Name       ): sp
Publick Key               :
  Algorithm Identifier    : RSA
  Modulus Length          : 512 Bits.
  Modulus                 :
    E1CF D175 90EE 43BC 4BC5 D215 695A 74CC D1E8 F301 4F09 2093 7B12 84C0
    2C07 DE4B E458 9D48 43CB 4F14 A075 0D09 FB57 71DB 4FC6 8FDF 1FEF AA6D
    13BB 96FB 88FA 1343
  Exponent                :
    01 00 01
Signature                 :
  Signature Algorithm     : SHA1 With RSA
  Signature Data Info     : 2048 Bits.
  Signature Data          :
    3C10 94F3 CE87 0040 C3D0 A59F 1F0E 84DC E21F CCFD CA7A 2A32 651B 3D27 F9D0
    F87A 6993 E22C 28F5 7954 ED49 1E90 A52C 8098 F686 5E51 18DA D713 D65E 81BB
    267A 1D70 957D FB2F C841 E155 AD3C 3B38 6796 FA62 F6EF 8D76 DEDF 09B2 52C3
    3496 AD4B BF06 1415 3111 DEDD B2BE 9C68 5584 0A3B BF41 90B3 05C4 5CA1 E079
```

```
    AADA 43B1 F48D 9DEE 9793 907E 262D 2CC5 325C F3D1 892C 54E7 4736 06A3 4883
    A239 B68D 5477 13A8 BDE0 D7F4 18C1 FD94 3116 48FC C701 BA86 D932 A5C8 C28C
    5FE0 D8CF BE39 CF77 5CCC A104 0189 FF0B 5598 DBB1 2EB5 6269 9683 31DF 19BB
    DDEB 8BC0 FFDA 4587 13E4 42FF 7AF1 BD63 ACE4 D469 37B7 03FA 78DD 4535 49FB
    36AA 4525 F6EF 33A8 F5DB 3934 5079 A536
```

"CERTIFICATE [CertFile] BASE64"

Through this command you can introduce a certificate in base64 format. Once this command has been executed without errors, the certificate is saved in the configuration and is displayed as the sequence of the FILE command.

Example:

```
CERTIFICATES config>cert wiscon base64

Introduce the Certificate (Base 64 format)
Enter <cr> to escape
-----BEGIN CERTIFICATE-----
MIIC6zCCAlSgAwIBAgICAlQwDQYJKoZIhvcNAQEEBQAwgakxCzAJBgNVBAYTAlVT
MRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlzb24xIDAeBgNVBAoT
F1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJEaXZpc2lvbiBvZiBJ
bmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBLSSBNYXN0ZXIgQ0Eg
LS0gMjAwMjA3MDFBMB4XDTAyMDYzMDIyMTYzOVoXDTI5MTExNjIyMTYzOVowgakx
CzAJBgNVBAYTAlVTMRIwEAYDVQQIEwlXaXNjb25zaW4xEDAOBgNVBAcTB01hZGlz
b24xIDAeBgNVBAoTF1VuaXZlcnNpdHkgb2YgV2lzY29uc2luMSswKQYDVQQLEyJE
aXZpc2lvbiBvZiBJbmZvcm1hdGlvbiBUZWNobm9sb2d5MSUwIwYDVQQDExxIRVBL
SSBNYXN0ZXIgQ0EgLS0gMjAwMjA3MDFBMIGfMA0GCSqGSIb3DQEBAQUAA4GNADCB
iQKBgQDJ3FDZym9Ja94DP7TUZXf3Vu3CZwqZzYThgjUT2eBJBYVALISSJ+RjJ2j2
CYpq3wesSgWHqfrpPnTgTBvn5ZZF9diX6ipAmC0H75nySDY8B5AN1RbmPsAZ51F9
7Eo+6JZ59BFYgowGXyQpMfhBykBSySnvnOX5ygTCz20LwKkErQIDAQABoyAwHjAP
BgNVHRMBAf8EBTADAQH/MAsGA1UdDwQEAwIBpjANBgkqhkiG9w0BAQQFAAOBgQB1
8ZXB+KeXbDVkz+b2xVXYmJiWrp73IOvi3DuIuX1n88tbIH0ts7dJLEqr+c0owgtu
QBqLb9DfPG2GkJ1uOK75wPY6XWusCKDJKMVY/N4ec9ew55MnDlFFvl4C+LkiS2YS
Ysrh7fFJKKp7Pkc1fxsusK+MBXjVZtq0baXsU637qw==
-----END CERTIFICATE-----
```

5.2. KEY RSA Command

This command enables you to work with the RSA keys generated in the Router.

Command	Operation
GENERATE	Generates a pair of random RSA keys.
CA-CHANGE	Changes the CA associated to the generated RSA key.

"KEY RSA GENERATE [CA NAME][SIZE(512/1024/2048)]"

Through this command you can generate a random RSA key and associate a CA name. I.e. generate a pair of public and private keys which are stored in the device disk on saving the configuration.

After generating the pair of keys, the device will suggest generating a CSR, *Certificate Signing Request*, indicating that the user should go to the CSR menu located in the CERT menu and execute the MAKE [RSA Key ID] command.

Example:

```
IPSec config>key rsa generate caname 512
RSA Key Generation.
Please, wait for a few seconds.
 RSA Key Generation done.
Checking..OK
Key Generation Process Finished. RSA Key ID: 1
Do not forget to save RSA keys.

It's a good moment to make the Certificate Signing Request (CSR) associated with
this RSA Key...If you want to do it, go to the CERT menu and then to the CSR menu
and execute the command MAKE 1.

IPSec config>
```

IMPORTANT NOTE: The generated RSA keys are saved in the device configuration but are not displayed when the "show configuration" command is used for security reasons. This means that if you execute the "show configuration" command and copy the displayed text to modify it and paste it in another device, all the configuration is copied with the exception of the generated RSA keys.

"KEY RSA CA-CHANGE"

This command permits you to change the CA associated with a previously generated RSA key.

Example:

```
IPSec config>list key rsa all
1 rsakey entries
Id.           Date.         Len         CA.                  Cert sn.
   1   06/18/03  11:46:16    512         caname               ---
atlaslocal IPSec config>key rsa ca-change 1 newca
Do not forget to save RSA keys changes.
IPSec config>lis key rsa all
1 rsakey entries
Id.           Date.         Len         CA.                  Cert sn.
   1   06/18/03  11:46:16    512           newca              ---
```

5.3. Obtaining certificates through CSR

You can obtain a certificate for a Teldat device by creating a Certificate Signing Request (CSR). The end objective is to achieve two files: the CA certificate *caname.cer*, and the Router one, *router.cer* (in cases where this isn't the root CA, you will have to install all the certificates from the path up to the root CA). The steps to carry out are as follows:

- Configure the values for the different CSR attributes, i.e. the *Subject-name* (**mandatory**), the *Alternative-name* (optional) and the *Password* (optional), in addition to selecting the *Signature Algorithm* used to create the certificate. To carry out this configuration, you need to access the CSR configuration menu and execute the respective commands explained in the following section "CSR Menu":
- *subject-name ["C=Country, L=Locality, ST=State or Province, O=Organization, OU=Organization Unit, CN=Common Name, E=Email ... "]*. This attribute is mandatory to generate the CSR.
- *alt-name [Alternative name]*. This attribute is optional.
- *password ciphered/plain [Password]*. This attribute is also optional.

- *Signature-algorithm MD5/SHA1/MD2*. The MD5 algorithm is used by default.

2. If you have a private key generated, you must create a CSR associated to this key. In order to do this, you execute the *make* command followed by the identifier for the said key from the CSR configuration menu (MAKE [RSA Key ID]). If you do not have a private key generated, you need to generate it (*key rsa generate* command). The private key will have a CA associated through a file name corresponding to the certificate installed in the device for this CA, *caname.cer*. (This operation can be carried out even if you do not have a CA certificate available.)
3. After generating the CSR you can save this in a file that later can be obtained through FTP or be printed through the console by executing the *print* command. Normally the CSR are encoded in base64.
4. The CSR must be delivered to the CA in order for a certificate is returned, *router.cer*. Normally at this point, the CA also sends a certificate from the CA itself, *caname.cer*.
5. The obtained certificates are installed in the device, sending them through FTP and executing the *quote site savebuffer* command, or by using the CERTIFICATE [CertFile] BASE64 command.
6. A template is created which will use the RSA method *template **1** ike method rsa.*
7. Finally the CA certificate is associated to the template being used, through the command *template **1** ike ca **caname***.
8. The last step is to save the configuration.

This means, the association between the components is as follows:

- **(Private Key, CSR)** = Association through the private key identifier.
- **(Private Key, CA)** = Association through the CA name.
- **(Private Key, Certificado de Equipo)** = Association through the CA and the certificate serial number. The CA must be associated to a template and the certificate must be loaded.

NOTE: Verisign does not admit certain characters in the CSR fields. The @ symbol is one of these, so an email address cannot be included. The error returned by Verisign is 105. This field must be left blank if the CSR is going to be delivered to Verisign.

The *list template all* command displays how everything has gone:

```
IPSec config>list template all
TEMPLATES
1 isakmp 3DES MD5  DES=1.1.1.1
   LifeTime:1h0m0s
   IKE MAIN
   RSA SIGNATURE
      CA      : SECTEST.CER. Expired.
      CRL     : disabled
       USER    : ROUTER.CER. Signature ok. Expired. Without Private Key.
   fqdn ID TYPE
   OAKLEY GROUP 1
```

5.4. CSR Menu

The CSR command is located in the CERT menu and is used to access the CSR (Certificate Signing Request) configuration menu. This latter menu contains the commands used to create the CSRs and the configuration for their different attributes:

Command	Function
ALT-NAME	Configures the CSR *Alternative-name* attribute.
CLEAN	Deletes the CSR from the RAM memory.
DELETE	Deletes the CSR file stored in a disk.
LIST	Displays the CSR files stored in a disk on the screen.
LOAD	Loads a CSR file from a disk to the RAM memory.
MAKE	Generates a CSR.
NO	Deletes an option or configures it with its default value.
PASSWORD	Configures the CSR *Password* attribute.
PRINT	Displays the CSR content on the screen.
SIGNATURE-ALGORITHM	Configures the *Signature-Algorithm* used in the CSR.
SUBJECT-NAME	Configures the CSR *Subject-name.*
EXIT	Exits the CSR configuration menu.

"ALT-NAME [Alternative-name]"

This command configures the *Alternative-name* attribute which forms part of the CSR generated in the future.

To delete the configured *Alternative-name*, use the "NO ALT-NAME" command.

Example:

```
CSR config>alt-name ?
  <1..128 chars>    Alternative-name text
```

```
CSR config>alt-name teldat.imasd.es
CSR config>
```

"CLEAN"

This command deletes the stored CSR from the RAM memory.

Example:

```
CSR config>clean
CSR cleaned OK
CSR config>
```

"DELETE [file name]"

This command deletes a CSR file stored on disk.

Example:

```
CSR config>delete ?
  <word>    File name
CSR config>delete prueba
CSR successfully deleted from disk
CSR config>
```

"LIST"

This command displays a list of CSR files stored on disk.

Example:

```
CSR config>list
 A:                    PRUEBA.CSR        494   06/25/09   10:38   Flash
 A:                   PRUEBA2.CSR        494   06/25/09   10:38   Flash
CSR config>
```

"LOAD [file name]"

This command loads a CSR file stored on disk to the RAM memory.

Example:

```
CSR config>load ?
  <word>    File name
CSR config>load prueba
 CSR loaded
CSR config>
```

"MAKE [RSA Key ID]"

This command is used to create a CSR associated to the private key that the entered identifier has. So the CSR can be created, the "Subject-name" attribute must be configured as you will see further on. Once the CSR has been created, the router will displays this on screen and ask if you want to store it on disk.

Example:

```
CSR config>make ?
  <1..65535>    RSA Key ID for the CSR request
CSR config>make 1
  Certificate Request
==================
-----BEGIN CERTIFICATE REQUEST-----
MIIBdTCCAR8CAQAwgYwxCzAJBgNVBAYTAmVzMQ8wDQYDVQQHEwZtYWRyaWQxDzAN
BgNVBAoTBnRlbGRhdDEOMAwGA1UECxMFaW1hc2QxEDAOBgNVBAsTB3BsYW50YTEx
GDAWBgNVBAMTD3RlbGRhdC5pbWFzZC5lczEfMB0GCSqGSIb3DQEJARMQcHJ1ZWJh
QHRlbGRhdC5lczBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDCCyRNzTmf9rc8OBn8
72T1Gl75lBdAUYK6G5IlQWxVvk2wn0+4wvsEHB2nvUiCrp/dxdhTnmdBZO/Q0nFS
uRzpAgMBAAGgLTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQQTMBGCD3RlbGRhdC5p
bWFzZC5lczANBgkqhkiG9w0BAQQFAANBAEoKwrsADiSfKt5hWPunuBEwxRhFRz7S
Ty4ykAUFaNuneaq7T6XCz77uszsVt4v5QQJ8N7M7wr0dQBIBNVxbwbY=
-----END CERTIFICATE REQUEST-----
Save in file(Yes/No)? n
CSR config>
```

"NO SUBJECT-NAME/ALT-NAME/PASSWORD/SIGNATURE-ALGORITHM"

With this command you can delete the configuration for the various CSR attributes such as the *Subject-name*, the *Alternative-name* and the *Password*, as well as reconfigure the *Signature-algorithm* to its default value (MD5).

"PASSWORD CIPHERED/PLAIN [Password]"

This command configures the *Password* attribute that will form part of the CSR generated in the future. This password can be entered encrypted through the CIPHERED subcommand, or in clear using the PLAIN subcommand.

To delete the configured password, use the "NO PASSWORD" command.

Example:

```
CSR config>password ?
  ciphered    Ciphered password
```

```
  plain       Plain password
CSR config>password pla
CSR config>password plain ?
  <4..95 chars>    Text
CSR config>password plain mypassword
CSR config>
```

"PRINT ASN.1/BASE64/READABLE"

This command displays the CSR stored in the RAM memory on the screen. It can be displayed in various formats: ASN.1, BASE64 or READABLE.

Syntax:

```
CSR config>print ?
  asn.1       ASN.1 format
  base64      BASE64 format
  readable    Readable format
CSR config>
```

Example 1:

```
CSR config>print asn.1
Certificate Request
===================
30 82 01 75 30 82 01 1F 02 01 00 30 81 8C 31 0B 30 09 06 03
55 04 06 13 02 65 73 31 0F 30 0D 06 03 55 04 07 13 06 6D 61
64 72 69 64 31 0F 30 0D 06 03 55 04 0A 13 06 74 65 6C 64 61
74 31 0E 30 0C 06 03 55 04 0B 13 05 69 6D 61 73 64 31 10 30
0E 06 03 55 04 0B 13 07 70 6C 61 6E 74 61 31 31 18 30 16 06
03 55 04 03 13 0F 74 65 6C 64 61 74 2E 69 6D 61 73 64 2E 65
73 31 1F 30 1D 06 09 2A 86 48 86 F7 0D 01 09 01 13 10 70 72
75 65 62 61 40 74 65 6C 64 61 74 2E 65 73 30 5C 30 0D 06 09
2A 86 48 86 F7 0D 01 01 01 05 00 03 4B 00 30 48 02 41 00 C2
0B 24 4D CD 39 9F F6 B7 3C 38 19 FC EF 64 F5 1A 5E F9 94 17
40 51 82 BA 1B 92 25 41 6C 55 BE 4D B0 9F 4F B8 C2 FB 04 1C
1D A7 BD 48 82 AE 9F DD C5 D8 53 9E 67 41 64 EF D0 D2 71 52
B9 1C E9 02 03 01 00 01 A0 2D 30 2B 06 09 2A 86 48 86 F7 0D
01 09 0E 31 1E 30 1C 30 1A 06 03 55 1D 11 04 13 30 11 82 0F
74 65 6C 64 61 74 2E 69 6D 61 73 64 2E 65 73 30 0D 06 09 2A
86 48 86 F7 0D 01 01 04 05 00 03 41 00 4A 0A C2 BB 00 0E 24
9F 2A DE 61 58 FB A7 B8 11 30 C5 18 45 47 3E D2 4F 2E 32 90
05 05 68 DB A7 79 AA BB 4F A5 C2 CF BE EE B3 3B 15 B7 8B F9
41 02 7C 37 B3 3B C2 BD 1D 40 12 01 35 5C 5B C1 B6
CSR config>
```

Example 2:

```
CSR config>print base64
Certificate Request
===================
-----BEGIN CERTIFICATE REQUEST-----
MIIBdTCCAR8CAQAwgYwxCzAJBgNVBAYTAmVzMQ8wDQYDVQQHEwZtYWRyaWQxDzAN
BgNVBAoTBnRlbGRhdDEOMAwGA1UECxMFaW1hc2QxEDAOBgNVBAsTB3BsYW50YTEx
GDAWBgNVBAMTD3RlbGRhdC5pbWFzZC5lczEfMB0GCSqGSIb3DQEJARMQcHJ1ZWJh
QHRlbGRhdC5lczBcMA0GCSqGSIb3DQEBAQUAA0sAMEgCQQDCCyRNzTmf9rc8OBn8
72T1Gl75lBdAUYK6G5IlQWxVvk2wn0+4wvsEHB2nvUiCrp/dxdhTnmdBZO/Q0nFS
uRzpAgMBAAGgLTArBgkqhkiG9w0BCQ4xHjAcMBoGA1UdEQQTMBGCD3RlbGRhdC5p
bWFzZC5lczANBgkqhkiG9w0BAQQFAANBAEoKwrsADiSfKt5hWPunuBEwxRhFRz7S
Ty4ykAUFaNuneaq7T6XCz77uszsVt4v5QQJ8N7M7wr0dQBIBNVxbwbY=
-----END CERTIFICATE REQUEST-----
CSR config>
```

Example 3:

```
CSR config>print readable
Certificate Request
===================
Version                  : V1
```

```
Subject:
  E  (Email              ): prueba@teldat.es
  CN (Common Name        ): teldat.imasd.es
  OU (Organizational Unit): planta1
  OU (Organizational Unit): imasd
  O  (Organization Name  ): teldat
  L  (Locality           ): madrid
  C  (Country Name       ): es
Publick Key               :
  Algorithm Identifier    : RSA
  Modulus Length          : 512 Bits.
  Modulus                 :
    C20B 244D CD39 9FF6 B73C 3819 FCEF 64F5 1A5E F994 1740 5182 BA1B 9225
    416C 55BE 4DB0 9F4F B8C2 FB04 1C1D A7BD 4882 AE9F DDC5 D853 9E67 4164
    EFD0 D271 52B9 1CE9
  Exponent                :
    01 00 01
CSR Attributes            :
  1.2.840.113549.1.9.14
    311E301C301A0603551D110413301182 : 1.0.0...U....0..
    0F74656C6461742E696D6173642E6573 : .teldat.imasd.es
CSR Signature
  CSR Signature Algorithm : MD5 With RSA
  CSR Signature Data Info : 512 Bits.
  Signature Data          :
    4A0A C2BB 000E 249F 2ADE 6158 FBA7 B811 30C5 1845 473E D24F 2E32 9005 0568
    DBA7 79AA BB4F A5C2 CFBE EEB3 3B15 B78B F941 027C 37B3 3BC2 BD1D 4012 0135
    5C5B C1B6
CSR config>
```

"SIGNATURE-ALGORITHM MD5/SHA1/MD2"

Use this command to select the encryption algorithm that you want to use to execute the CSR signature. This signature can be MD5, SHA1 or MD2, default being MD5.

To return to the default value, use the "NO SIGNATURE-ALGORITHM" command.

Example:

```
CSR config>signature-algorithm ?
  md5     Set Signature Algorithm of CSR to MD5 algorithm
  sha1    Set Signature Algorithm of CSR to SHA1 algorithm
  md2     Set Signature Algorithm of CSR to MD2 algorithm
CSR config>signature-algorithm sha1
CSR config>
```

"SUBJECT-NAME [Subject-name in X500 format]"

This command configures the *Subject-name* attribute for the CSR to be generated in the future. This attribute must be configured so the CSR can be created.

To configure the *Subject-name*, the various fields for this are entered in a string of characters separated by commas and in X500 format as shown below:

X500 Format: " **C** = Country**, L** = Locality**, ST** = State or Province**, O** = Organization,
OU = Organization Unit**, CN** = Common Name**, E** = Email ... "

It's possible to modify the configured *Subject-name*, i.e. add new fields to it. To do this you can re-execute the command, storing this in different lines. The maximum number of fields that this can have is 20, and you can enter various fields of the same type.

To delete the configured *Subject-name* use the "NO SUBJECT-NAME" command.

Example:

```
CSR config>subject-name "c = es, l = madrid, o = teldat, ou = imasd, ou = planta1"
CSR config>subject-name "cn = teldat.imasd.es, e = prueba@teldat.es"
CSR config>
```

NOTE1: if you wish to enter the ' = ' , ' \ ' o ' , ' characters in any of the fields, you must enter the escape character ' \ 'before them.

NOTE2: You are not allowed to use double commas ' " ' in any of the fields

"EXIT"

This command permits you to exit the CSR menu and return to the CERT menu.

Example:

```
CSR config>exit
CERTIFICATES config>
```

5.5. Obtaining certificates through SCEP

You can obtain a certificate for a Teldat device through the *Simple Certificate Enrollment Protocol*, SCEP. The idea is to get two files: the CA certificate, *caname.cer*, and the Router certificate, *router.cer* in cases where this isn't the root CA, you will have to install all the certificates from the path up to the root CA). I.e. this method of getting certificates is an alternative to getting certificates through CSR as explained in prior sections. We recommend that you read the section on *obtaining certificates through CSR* before continuing.

The SCEP permits you to get the certificates through a connection to a server (normally the CA). The device establishes an HTTP connection to the server and they exchange information over this.

The protocol is based on the following procedures:

1. Installing the CA certificate, *Install*
2. Installing the user certificate, *Enroll*

Install

The first thing to do is to install the CA certificate in the device. There are various options:

- Through configuration: I.e. entering the commands that define a certificate file. Example:

  Example:

```
file new TELDATCA.CER
file add 0x308202AF30820218A003020102020101300D06092A864886F70D010104050030
file add 0x6B310B3009060355040613025553310B3009060355040813024E433110300E06
file add …
file end 0xC1809E37BB050F7D27DB2C2ACC8AD4
```

- Loading a file in base64 format using the CERTIFICATE [CertFile] BASE64 command.
- Loading the file through ftp and executing the CERTIFICATE [CertFile] LOAD command.
- Executing the install-ca [group] command in the SCEP menu.

The Install process must have the domain name configured in the group SCEP menu through the DOMAIN command or in the IP menu through the DNS-DOMAIN-NAME command.

Enroll

Once the CA certificate has been installed, you need to install the user certificate. Before establishing contact with the server through SCEP, the device needs to have an RSA key in order to request the certificate. This key must be previously generated or it can be automatically generated if it doesn't exist, as explained in the section on the commands. Basically the device encapsulates a CSR request and sends it to the server which validates it and generates the corresponding certificate and returns it also encapsulated.

The *Enroll* process can be manually executed to install in the user certificate with human presence or you can configure an automatic installation. This means the device will try and get the certificate, provided it hasn't got one already or its validity period has expired.

View status

The obtained certificates automatically install in the device and you can check their presence from the CERT menu. These certificates have a name assigned in the SCEP configuration and are associated to a template as mentioned in the section on *obtain certificates through CSR*. The *list template all* command displays the general status of the configuration. The LIST command on the SCEP mentoring menu allows you to list the status of the configured groups.

CONFIGURATION

The device allows you to configure various SCEP servers with their corresponding parameters within groups which can be added to the configuration. To enter the SCEP menu, you need to execute the *scep* command in the IPSec CERT menu.

Command	Function
CA-CHAIN-INSTALL	Installs the chain of certificates up to the root CA.
CAPABILITIES	Displays the commands supported by the server.
ENROLL	Executes the *Enroll* protocol for a SCEP group.
GROUP	Creates or enters the configuration of an SCEP group.
INSTALL-CA	Executes the *Install* protocol for a SCEP group.
NEXT-CA-INSTALL	Installs the renewed CA certificates.
NO	Deletes or configures an option with its default value.
EXIT	Exits the SCEP configuration menu.

"CA-CHAIN-INSTALL [group]"

Executes the *GetCACertChain* query for an SCEP group.

This command should only be executed in cases where the INSTALL command doesn't install the chain of certificates up to the root CA. Some older servers offer the root CA certificate through the INSTALL command, instead of the complete chain. Only in this case do we recommend using this command.

Example:

```
hub1 SCEP config$ca-chain-install 1
Installing CA certificate...

Opening...
```

```
172.24.75.193...

Sending Query. Waiting Answer...

Certificate name: TELDAT.CER

Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y

Saving CA certificate teldat

Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):

**************************************
Do not forget to save the configuration!
**************************************
```

"CAPABILITIES [group]"

Executes the *GetCACaps* query for an SCEP group which displays the functionalities supported by the server. This command began to be seen in 2008, consequently servers with older versions do not offer a correct response and the result shown on the console is something like "Parsing error" or "Command error".

Example:

```
hub1 SCEP config$capabilities 1
Opening...
172.24.75.193...

Sending Query. Waiting Answer...
"GetNextCACert"
"POSTPKIOperation"
```

"ENROLL [group]"

Executes the *Enroll* process for an SCEP group. This forces the operation to execute independently of the AUTOENROLLMENT parameter configured in the group.

On receiving the certificate, the certificate *fingerprint* is displayed so the user can accept it. The fingerprint is the certificate's MD5 hash.

Don't forget to save the configuration so the certificate and the generated passwords are retained after a reboot.

Example:

```
SCEP config$enroll 1

Building CSR...

Ciphering envoloped data...

Building signature...

Opening...
172.24.75.193...

Sending Query. Waiting Answer...
```

```
Certificate name: ROUTER.CER

Fingerprint: 42 A4 8F 61 E4 1D 39 91 7B 34 0B EA C6 09 B3 47
Do you accept the certificate received(Yes/No)? y

Saving CA certificate router

Version                   : V3
Serial Number             : 1C
Algorithm Identifier      : MD5 With RSA
Issuer:
  CN (Common Name        ):

**************************************
Do not forget to save the configuration!
**************************************
```

"GROUP [group]"

Creates or enters the configuration for a SCEP group.

Example:

```
SCEP config$group 1

-- Scep group user configuration --
SCEP group 1 config$
```

"INSTALL-CA [group]"

Executes the *Install* process for an SCEP group.

On receiving the certificate, the certificate *fingerprint* is displayed so the user can accept it. The fingerprint is the certificate's MD5 hash.

Don't forget to save the configuration so that the certificate and the generated passwords are retained after a reboot.

Example:

```
hub1 SCEP config$install-ca 1
Installing CA certificate...

Opening...
172.24.75.193...

Sending Query. Waiting Answer...

Certificate name: TELDAT.CER

Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y

Saving CA certificate teldat

Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):

**************************************
Do not forget to save the configuration!
**************************************
```

"NEXT-CA-INSTALL [group]"

Executes the *GetNextCACert* query for an SCEP group which allows you to obtain the new CA certificates when the old ones are at the point of expiring. This command began to be seen in 2008; consequently servers with older versions do not offer a correct response and the result shown on the console is something like "Parsing error" or "Command error".

Example:

```
hub1 SCEP config$next-ca-install 1
Installing CA certificate...

Opening...
172.24.75.193...

Sending Query. Waiting Answer...

Certificate name: TELDAT.CER

Fingerprint: 79 34 C8 FA 4C 21 AD 82 45 5F 4C 51 C5 D7 9A 64
Do you accept the certificate received(Yes/No)? y

Saving CA certificate teldat

Version                   : V3
Serial Number             : 009C 78A9 B776 C6E8 9E46 CF33 8889 E779
Algorithm Identifier      : SHA1 With RSA
Issuer:
  CN (Common Name        ):

**************************************
Do not forget to save the configuration!
**************************************
```

SCEP GROUP CONFIGURATION

Command	Function
ALTERNATIVE-NAME	Alternative name for the CSR petition.
AUTOENROLLMENT	Configures the *auto-Enroll* mode.
CA-CERT-NAME	Name given to the associated CA certificate.
CGI-PATH	URL path.
CHALLENGE-PASSWORD	CSR password.
DEBUG	Debugging mode.
DOMAIN-NAME	Name of the CA domain.
GENERATEKEY	Generates the RSA key if the configured key can't be found.
IP-ADDRESS	Adds the configured IP to the CSR *Subject*.
PORT	Port where the server listens.
REGENERATEKEY	Regenerates the RSA key in each enrollment.
RSA-KEY-ID	RSA key identifier.
RSA-KEY-LENGTH	RSA key length.
SERIAL-NUMBER	Adds the device serial number to the CSR *Subject*.
SIGNER-NAME	Message signer name.
SUBJECT-NAME	CSR *Subject-name*.
URL	Server URL.
USER-CERT-NAME	Name given to the associated user certificate.
NO	Deletes or configures an option with its default value.
EXIT	Exits the SCEP configuration menu.

"ALTERNATIVE-NAME [Alternative-name]"

Use this command to configure the *Alternative-name* that forms part of the CSR which is generated in the Enroll process.

To delete the *Alternative-name*, use the "NO ALTERNATIVE-NAME" command.

Example:

```
SCEP group 1 config$alternative-name ?
  <word>    Text
SCEP group 1 config$alternative-name teldat.imasd.es
```

"AUTOENROLLMENT [time]"

Use this command to configure the automatic operating mode. In this mode the device executes the *Enroll* process provided that one of the below circumstances is fulfilled:

- That the certificate with the identifier name configured with the USER-CERT-NAME exists.
- The certificate identified through the USER-CERT-NAME command has less time than twice the time period configured to expire.

The device periodically checks to see if the circumstances above occur. The checking period is configured using the *period* parameter.

So the *Enroll* process can execute, you need to have previously carried out the *Install* process. Please remember that the *Install* process is not automatic.

To disable the automatic *Enroll* process, execute NO AUTOENROLLMENT.

This process is disabled by default.

Example:

```
SCEP group 1 config$autoenrollment ?
  <1m..1d>    Time value
SCEP group 1 config$autoenrollment 30m
```

"CA-CERT-NAME [name]"

This command identifies the name the CA certificate has once it's been obtained, or the name that it currently has if it is already loaded.

The default value for this command is "teldat".

"CGI-PATH [url-path]"

This command specifies the path that the server URL has. This is normally a CGI. Default is "/cgi-bin/pkiclient.exe".

```
SCEP group 1 config$cgi-path ?
  <word>    Text
SCEP group 1 config$cgi-path /~pkii/pkiclient.php
```

"CHALLENGE-PASSWORD [password]"

This command configured the *Password* attribute that forms part of the CSR generated in the *Enroll* process.

"DEBUG"

This command is only active for operations forced from the console. When this command is enabled the requests sent to the server are printed so they can be subsequently analyzed.

"DOMAIN-NAME"

This command establishes the CA domain name. In cases where this parameter isn't configured, the CA domain name is taken from the configuration in IP using the DNS-DOMAIN-NAME command.

If this command isn't configured and the DNS-DOMAIN-NAME doesn't exist, the *Install* process ends in error.

"GENERATEKEY"

This command provokes the generation of an RSA key in the *Enroll* process if the one configured with the RSA-KEY-ID command can't be found.

If the key configured with the RSA-KEY-ID command exists, then this command doesn't have any effect.

If the key configured with the RSA-KEY-ID doesn't exist and this command isn't enabled, the *Enroll* process terminates with error.

Please see the RSA-KEY-ID and the REGENERATEKEY commands.

This command is disabled by default.

"IP-ADDRESS [ip] | [interface]"

If this command is configured, the IP is included in the CSR *Subject* generated in the *Enroll* procedure. You can configure an IP or an interface; in the latter the interface's primary IP is used. The IP is encoded in the *unstructuredAddress* attribute. By default the IP is not included in the CSR.

Example:

```
SCEP group 1 config$ip-address ?
  <a.b.c.d>      Ipv4 format
  <interface>    Interface name
SCEP group 1 config$ip-address ethernet0/0
```

"PORT"

This command specifies the port where the server listens. Default is 80, i.e. HTTP.

"REGENERATEKEY"

This command forces the *Enroll* process to generate an RSA key every time it's executed, provided that the key configured with the RSA-KEY-ID exists. Important; please note that if the RSA key specified with the RSA-KEY-ID command doesn't exist and the GENERATEKEY command isn't enabled, the *Enroll* process terminates with error independently of the value of this command.

Please see the RSA-KEY-ID and the REGENERATEKEY commands.

This command is disabled by default.

"RSA-KEY-ID [RSA key id]"

This command specifies the RSA key used in this SCEP group. The RSA keys are generated in the IPSEC menu using the KEY RSA GENERATE command. If the configured key can't be found or this command is disabled, the device generates an RSA key in the *Enroll* process provided that the GENERATEKEY command is enabled.

Please see the RSA-KEY-ID and the REGENERATEKEY commands.

This command is disabled by default.

In example mode, the *Enroll* process has the following effects in these configurations.

- (no rsa-key-id) and (generate):
  - A key is generated in the first *Enroll* process which is retained in successive *Enrollments*.
- (no rsa-key-id) and (generate) and (regenerate):
  - A key is generated in the first *Enroll* process which is renewed in successive *Enrollments*.
- (rsa-key-3):
  - If the key with id 3 exists, it is used in the first *Enroll* process and in successive ones.
  - If it doesn't exist then the *Enroll* process terminates with error.
- (rsa-key-3) && (regenerate):
  - If the key with id 3 exists, a new one is regenerated in successive *Enroll* processes.
  - If it doesn't exist then the *Enroll* process terminates with error.

"RSA-KEY-LENGTH [length in bits]"

Specifies the automatically generated RSA key length in bits. This command has no effect if the indicated key already exists through the RSA-KEY-ID command.

If the key indicated with the RSA-KEY-ID doesn't exist, the keys automatically generated in the *Enroll* process take the length specifically specified here.

If the key indicated with the RSA-KEY-ID doesn't exist and this command isn't configured, the keys automatically generated in the *Enroll* process take the length from the CA certificate module.

"SERIAL-NUMBER"

If this command is configured, the device serial number is included in the CSR *Subject* generated in the *Enroll* procedure. The serial number is encoded in the *serialNumber* attribute. By default the serial number is not included in the CSR.

"SIGNER-NAME [name]"

This command specifies the identifier that is attached as the signer-name for the message sent in the *Enroll* process.

If this command isn't configured, the signer-name is obtained from the domain name configured in the IP menu through the DNS-DOMAIN-NAME command preceded by the device name, configured through the SET HOSTNAME command found in the configuration root menu, and a period.

Example:

DNS-DOMAIN-NAME = teldat.com

HOSTNAME = imasd

The signer-name is consequently: imasd.teldat.com

If this command isn't configured and the HOSTNAME or the DNS-DOMAIN-NAME doesn't exist then the *Enroll* process terminates with error.

“SUBJECT-NAME [name]”

This command is used to configure the CSR *Subject-name* attribute which is generated in the *Enroll* process.

To configure the *Subject-name*, you need to introduce, in a string of characters, the different fields for this, separated by commas and en X500 format, as shown below:

X500 format: " **C** = Country**, L** = Locality**, ST** = State or Province**, O** = Organization, **OU** = Organization Unit**, CN** = Common Name**, E** = Email ... "

Example:

```
SCEP group 1 config>subject-name "c = es, l = madrid, o = teldat, ou = imasd, ou =
planta1"
SCEP group config>subject-name "cn = teldat.imasd.es, e = prueba@teldat.es"
```

NOTE1: if you want to enter the characters ‘ = ’ , ‘ \ ’ o ‘ , ’ in some of the fields, you need to enter the escape character ‘ \ ’ in front of them.

NOTE2: You are not allowed to use inverted commas ‘ “ ’ in any of these fields.

If this command isn’t configured, the CSR subject-name is obtained from the domain name configured in the IP menu through the DNS-DOMAIN-NAME command preceded by the device name, configured through the SET HOSTNAME command found in the configuration root menu, and a period.

Example:

DNS-DOMAIN-NAME = teldat.com

HOSTNAME = imasd

The subject-name consequently is: imasd.teldat.com

If this command isn’t configured and the HOSTNAME or the DNS-DOMAIN-NAME doesn’t exist then the *Enroll* process terminates with error.

“URL [url]”

This command specifies the URL where the server listens.

Examples:

```
SCEP group 1 config> url ca.teldat.es
SCEP group 1 config> url 172.24.78.78
```

If you specify a URL that isn’t an IP address, the device must have a valid DNS server configured, which is capable of resolving the domain name specified in the URL.

“USER-CERT-NAME [name]”

This command identifies the name the user certificate has once it has been obtained or the name the CA certificate already has if it’s already loaded.

The default value for this command is “router”.

5.6. Certificate Revocation List CRL

Sometimes it's necessary to terminate or invalidate a certificate for various reasons, such as changing the name, device, or because there is a security risk. There are lists of certificates that have been made redundant which can be checked to see if a certificate is still in force. This list is known as CRL, *Certificate Revocation List.*

A device can obtain a CRL from a LDAP server (Lightweight Directory Access Protocol), which is normally located in the CA itself. The IPSec LDAP menu permits you to configure four servers where you can download the CRL. Subsequently this server is assigned a *template*. Once the CRL has been obtained, it is saved in the non-volatile memory so it is not lost when the device is restarted.

The device supports DeltaCRLs. These are publications that are carried out between CRLs publication to only communicate the intermediate changes

You can also download the CRL, installing it in the device non-volatile memory through FTP.

In order to configure the CRL you need to define the search parameters in order to access the server and in addition activate the use of the CRL in the template. To configure the search parameters, use the IPSec *LDAP commands* and to activate the use, use the CRL command found in the *template* menu.

a) IPSec LDAP Command

Command	Function
SERVER #	Configures the LDAP# server parameters.
TIMER	CRL query time in the LDAP server

"LDAP SERVER [ID] DESTINATION ADDRESS [DirIP]"

Use this command to configure the LDAP server IP address or the domain name.

You can choose to let the device obtain the server address by searching in the CA certificate *CRL Distribution Points* extension. To achieve this behavior, you need to configure the *use-ca-subj-as-dn* option in the *template* as indicated in the section on the template CRL Command.

Example:

```
IPSec config>ldap server 1 destination address ldap.teldat.es
```

"LDAP SERVER [ID] DESTINATION PORT [Port]"

Use this command to configure the port where the LDAP server listens. Default port is 389.

Example:

```
IPSec config>ldap server 1 destination port 370
```

"LDAP SERVER [ID] SOURCE-ADDRESS [DirIP]"

Use this command to configure the source IP address which is used in petitions to the LDAP server.

Example:

```
IPSec config>ldap server 1 source-address 2.2.2.2
```

"LDAP SERVER [ID] DN [Distinguished Name]"

Use this command to configure the DN, *distinguished name*, which is used in petitions to the LDAP server.

You can choose to let the device obtain the DN by searching in the CA certificate *CRL Distribution Points* extension. To achieve this behavior, you need to configure the *use-ca-subj-as-dn* option in the *template* as indicated in the section on the template CRL Command.

Example:

```
IPSec config>ldap server 1 dn "ou=For Test Purposes Only,o=Teldat"
```

"LDAP SERVER [ID] AUTHENTICATION [String]"

Use this command to configure the simple authentication string which is used in petitions to the LDAP server.

Example:

```
IPSec config>ldap server 1 authentication "teldat"
```

"LDAP SERVER [ID] NAME-AUTH [String]"

Use this command to configure the simple authentication name which is used in petitions to the LDAP server.

Example:

```
IPSec config>ldap server 1 name-auth "user@teldat.com"
```

"LIST LDAP SERVER"

Displays a list of the configured servers. With that configured in the above examples, you get:

Example:

```
IPSec config>list ldap server
LDAP Server 1
    destination: ldap.teldat.es
    destination port: 370
    source address: 2.2.2.2
    dn: ou=For Test Purposes Only,o=Teldat
    name used for authentication: user@teldat.com
    authentication: teldat
```

"LDAP TIMER [SECONDS]"

Establishes the time between CRLs searches. The default value is one day.

There is an optional field in the CRL that establishes when you should execute the *Next Update*, and another one, *Next Publish*, that establishes when the next publication of another CRL or a DeltaCRL is going to be carried out.. Updating is carried out when the date indicated by *Next Update*, *Next Publish* arrives or when the period programmed between searches times out, whichever occurs first.

Example:

```
IPSec config>ldap timer 2d
IPSec config>lis ldap timer

Period to check LDAP servers: 48h0m0s
```

- *Attributes*

The default search attribute used in LDAP to obtain the CRL in IPSec is as follows:

certificaterevocationlist;binary

You can select another attribute by changing the LDAP global configuration, as explained in manual *Dm790-I LADP Protocol*. When you have configured an attribute in the LDAP global configuration, this is used in the CRL search instead of the using the default one.

The rest of the parameters used for the search are obtained from those configured in the LDAP global configuration.

b) Template CRL Command

Command	Function
OPTIONAL	Continuous even if the CRL is not available.
ALWAYS	The CRL must always be available.
LDAP-SERVER	Assigns an IPSec LDAP server.
USE-CA-SUBJ-AS-DN	Uses the CA *subject* as CRL DN.

"TEMPLATE [ID] IKE CRL OPTIONAL"

When you try and use a certificate, if the CRL is not available, the process will continue assuming that the certificate has not been revoked. The OPTIONAL or ALWAYS command must be active so the device can use the CRL.

Example:

```
IPSec config>template 22 ike crl optional
```

"TEMPLATE [ID] IKE CRL ALWAYS"

When you try and use a certificate, if the CRL is not available, the process will interrupted assuming that the certificate might have been revoked. The OPTIONAL or ALWAYS command must be active so the device can use the CRL.

Example:

```
IPSec config>template 22 ike crl always
```

"TEMPLATE [ID] IKE CRL LDAP-SERVER [ID]"

Use this command to assign an LDAP server configured in IPSec to the template. When you search for the CRL through LDAP, the connection parameters are taken from the server assigned through this command.

Example:

```
IPSec config>template 22 ike crl ldap-server 2
```

"TEMPLATE [ID] IKE CRL USE-CA-SUBJ-AS-DN"

When a CRL search is initiated through LDAP, it tries to obtain the server DN and address in the CA certificate *CRL Distribution Points* extension.

- If the DN cannot be found, this is taken from the CA certificate *subject* field.
- If the server address cannot be found, then the address configured in the assigned LDAP server is used.

If you want to use a different DN from the CA *subject*, or you don't want to use the address of the server located between the CA certificate extensions, you need to configure "no use-ca-subj-as-dn" and configure the DA and IP address in the assigned LDAP server as indicated in the section on IPSec LDAP Command.

Example:
DN the same as the CA subject.

```
IPSec config>template 22 ike crl use-ca-subj-as-dn
```

Example:
DN different from the CA subject.

```
IPSec config>template 22 ike crl ldap-server 2
IPSec config>template 22 ike crl no use-ca-subj-as-dn
IPSec config>ldap server 2 default
IPSec config>ldap server 2 destination address 81.11.11.121
IPSec config>ldap server 2 dn "CN=Test,C=ES"
```

The *list template all* command displays the configuration status:

```
IPSec config>list template all
TEMPLATES
22 isakmp 3DES MD5  DES=0.0.0.0
   LifeTime:1h0m0s
   IKE MAIN
   RSA SIGNATURE
      CA      :VRSGNCA.CER
               -ou=For Test Purposes Only,o=Teldat
               -Without Private Key
               -   Signature ok.
      CRL     : VRSGNCA.CRL
               -Search of CRL by Subject of CA failed, VRSGNCA.CER
               -ou=For Test Purposes Only,o=Teldat
               -Last update 0h12m31s
               -Next update in 0h0m40s
               -Number of items 5
               -ALWAYS enabled => CA must always be available
               -LDAP server number 3
      USER    :
   fqdn ID TYPE
   OAKLEY GROUP 1
```

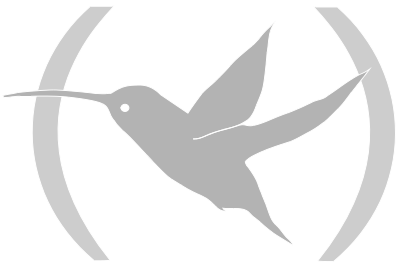
Exit the CRL menu to carry out operations with the CRL lists, such as list, delete, load and download. Executing commands from this menu does not involve changes in the configuration

The “time-to-expire” command permits you to temporarily advance the CRL search at a different time to that programmed.

```
IPSec config>cert

-- Cert user configuration --
CERTIFICATES config>crl

-- CRL user configuration --
CRL config>?
  delete            Delete a CRL
  list              List CRLs
  load              Load a CRL
  print             Print a CRL
  time-to-expire    Configure the time to expire of a CRL
  unload            Unload a CRL
  exit
CRL config>print all

Name: VRSGNCA.CRLVersion                    : V2
Algorithm Identifier      : SHA1 With RSA
DN                        : ou=For Test Purposes Only,o=Teldat
This Update               : Wed Oct 29 08:00:07 2008

Next Update               : Thu Oct 30 08:00:07 2008
```

```
Last update               : 0h18m4s ago
Next update in            : 0h0m45s
Ldap status               : LDAPSTATUS_NOSUCHOBJECT
Number of items           : 5

Signature                 :
  Signature Algorithm     : SHA1 With RSA
  Signature Data Info     : 2048 Bits.
  Signature Data          :
    5C33 20CC FACA BD65 76AB 3FDB 4786 E620 FC0E 0DA4 B934 E745 2ACC 2453 9177
    9DCB 07D1 1FEB C6A1 0812 896E 6042 1DC2 A94D 85B6 60FA 4656 C07D 22BD 5C37
    6907 4765 E6F7 1CCC 6F44 B651 68F3 AC39 6886 9A79 13FC FAD4 8F79 04BF 69CA
    F68F 08BA A85D AB22 5BD2 1BA3 8B96 8961 380D 2B0C E157 274B 72C1 FB92 71D6
    0E3D 757A FAB5 A83B E903 9B13 A225 0183 2381 629E 6DE1 C099 2841 AC9E 6915
    6A05 25B1 0133 804D 07D1 A8D1 E94E 74C5 6745 0D65 0DA4 2776 215A 84B7 E0F8
    2350 EBD1 00DE 1B62 DCE7 0288 8A55 9199 03AB DFBF 185F 40F0 D12D 9C15 A6C2
    8CF1 9BD9 4329 81E0 3664 26A5 C831 F13F 9107 929A D2C1 8ECA 6F11 6D5F 4D4A
    15DB FBED 3C1A 7509 6464 DC78 C203 3935
```

Chapter 3
Monitoring



1. Introduction

IPSec monitoring in the **Teldat Routers** is carried out once the SPD elements have been configured.

The difference with respect to the configuration is that now you are not going to vary any parameter. The parameters need to be listed and if they are varied, this will be temporary. Changes carried out in the monitoring will only be valid until the router is restarted.

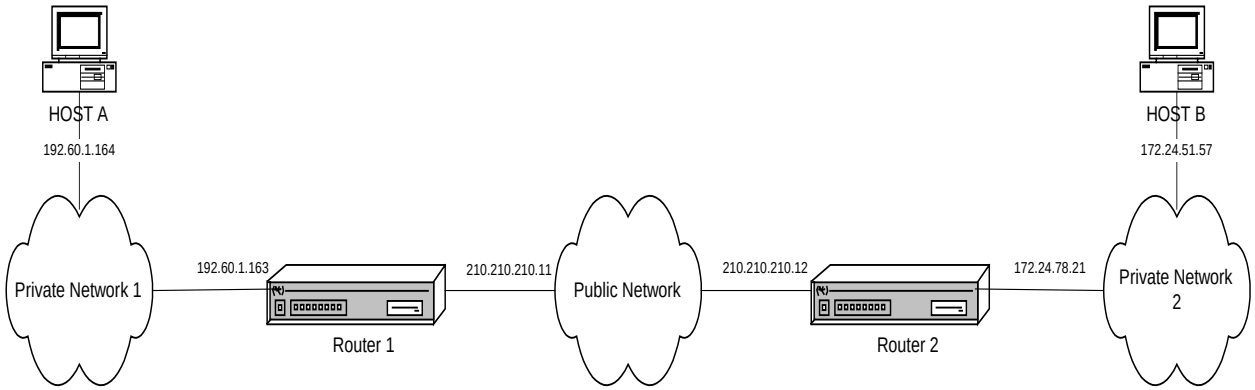
As seen in the introduction of this manual, the SAs (*Security Association*) are security connections that are created once the SPD has been consulted and contain the security information (authentication and encryption keys) needed in order to process the packet. Therefore when you create an SA, what you have is a connection established in order to securely transmit data between the two ends of the Tunnel.

There are two types of SAs, those of the first phase or ISAKMP and those of the second phase. The latter can be Dynamic SAs or Manual SAs. You must take into account that in the SAs, there is a clear difference between the Dynamic SAs and the ISAKMP SAs, with respect to the Manual SAs. The Manual SAs are permanent connections, meaning that when the Manual Templates are configured a connection is established between Tunnel ends. Contrariwise, the Dynamic SAs and the ISAKMP SAs, as they are dynamic only appear when using the connection between the Tunnels ends, i.e. when the Tunnel is established.

The monitoring displays an operation list of the previously configured connections, the ISAKMP SAs or from the first phase and Dynamic and Manual SAs or the second phase. Additionally this permits you, among other options, to eliminate the said connections.

Firstly the steps to follow are described in order to access to the said monitoring and secondly the available commands are explained in detail. Finally a problems and solutions reference is provided which can normally be found in the IPSec negotiations.

All of the examples seen for each monitoring command are based on the following scenario.



2. IPSec Monitoring

2.1. Initial Monitoring

This section describes the steps needed to access the IPSec monitoring in the **Teldat Router**. In order to enter the monitoring environment you must introduce the following commands:

```
*p 3
Console operator
+protocol ip
IP+ipsec
IPSec protocol monitor
IPSec+
```

Within the IPSec protocol monitoring environment the following commands are available:

Command	Operation
? (help)	Lists the commands or their available options.
address-to-ban	Introduces the Ipv4 addresses to be banned when using the protocol.
bitrate	Displays the real time rate for encapsulation and decapsulation.
cert	Enters the certificates monitoring menu.
clear	Clears the cache memory and the SAs (*Security Associations*).
filter-by-host	Only the events related with a specific *hostname* are monitored.
filter-dpd	The DPD (*Dead Peer Detection*) events are included in the register.
hardware	Functions related to the encryption cards (hardware encryption).
hostname-to-ban	Introduces the *hostnames* to be banned when using the protocol.
list	Lists the protocol elements.
monitor-level	Sets the monitoring level.
no	Negates a command or sets the default value.
shutdown	Closes all the open connections in order and disables the protocol.
stop-on-message	Stops the negotiation process message register when one specific one occurs.
exit	Exits the IPSec monitoring menu.

2.2. Monitoring Commands

a) address-to-ban

Through this command a range of IP address are introduced which cannot include the source or destination addresses for an SA (*Security Association*), preventing them from using the IPSec protocol.

"address-to-ban [IP add][mask]"
Prohibits the use of the IPSec protocol to IP addresses included in this range [IP add][mask].

Example:

```
IPSec+address-to-ban 210.210.210.0 255.255.255.0
IPSec+
```

In order to access the banned address information, use the **list banned** command as well as the **no address-to-ban** command to unblock them.

b) bitrate

The command permits real time monitoring of the encapsulation and decapsulation rate for the protocol packets. When this rate noticeably varies, a new entry is produced in the table. Strike any key to stop monitoring.

Example:

```
IPSec+bitrate
 Enc rate (bps/pps)  Dec rate (bps/pps)
 ------------------  ------------------
        480/    1          480/    1 (15:29:24)
       1136/    3          808/    2 (15:29:35)
       1456/    4         1808/    5 (15:29:36)
        480/    1          480/    1
```

c) cert

This command permits you to enter the certificates monitoring menu.

Example:

```
IPSec+cert

-- Cert user monitoring --
CERTIFICATES monit+
```

d) clear

Once you have selected **clear**, the following subcommands are available:

Command	Function
counters	Deletes the encryption queue and the used SAs counters.
sa	Cuts the SAs established connections.
statistics	Deletes the protocol statistics.

"clear counters"

Deletes the encryption queue and the used SAs counters. The contents of these counters can be viewed by using the **list advanced** command.

Example:

```
IPSec+clear counters
  All counters have been reset.
IPSec+
```

"clear statistics"

Deletes the protocol statistics. These statistics can be displayed through the **list statistics** command.

Example:

```
IPSec+clear statistics
  All IPSec statistics have been reset.
IPSec+
```

- *clear sa*

With this command you can cut the established connection between the Tunnel ends. The said interruption will depend on which type of SA you have.

If the SA is a Manual SA there is no reason to eliminate it as seen earlier, the connection is permanent therefore cannot be cut. What can be done is to eliminate the Dynamic SAs and the ISAKMP SAs.

Command	Operation
all	Eliminates all the Dynamic SAs and the ISAKMP SAs.
hostname-filter	Eliminates the SAs where a specific device participates.
in	Eliminates the input Dynamic SAs.
negotiation	Eliminates the ISAKMP SAs or first phase SAs.
out	Eliminates the output Dynamic SAs.

"clear sa all"

Eliminates all the ISAKMP SAs and all the dynamic SAs.

Example:

```
IPSec+clear sa all
 Clearing IPSec Connections...  Done
IPSec+
```

"clear sa hostname-filter [hostname]"

Eliminates the Dynamic SAs and the ISAKMP SAs where a specific device participates, determined by [hostname]. This permits you to use an asterisk (*) in order to include all the *hostnames* which begin with the same characters. In the following example, all the SAs in which devices whose *hostname* begins with HOST_ are eliminated.

Example:

```
IPSec+clear sa hostname-filter HOST_*
 HOST_*-->70.70.70.2
 Connection 2 cleared
 Connection 3 cleared
 Connection 1 cleared
IPSec+
```

By selecting any of the commands between **clear sa in**, **clear sa negotiation** and **clear sa out**, the following options open:

Command	Function
address-filter	Eliminates the SAs with source or destination Ipv4 included in a specific range.
all	Eliminates all the selected type.
connection	Eliminates the SA specified by an introduced number.
hostname-filter	Eliminates the SAs where a specific device participates.

"clear sa in/negotiation/out address-filter [ip add][mask]"

Clears the corresponding SAs (Input Dynamics/ISAKMPs/Output Dynamics) with a source or destination address that is included within the range defined by [ip add][mask].

Example:

```
IPSec+clear sa negotiation address-filter 210.210.210.12 255.255.255.255
 Connection 1 cleared
IPSec+
```

"clear sa in/negotiation/out all"

Clears all the selected type SAs (Input Dynamics/ISAKMPs/Output Dynamics).

Example:

```
IPSec+clear sa negotiation all
 All IPSec connections cleared
IPSec+
```

"clear sa in/negotiation/out connection [id]"

The [id] field is the SA identification number. This only clears the Dynamic SA or ISAKMP SA as applicable, defined by the [id] number.

Example:

```
IPSec+clear sa negotiation connection 1
 Connection 1 cleared
IPSec+
```

"clear sa in/negotiation/out hostname-filter [hostname]"

Eliminates the corresponding SAs (Input Dynamics/ISAKMPs/Output Dynamics) where a specific device participates, determined by [hostname].

Example:

```
IPSec+clear sa negotiation hostname-filter HOST_H
 HOST_H-->70.70.70.2
 Connection 1 cleared
IPSec+
```

e) filter-by-host

Using the **filter-by-host** entails delimiting the protocol monitoring to the events where this is present in a specific device, identified by its *hostname.*

"filter-by-host [hostname]"

Monitoring is circumscribed to that related to the device identified by the [hostname].

Example:

```
IPSec+filter-by-host HOST_H
 Filter activated with hostname HOST_H
IPSec+
```

You can see the [hostname] defined is the filter, should this be active, through the **list negotiation filter** command. To deactivate this, use the **no filter-by-host** command.

f) filter-dpd

This command activates filtering which shows the DPD (*Dead Peer Detection*) protocol events and registers

Example:

```
IPSec+filter-dpd
IPSec+
```

You can check that this filter is active through the **list negotiation filter** command; to deactivate it, use the **no filter-dpd** command.

g) hardware

Command	Function
enable	Enables the encryption card.

list	Lists the status of the hardware the IPSec uses.
test	Analyzes to see if there is encryption hardware present.

"hardware enable cf1531/mpc8272/mpc85xx/ts422"
Enables the selected hardware component, if it's available, to accelerate the encryption process.

Example:

```
IPSec+hardware enable ts422
IPSec+
```

The **hardware list** command shows if the encryption card is enabled; to disable this, use the **no hardware enable** command.

"hardware list"
Lists the status of the hardware used to accelerate the IPSec encryption process.

Example:

```
IPSec+hardware list

Hardware: TS-422
  SLOT 1. Revision: 0
    Status: OK. Access enabled.
        DES encode:719724  DES decode:719714
        AES encode:5  AES decode:3
        HASH:1439609
        RSA:933
        RNG:170
        MEMORY ALLOCATED:104 bytes (free 8088)

Hardware: CF1531
    Card not found
IPSec+
```

"hardware test"
Tests the encryption cards enabled in the device.

Example:

```
IPSec+hardware test
Warning! This will stop routing completely.
Are you sure to continue (Yes/No)? y

Test options:
1: Exhaustive test
2: Continuous writing
3: Only PKEU test
Choose a number[1]?
```

h) hostname-to-ban

Using the **hostname-to-ban** command, this specifies the *hostname* for a device which does not form a part of any SA (*Security Association*), blocking the use of the IPSec protocol

"hostname-to-ban [hostname]"
Prevents an IPSec tunnel from being established with a device whose *hostname* is [hostname].

Example:

```
IPSec+hostname-to-ban HOST_H
IPSec+
```

To access the information on the blocked devices, use the **list banned** command and the **no hostname-to-ban** command to unblock them.

i) list

This command displays information on the protocol monitoring. The following commands can be used:

Command	Function
access-lists	Displays information relative to the access lists.
address-filter	Offers all the protocol information related to the IP addresses included in a determined range.
advanced	Presents the contents of the used SAs and the encryption queue.
banned	Lists the addresses or devices that cannot use IPSec.
certificate_number	Certificate identified by the number assigned in the IKE negotiation process.
hostname-filter	Offers all the protocol information related to the device identified by its *hostname*.
negotiation	IKE negotiation process register (*Internet Key Exchange*).
notification	IKE negotiation notification messages.
sa	Security Associations.
statistics	IPSec protocol operating statistics.

- *list access-lists*

Once you enter this command, the following set of options unfolds:

Command	Function
address-filter	Displays the access lists information which includes the IP addresses contained in a determined range.
all	Presents all the information on the access lists (cache and entries).
Cache	Displays the cache for the access lists associated to IPSec.
entries	Displays the defined entries in the access lists associated to IPSec.

"list access-list address-filter [IP add][mask]"

The displayed information is limited to cases where an IP address contained in the range [IP add][mask] intervenes.

"list access-list all"

This command displays all the information available on the access lists assigned to IPSec, which partly consists of the cache and partly the list entries.

"list access-list cache"

This displays all the cache data for the access lists assigned to IPSec.

"list access-list entries"

This lists all the entries defined in the access lists assigned to IPSec.

- *list address-filter*

"list address-filter [IP add][mask]"

This command selects and displays all the IPSec monitoring information related to the IP addresses container in the [IP add][mask] range.

Example:

```
IPSec+list address-filter 210.210.210.12 255.255.255.255
SA OUT
SA 3 SPI=0x23951B2E
SA UP, ESP-DES ESP-MD5 SRC=210.210.210.12 DES=210.210.210.11
LifeTime:24h0m0s 4608000 kbytes  (23h46m31s 4608000 kbytes )
encode pkts:0 (err:0), decode pkts:0 (err:0)
```

- *list advanced*

Use this command to read the values for the encryption queue counter and the SAs usage counter. Additionally, this informs you if the protocol has been disabled through the **shutdown** command.

Example:

```
IPSec+list advanced
  Cipher Queue Size:            50
  Cipher Queue Water Mark:      3  reached 18h59m11s ago.
  Current Queue Level:          0
  Max SA simultaneous:          2  reached 21h53m55s ago.
  Current number of SA:         0
  Max negotiation duration:     17 msec
  Max tunnel supported:         40

  IPSec active
IPSec+
```

These values are initialized through the **clear counters** command.

- *list banned*

This lists the range of IP addresses and hostnames which cannot form part of an IPSec tunnel because they have been banned through the **address-to-ban** or **hostname-to-ban** commands.

Example:

```
IPSec+list banned

Banned addresses:
210.210.210.0

Banned hostnames:
HOST_H

IPSec+
```

- *list certificate_number*

"list certificate_number [id]"

This shows the information related to the certificate identified by the number [id] assigned during the IKE negotiation process.

- *list hostname-filter*

"list hostname-filter [hostname]"

This command selects and displays all the monitoring information related to the device identified by [hostname]. In the same way as the rest of the IPSec commands where you must enter a [hostname],

you can use an asterisk to select all the devices whose *hostname* begins with the characters preceding the said asterisk.

- *list negotiation*

Linked within this command, related to the IKE negotiation process register, the following options can be found:

Command	Function
address-filter	Provides information only on negotiations where a specific device, determined by its IP address, has intervened.
all	Displays the full negotiation process.
between	Only displays the IKE negotiation for the IPSec tunnels between specific devices.
filter	Specifies which filters are enabled.
hostname-filter	Provides information only on negotiations where a specific device, determined by its *hostname*, has intervened.
order	Displays the process ordered by conversations between pairs.

"list negotiation address-filter [IP add]"
The negotiation process displayed is limited to the entries where the device with address Ipv4 [IP add] intervenes.

"list negotiation all"
Displays all IKE negotiation process that has not yet been consulted.

Example:

```
IPSec+list negotiation all

210.210.210.12 210.210.210.11: (09:34:07)
              (* 36:--------- Local Starting Neg ---------)
              (* 06:--------- Creating ISAKMP NEG ---------)(# 1(0x1))
  (HDR 0)
      (HDR sa)
      (prop 1 isakmp #1)
      (trans 1 id=1)
              (encryp tdes)
              (hash sha)
              (grp desc 1)
              (auth presh)
              (life sec)
              (duration 43200)
  (09:34:22)
              (* 17:--------- Purgetime SA NEG ---------)
  (09:34:23)(* 07:--------- Deleting ISAKMP NEG ---------)(# 1(0x1))
  (09:34:24)
              (* 36:--------- Local Starting Neg ---------)
              (* 06:--------- Creating ISAKMP NEG ---------)(# 2(0x2))
  (HDR 0)
      (HDR sa)
      (prop 1 isakmp #1)
      (trans 1 id=1)
              (encryp tdes)
              (hash sha)
              (grp desc 1)
```

```
              (auth presh)
              (life sec)
              (duration 43200)
210.210.210.11 210.210.210.12: (HDR 0)
      (HDR sa)
      (prop 1 isakmp #1)
      (trans 1 id=1)
              (encryp tdes)
              (hash sha)
              (grp desc 1)
              (auth presh)
              (life sec)
              (duration 43200)
210.210.210.12 210.210.210.11:
              (* 01:--------- Matching template ---------)(# 1(0x1))
  (HDR 0)
      (HDR keyx)
      (HDR nonce)
      (vendor 13)(*  t:  8 , Teldat)
      (vendor attrcfg)
      (vendor xauth)
      (vendor dpd)
210.210.210.11 210.210.210.12: (09:34:25)
  (HDR 0)
      (HDR keyx)
      (HDR nonce)
      (vendor 13)(*  dc
                          6L Teldat)
      (vendor attrcfg)
      (vendor xauth)
      (vendor dpd)
210.210.210.12 210.210.210.11:
              (* 08:--------- Creating ISAKMP SA ---------)
  (HDR 0)
      (id addr4 prot=17 port=500)(# 210.210.210.12)
      (HDR hash)
210.210.210.11 210.210.210.12: (HDR 0)
      (id addr4 prot=17 port=500)(# 210.210.210.11)
      (HDR hash)
210.210.210.12 210.210.210.11:
              (* 09:--------- Creating ISAKMP SA id ---------)(# -1396216867
                                                                (0xacc76bdd))
  (HDR acc76bdd)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# -644186898(0xd99a7cee))
      (trans 1 id=tdes)
              (encap tunnel)
              (life sec)
              (duration 14400)
              (life kbytes)
              (duration 10000)
              (auth alg md5)
      (HDR nonce)
      (id addr4 prot=0 port=0)(# 172.24.51.155)
      (id addr4 prot=0 port=0)(# 200.100.0.21)
210.210.210.11 210.210.210.12: (HDR acc76bdd)
      (HDR hash)
      (HDR sa)
      (prop 1 esp #1)(# 1118431673(0x42a9e9b9))
      (trans 1 id=tdes)
              (encap tunnel)
              (life sec)
              (duration 14400)
              (life kbytes)
              (duration 10000)
              (auth alg md5)
      (HDR nonce)
      (id addr4 prot=0 port=0)(# 172.24.51.155)
```

```
      (id addr4 prot=0 port=0)(# 200.100.0.21)
210.210.210.12 210.210.210.11:
              (* 01:--------- Matching template ---------)(# 3(0x3))
  (HDR acc76bdd)
      (HDR hash)
              (* 11:--------- Creating SA IN ---------)(# -644186898(0xd99a7cee))
              (* 12:--------- Creating SA OUT ---------)(# 1118431673(0x42a9e9b9))
              (* 40:------!!! CONNECTED !!!------)
0.0.0.0: (09:35:57)

IPSec+
```

"list negotiation between [IP1 add][IP2 add]"

Selects and displays the negotiation process carried out between devices determined by addresses Ipv4 [P1 add] and [IP2 add].

"list negotiation filter"

Specifies which filters are enabled, among which the following can be found: *Negotiation filter*, *Event filter* and *DPD filter*. These have been enabled through the **filter-by-host**, **filter-dpd** commands, or in the IPSec configuration menu, using the **event address-filter** command; this latter establishes an *Event filter* with a range of IP addresses to be monitored. These filters affect the results of the **bitrate**, **list negotiation** and **list statistics** commands.

Example:

```
IPSec+list negotiation filter
Negotiation filter:  Hostname:HOST_* ,  Address:0.0.0.0
Event filter: Hostname:HOST_* ,  Address:255.255.255.255
DPD filter
IPSec+
```

"list negotiation hostname-filter [hostname]"

The displayed negotiation process is limited to the entries where a device identified by [hostname] intervenes.

- *list notification*

Displays the IKE negotiation notification messages. The proposed failed negotiations, incompatible or deleted SAs, etc.

Example:

```
IPSec+list notification

  (Time ************************************ 0h14m5s)
IPSec+
```

- *list sa*

You can display all the SAs with this command. With this list you will know if the connections are active or not.

The Manual SAs, since they are permanent connections, will always been seen on the list. However, the Dynamic SAs and the ISAKMP SAs as they are dynamic will only be seen if using the connection between Tunnel ends, i.e. you are transmitting data.

Command	Operation
in	Lists the input Dynamic and Manual SAs.
..negotiation	Lists the ISAKMP SAs or the first phase SAs.

out	Lists the output Dynamic and Manual SAs.

Each of these commands also has a last set of options associated:

Command	Function
address-filter	Only lists the SAs that intervene in an IP address included in a range.
all	Displays information on all the selected type SAs.
hostname	(*Only for negotiation*) Lists the ISAKMP SAs where a specific device has participated.

"list sa in/negotiation/out address-filter [IP add][mask]"

Lists the corresponding active SAs (input/ISAKMP/output) where the source or destination address is included in a range defined by [IP add][mask].

Example:

```
IPSec+list sa negotiation address-filter 210.210.210.11 255.255.255.255
SA NEGOTIATION
  SA 54 (i_cookie=0xd5a04a00ce28530c r_cookie=0x4734ac9b10a99cf9)
  Inic=210.210.210.11  Resp=210.210.210.12
  SRC=210.210.210.11 DES=210.210.210.12 STATE=5
   LifeTime:12h0m0s (11h57m57s)
   ClientSRC=192.60.1.164 ClientDES=172.24.51.57 Rule=0 Ifc=ppp200
   ISAKMP_SA available, STATE=ESTABLISH :Purgetime=15
      ISAKMP_NEGII id 0xb906469c, (0xa22a731e/0xc508758d)
        SRC=192.60.1.164/32 DES=172.24.51.57/32
        LifeTime:4h0m0s 10000 kbytes  (3h57m56s 9991 kbytes )
        encode pkts:120 (err:0), decode pkts:120 (err:0)
IPSec+
```

"list sa in/negotiation/out all"

Lists all the selected type active SAs (input/ISAKMPs/output). In cases of input or output SAs, this provides information on all the active Manual SAs together with the Dynamic ones.

Example:

```
IPSec+list sa in all
SA IN
   SA 245 SPI=0x1c839ab2
   SA UP, ESP-3DES ESP-SHA1     SRC=210.210.210.12 DES=210.210.210.11
   LifeTime:1h0m0s (0h56m25s)
   encode pkts:0 (err:0), decode pkts:0 (err:0)
   DPD: idle for 4(15) seconds
IPSec+
```

"list sa negotiation hostname [hostname]"

Displays the active ISASMP SAs where one of the IPSec tunnel ends is a device identified by [hostname].

- *list statistics*

This lists the IPSec protocol operation statistics.

Example:

```
IPSec+list statistics

----ESP/AH Statistics:----

Input Stats
-----------
  Frames ok    13
  Frames error 0
  ---> Out-of-Order frames      0
  ---> Unknown payload protocol 0
  ---> Internal errors          0
  Frames/sec 1 (max 1)
  kbits/sec 0 (max 0)
Output Stats
------------
  Frames ok    13
  Unknown authentication algorithm 0
  Frames/sec 1 (max 1)
  kbits/sec 0 (max 0)


----IPSEC Forwarding Statistics:----

Sa in not found                0
Sa out Template not found      0
Sa out not found(only manual)  0

----IKE Statistics:----
Negotiation phase I            1
Negotiation phase II           1
Check Hash Error phase I       0
Check Hash Error phase II      0
Drops Collision IKE messages   0
Drops Waiting IKE Processing   0

   Cypher queue empty:          0

   Number of open connections not notified during last connevent-period: 0

IPSec+
```

To restart the corresponding variables, use the **clear statistics** command.

- *monitor-level*

The **monitor-level** command has one option, **verbose**, used to specify that the monitoring information to be showed is shown in detail.

"monitor-level verbose"
The enables the monitoring level detailed information.

Example:

```
IPSec+monitor-level verbose
IPSec+
```

To disable the *verbose* monitoring mode, use the **no monitor-level verbose** command.

j) no

The aim of this command is to simply execute the opposite actions for other protocol commands. The menu that opens in **no** is a replica of the initial menu although it only contains those commands where the no function exists.

Command	Function
address-to-ban	Unblocks the banned Ipv4 addresses for IPSec.
filter-by-host	Stops limiting monitoring for a specific device.
filter-dpd	Does not include the DPD protocol events and registers.
hardware	Using the **enable** option, this disables a specific encryption card.
hostname-to-ban	Unblocks the devices banned for IPSec.
monitor-level	Using the **verbose** option, this reduces the amount of detailed monitoring information displayed.
shutdown	Enables the IPSec protocol.
stop-on-message	Does not stop the negotiation process register when a determined message is produced.

k) shutdown

Shuts down all the open IPSec connections in an orderly manner and disables the protocol.

Example:

```
IPSec+shutdown
 Clearing IPSec Connections...  Done
IPSec+
```

WARNING: If you are accessing through a remote console through an IPSec session and you execute this command, you will immediately lose access and won't be able to recover it until the device is rebooted or until you execute the* no shutdown *command through an access mode which is not protected by IPSec.

l) stop-on-message

Stops the negotiation process message register when a specific message occurs. The options for this command are the numbers that identify each one of the possible messages that stop the register.

Command	Function
01	--------- Matching template ---------
02	--------- Matching SA NEG ---------
03	--------- Decryption error ---------
04	--------- Retransmission ---------
05	--------- Unable to initiate. Unknown destination ---------
06	--------- Creating ISAKMP NEG ---------
07	--------- Deleting ISAKMP NEG ---------
08	--------- Creating ISAKMP SA ---------
09	--------- Creating ISAKMP SA id ---------
10	--------- Unable to make ISAKMP SA ---------
11	--------- Creating SA IN ---------
12	--------- Creating SA OUT ---------
13	--------- Deleting AL ENTRY ---------
14	--------- Deleting SA IN ---------
15	--------- Deleting SA OUT ---------
16	--------- KeepAlive Deleting SA ---------
17	--------- Purgetime SA NEG ---------

18	--------- Purgetime SA NEG II ---------
19	--------- Isakmp SA negotiating I ---------
20	--------- Negotiation on Phase I. Phase II not allowed ---------
21	--------- Invalid NegII or Isakmp SA ---------
22	--------- Initiated Renegotiation Timer for SA OUT ---------
23	--------- Renegotiation Timer for SA OUT expired ---------
24	--------- Attempt to renegotiate delayed ---------
25	--------- Matching CRL ---------
26	--------- Invalid ID information ---------
27	--------- No Dir Info in SA ---------
28	--------- Isakmp SA negotiating II ---------
29	--------- Negotiation from banned host stopped ---------
30	--------- DPD Deleting SAs ---------
31	--------- Max tunnel supported reached ---------
32	--------- Renegotiation using DNAT ---------
33	--------- Lifetime changed ---------
34	--------- Searching Backup Peer ---------
35	--------- Renegotiation Timer for SA NEG expired ---------
36	--------- Local Starting Neg ---------
37	--------- Remote Starting Neg ---------
38	--------- Local Starting BackUp Neg ---------
39	--------- Remote Starting BackUp Neg ---------
40	------!!! CONNECTED !!!------
100	*** Any Notify Message ***

Below you can see an example where the negotiation process register is interrupted when the message is 36. After executing the **stop-on-message** command, all the ISAKMP SAs are eliminated so a new negotiation is produced; this happens if there is IPSec traffic (not permanent SAs). Listing the negotiation process you can see how the register stops when the selected message occurs.

Example:

```
IPSec+stop-on-message 36
Activated stop on message number 36

IPSec+clear sa negotiation all
 All IPSec connections cleared

IPSec+list negotiation all
210.210.210.11 210.210.210.12: (12:21:10)
              (* 09:--------- Creating ISAKMP SA id ---------)(# 935067731(0x37b
c0053))(* 07:--------- Deleting ISAKMP NEG ---------)(# 66(0x42))
              (* 36:--------- Local Starting Neg ---------)
0.0.0.0: (12:21:11)
**** REGISTRY STOPPED BY MESSAGE NUMBER 36 ****

IPSec+
```

2.3. Certificates Monitoring Commands

Command	Function
? (HELP)	Lists the commands or their available options.
crl	Enters the CRLs monitoring menu.
list	Lists the device's certificates.
scep	Enters the SCEP monitoring menu.
exit	Exits the certificates monitoring menu.

a) crl

Enters the CRLs monitoring menu. The commands available in this menu are as follows:

Command	Function
? (HELP)	Lists the commands or their available options.
list	Lists the device's CRSs.
exit	Exits the certificates monitoring menu.

- *list*

Command	Function
existent	Lists the non-volatile memory CRLs in the device.
loaded	Lists the active CRLs.

- *list existent*

Displays the CRL lists that exist in the non-volatile memory CRLs in the device.

Example:

```
CRL monit>list existent
 A:                    TELDAT.CRL       1273   05/19/11   17:10   Flash
 A:                   WIN2008.CRL       1034   05/26/11   15:28   Flash
```

- *list loaded*

Displays the active CRL lists.

Example:

```
CRL monit>list loaded
Name
----
WIN2008.CRL
```

b) list

Command	Function
loaded-certificates	Lists the active certificates in the device.
disk-certificates	Lists the active certificates in the disk.
config-certificates	Lists the active certificates in the configuration.

- *list loaded-certificates*

Displays the active certificates in the device and their status.

Example:

```
CERTIFICATES monit+list loaded-certificates

WIN2008.CER (from config)

 Issuer: A:WIN2008.CER
 Status:
               -cn=jorge,dc=pruebas,dc=com
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
 Fingerprint: F0 3B 4D 68 F6 B6 24 51 46 EB 91 7B AB 9C 91 3D

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

TELDAT08.CER (from config)

 Issuer: A:WIN2008.CER
 Status:
               -cn=routerjose,dc=com,dc=pruebas
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
 Fingerprint: E4 A1 80 6B 72 5A AC 3B 73 4F BA 23 92 03 CD E8

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

WIN200EN.CER (from config)

 Issuer: A:WIN2008.CER
 Status:
               -cn=integrate,ou=international,o=teldat,l=madrid,s=madrid,c=ES
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
 Fingerprint: D9 17 8C 69 CF AE DE 00 E2 E1 41 EF 6F 75 9F E6

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

WIN200OF.CER (from config)

 Issuer: A:WIN2008.CER
Status:
               -cn=integrate,ou=international,o=teldat,l=madrid,s=madrid,c=ES
               -Without Private Key
               -CA certificate: A:WIN2008.CER
               -Status:Signature ok.
 Fingerprint: 3D 6D 83 14 54 A3 6D 44 49 9C 4C 9F B0 B6 50 D6

-------------------------------------------
```

- *list disk-certificates*

Displays the certificates saved in the disk.

Example:

```
CERTIFICATES monit+list disk-certificates
 A:                   CACHAIN.CER       4076   05/11/11   16:33   Flash
 A:                    TMXENR.CER       1492   05/17/04   13:44   Flash
 A:                   SECTEST.CER       1248   05/17/04   13:47   Flash
 A:                  CENTRAL2.CER       1706   05/17/04   13:48   Flash
 A:                       OF8.CER       1650   05/17/04   13:48   Flash
 A:                   STROUTE.CER       1200   05/17/04   13:48   Flash
```

- *list config-certificates*

Displays the certificates saved in the configuration.

Example:

```
CERTIFICATES monit+list config-certificates
ROUTER.CER
CACHAIOF.CER
CACHAIEN.CER
CACHAIN.CER
WIN200OF.CER
WIN200EN.CER
WIN2008.CER
TELDAT08.CER
```

c) scep

Enter in the SCEP monitoring menu. The commands available in this menu are as follows:

Command	Function
? (HELP)	Lists the commands or their available options.
ca-chain-install	Installs the chain of certificates up to the root CA.
capabilities	Displays the commands supported by the server.
enroll	Executes the *Enroll* protocol for an SCEP group.
install-ca	Executes the *Install* protocol for an SCEP group.
list	Lists the status of the SCEP groups.
next-ca-install	Installs the renewed CA certificates.
exit	Exits the certificates monitoring menu.

- *ca-chain-install, capabilities, enroll, install-ca, next-ca-install*

The behavior of these commands is the same as those explained in section 5.5 Obtaining Certificates through SCEP.

- *list*

Displays the status of the SCEP groups.

Example:

```
Group: 2, enrolltime: 10 min, timeleft: 5 min, status: IDLE
    URL: 192.168.213.119, CGI_PATH: /certsrv/mscep/mscep.dll
    CA: win2008
        Key Encipherment Cert (Encryption): A:WIN200EN.CER
        Digital Signature Cert (Enrollment): A:WIN200OF.CER
    There is a valid certificate: A:TELDAT_C.CER
```

2.4. IPSecFT monitoring commands

Access the IPSecFT monitoring submenu from the IPSec monitoring menu by entering "fault-tolerant".

```
*p 3
Console Operator


+protocol ip

-- IP protocol monitor --
```

```
IP+ipsec

-- IPSec protocol monitor --

IPSec+fault-tolerant

-- Fault tolerant IPSec recovery monitor --
Router1 IPSecFT monitor+
```

The following commands are found in this submenu:

Command	Function
? (help)	Lists the available commands or their options.
list	Lists information on the protocol.
clear	Resets the protocol counters.
exit	Exits the IPSecFT monitoring menu.

a) list

Lists the IPSecFT information. The options for this command are:

Command	Function
? (help)	Lists the available commands or their options.
all	Lists all the available information for the IPSecFT.
backup-task	Lists information on the IPSecFT backup tasks.
local-tunnels	Lists the IPSec sessions that the IPSecFT locally manages.
main-task	Lists the information on the IPSecFT main task.
queue	Lists information on the IPSecFT local queue.
remote-tunnels	Lists the remotely managed IPSec sessions.

- *list all*

Lists all the available information on the protocol.

Example:

```
IPSecFT monitor+list all
Backup task state:
        Running:                  TRUE
        Config change pending:    FALSE
        Current connections:      1
        Accumulated connections:  2
        Unregistered connections: 0
                Connection 1:
                 Time since creation:     0d05h02m13s
                 Current state:           STANDBY
                 Local [address/port]:    192.168.212.219:52912
                 Remote [address/port]:   192.168.212.218:1025
                 Internal ID:             0x00000002
                 Session ID:              0x1636
                 Inactivity timeout:      500 milliseconds
                 Inherit condition:       VRRP
                 Last packet received:    180555
                 Number of tunnels:       0
                 Monitored IP 01:         192.168.3.225

List of remote tunnels:
Session ID:              0x1636
        Initialized:          TRUE
```

```
        Number of tunnels:    0

Main task state:
        Running:                              TRUE
        Suspended:                            FALSE
        Config change pending:                FALSE
        Accumulated conn retries:             7
        Accumulated ACK with error:           0
          Invalid length:                       0
          Invalid version:                      0
          Invalid type:                         0
          Invalid num seq:                      0
          'not add' flag:                       0
          'not del' flag:                       0
        Accumulated timeouts receiving data: 2
        Accumulated communication errors:    5
        Time since last conn retry:          0d05h02m15s
        Current state:                       STANDBY
        Session ID:                          0x0d6e
        Local [address/port]:                192.168.212.219:1030
        Remote [address/port]:               192.168.212.218:52912
        Inactivity timeout:                  500 milliseconds
        Keepalive period:                    100 milliseconds
        Last packet sent:                    180519
        Number of tunnels:                   1
        Monitored IP 01:                     192.168.3.225

List of local tunnels:
ID 0x84b83a40: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi 0x84b83a40
               action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1

Message queue:
        Initialized:             TRUE
        Queue size:              1000
        Available elements:      1000
        Used elements:           0
        Sent to queue:           243880
        Errors sending to queue: 0
        Queue full counter:      0
        Retrieved from queue:    243880
```

There are five large blocks providing different information. These are explained below:

- Block 1: "**Backup task state**". Provides information on the IPSecFT backup tasks.
  - "**Running**": Indicates if the backup is executing or not.
  - "**Config change pending**": Indicates if the backup task has a configuration change pending.
  - "**Current connections**": Number of active backup tasks.
  - "**Accumulated connections**": Number of accumulated backup sessions since the last reset.
  - "**Unregistered connections**": Backup tasks that could not be registered.
    - "**Connection X**": The information following this refers to backup task number X.
      - "**Time since creation**": Time since the backup task was created.
      - "**Current state**": Backup task current state.
      - "**Local [address/port]**": Source address/port the backup task has connected with for its corresponding main task.
      - "**Remote [address/port]**": Destination address/port the backup task has connected with for its corresponding main task.

- "**Internal ID**": Backup task internal ID.
- "**Session ID**": Backup task ID.
- "**Inactivity timeout**": Maximum inactivity time permitted.
- "**Inherit condition**": Condition to inherit the IPSec sessions.
- "**Last packet received**": Last IPSecFT packet received.
- "**Number of tunnels**": Number of IPSec session the backup task is managing.
- "**Monitored IP XX**": IP address number XX this backup task is monitoring.

- Block 2: "**List of remote tunnels**". Lists the IPSec sessions the backup tasks are managing.
  - "**Session ID**": The information following this refers to backup task with the indicated ID.
    - "**ID 0xXXXXXXXX**": Displays information on the IPSec session with the indicated ID.
  - "**Initialized**": Indicates if this list of IPSec sessions has initialized.
  - "**Number of tunnels**": Number of IPSec sessions managed by the session with the indicated ID.

- Bloque 3: "**Main task state**". Lists information on the main IPSecFT task.
  - "**Running**": Indicates if the main task is executing or not.
  - "**Suspended**": Indicates if the main task is suspended or not.
  - "**Config change pending**": Indicates if the main task has a configuration change pending.
  - "**Accumulated conn retries**": Main task connection retries with its corresponding backup task.
  - "**Accumulated ACK with error**": ACK received with error.
    - "**Invalid length**": ACK received with error due to packet length.
    - "**Invalid version**": ACK received with error due to the protocol version.
    - "**Invalid type**": ACK received with error due to the type of packet received.
    - "**Invalid num seq**": ACK received with error due to sequence number.
    - "'**not add' flag**": ACK received with error due to a notification stating the specified IPSec session could not be added.
    - "'**not del' flag**": ACK received with error due to a notification stating the specified IPSec session could not be deleted.
  - "**Accumulated timeouts receiving data**": Accumulated timeouts while waiting to received data from the backup task.
  - "**Accumulated communication errors**": Accumulated communication errors that have occurred.
  - "**Time since last conn retry**": Time elapsed since the last connection retry with the backup task.
  - "**Current state**": Current state of main task.
  - "**Session ID**": Main task ID.
  - "**Local [address/port]**": Source address/port used in the connection with the backup task.

- "**Remote [address/port]**": Destination address/port used to connect with the backup task.
- "**Inactivity timeout**": Maximum time permitted without receiving data from the backup task.
- "**Keepalive period**": Time waited for the next main task action.
- "**Last packet sent**": Number of packets sent.
- "**Number of tunnels**": Number of IPSec sessions the main task manages.
- "**Monitored IP XX**": IP address number XX this main task is monitoring.

- Block 4: "**List of local tunnels**". Lists the IPSec session the main task is managing.
  - "**ID 0xXXXXXXXX**": Displays information on the IPSec session with the indicated ID.
  - "**Initialized**": Indicates if this list of IPSec session has initialized.
  - "**Number of tunnels**": Number of IPSec sessions the main task manages.

- Block 5 "**Message queue**": Lists the IPSecFT message queue information.
  - "**Initialized**": Indicates if the message queue is initialized or not.
  - "**Queue size**": Indicates the size of the message queue.
  - "**Available elements**": Number of available elements in the message queue.
  - "**Used elements**": Number of used elements in the message queue.
  - "**Sent to queue**": Number of elements sent to the message queue.
  - "**Errors sending to queue**": Number of errors sending elements to the queue.
  - "**Queue full counter**": Number of times that the queue is full when trying to enter an element.
  - "**Retrieved from queue**": Number of elements retrieved from the queue.

- *list backup-task*

Displays information on the backup tasks in IPSecFT.

Example:

```
IPSecFT monitor+list backup-task
Backup task state:
        Running:                 TRUE
        Config change pending:   FALSE
        Current connections:     1
        Accumulated connections: 2
        Unregistered connections: 0
                Connection 1:
                 Time since creation:     0d05h42m46s
                 Current state:           STANDBY
                 Local [address/port]:    192.168.212.219:52912
                 Remote [address/port]:   192.168.212.218:1025
                 Internal ID:             0x00000002
                 Session ID:              0x1636
                 Inactivity timeout:      500 milliseconds
                 Inherit condition:       VRRP
                 Last packet received:    204781
                 Number of tunnels:       0
                 Monitored IP 01:         192.168.3.225
```

For further information on the meaning of each field, please see the "list all" monitoring command example in this section.

- *list local-tunnels [Filter]*

Displays information on the IPSec sessions the main task is managing and that match the indicated filter.

Example:

```
IPSecFT monitor+list local-tunnels
List of local tunnels:
ID 0x84b83a40: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi 0x84b83a40
               action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1
```

In this example there is no specified filter so all the IPSec sessions the main task is managing are shown.

Example:

```
IPSecFT monitor+list local-tunnels 10.10.1.7
List of local tunnels:
ID 0x04483a10: ep_src 192.168.219.225(0), ep_dest 10.10.1.7(0), spi 0x04483a10
               action permit, src 192.168.212.0/23, dst 10.10.1.7/32
ID 0x342c3c50: ep_src 192.168.219.225(0), ep_dest 10.10.1.70(0), spi 0x342c3c50
               action permit, src 192.168.212.0/23, dst 10.10.1.70/32
ID 0x14483a01: ep_src 192.168.219.225(0), ep_dest 10.10.1.71(0), spi 0x14483a01
               action permit, src 192.168.212.0/23, dst 10.10.1.71/32
ID 0x74283ea0: ep_src 192.168.219.225(0), ep_dest 10.10.1.72(0), spi 0x74283ea0
               action permit, src 192.168.212.0/23, dst 10.10.1.72/32
ID 0x84ba3ab4: ep_src 192.168.219.225(0), ep_dest 10.10.1.73(0), spi 0x84ba3ab4
               action permit, src 192.168.212.0/23, dst 10.10.1.73/32
        Initialized:          TRUE
        Number of tunnels:    650
```

In this example filter "10.10.1.7" has been specified. Here all the IPSec sessions the main task is managing and that match the said filter are shown: in this case 5 out of 650 sessions.

For further information on the meaning of each field, please see the "list all" monitoring command example in this section.

- *list main-task*

Displays information on the IPSecFT main task.

Example:

```
IPSecFT monitor+list main-task
Main task state:
        Running:                              TRUE
        Suspended:                            FALSE
        Config change pending:                FALSE
        Accumulated conn retries:             7
        Accumulated ACK with error:           0
          Invalid length:                       0
          Invalid version:                      0
          Invalid type:                         0
          Invalid num seq:                      0
          'not add' flag:                       0
          'not del' flag:                       0
        Accumulated timeouts receiving data: 2
        Accumulated communication errors:    5
        Time since last conn retry:          0d05h46m28s
        Current state:                       STANDBY
        Session ID:                          0x0d6e
        Local [address/port]:                192.168.212.219:1030
```

```
        Remote [address/port]:              192.168.212.218:52912
        Inactivity timeout:                 500 milliseconds
        Keepalive period:                   100 milliseconds
        Last packet sent:                   206932
        Number of tunnels:                  1
        Monitored IP 01:                    192.168.3.225
```

For further information on the meaning of each field, please see the "list all" monitoring command example in this section.

- *list queue*

Displays information on the IPSecFT message queue.

Example:

```
IPSecFT monitor+list queue
Message queue:
        Initialized:             TRUE
        Queue size:              1000
        Available elements:      1000
        Used elements:           0
        Sent to queue:           273224
        Errors sending to queue: 0
        Queue full counter:      0
        Retrieved from queue:    273224
```

For further information on the meaning of each field, please see the "list all" monitoring command example in this section.

- *list remote-tunnels [Filter]*

Displays information on the IPSec session that the backup tasks are managing and match the indicated filter.

Example:

```
IPSecFT monitor+list remote-tunnels
List of remote tunnels:
Session ID:              0x0d6e
        ID 0x3caf53e8: ep_src 192.168.3.225(500), ep_dest 192.168.3.100(500), spi
0x3caf53e8
                       action permit, src 192.168.212.0/23, dst 10.10.2.0/24
        Initialized:          TRUE
        Number of tunnels:    1
```

In this example there is no specified filter so all the IPSec sessions the backup tasks are managing are shown.

Example:

```
IPSecFT monitor+list remote-tunnels 10.10.1.7
List of remote tunnels:
Session ID:              0xca8f
        ID 0x04483a10: ep_src 192.168.219.225(0), ep_dest 10.10.1.7(0), spi
0x04483a10
                       action permit, src 192.168.212.0/23, dst 10.10.1.7/32
        ID 0x342c3c50: ep_src 192.168.219.225(0), ep_dest 10.10.1.70(0), spi
0x342c3c50
                       action permit, src 192.168.212.0/23, dst 10.10.1.70/32
        ID 0x14483a01: ep_src 192.168.219.225(0), ep_dest 10.10.1.71(0), spi
0x14483a01
                       action permit, src 192.168.212.0/23, dst 10.10.1.71/32
```

```
        ID 0x74283ea0: ep_src 192.168.219.225(0), ep_dest 10.10.1.72(0), spi
0x74283ea0
                       action permit, src 192.168.212.0/23, dst 10.10.1.72/32
        ID 0x84ba3ab4: ep_src 192.168.219.225(0), ep_dest 10.10.1.73(0), spi
0x84ba3ab4
                       action permit, src 192.168.212.0/23, dst 10.10.1.73/32
        Initialized:          TRUE
        Number of tunnels:    650
```

In this example filter "10.10.1.7" has been specified. Here all the IPSec sessions the backup tasks are managing and that match the said filter are shown: in this case 5 out of 650 sessions in a single backup task.

For further information on the meaning of each field, please see the "list all" monitoring command example in this section.

b) clear

Resets the counters in the IPSecFT monitoring. The options for this command are as follows:

Command	Function
? (help)	Lists the commands or their available options.
all	Resets all the counters that allow this.
backup-task	Resets the counters relative to the backup tasks that permit this.
main-task	Resets the counters relative to the main task that permit this.
queue	Resets the IPSecFT message queue counters that permit this.

- *clear all*

Resets all the counters in the IPSecFT monitoring that permit this.

Example:

```
IPSecFT monitor+clear all
```

- *clear backup-task*

Resets the monitoring counters in IPSecFT relative to the backup tasks and that permit this.

Example:

```
IPSecFT monitor+clear backup-task
```

- *clear main-task*

Resets the monitoring counters in IPSecFT relative to the main task and that permit this.

Example:

```
IPSecFT monitor+clear main-task
```

- *clear queue*

Resets the monitoring counters in IPSecFT relative to the message queue and that permit this.

Example:

```
IPSecFT monitor+clear queue
```

2.5. Diagnosing problems in the IKE negotiation

In this section, we are going to give some typical example problems that often appear during IKE negotiation due to configuration errors. It is very important to know how to identify which phase the negotiation is in. To obtain this information, simply check the number associated to message header causing the error. If this is 0, this means that this is a phase 1 message and if it is distinct to zero, then it pertains to phase 2. The message producing the error usually is the one preceding the warning message indicating that an error has occurred. For example:

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encryp des)
  (hash sha)(grp desc 1)(auth rsa)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 57(0x39))(HDR 24343432)
  (notif isakmp no proposal chosen)
```

The message provoking the error was the one sent by 172.24.51.57 whose HDR has identifier 0. This means it is an error produced in the first phase of negotiation.

Another important piece of data is to know who initiated the negotiation, i.e. who was the *initiator*.

a) the device does not initiate the negotiation

Origin

The access control list has not been correctly configured.

This message is produced because the device could not match the packet, which should set off the negotiation, with an IPSec entry in the access control list.

Solution

Check the access control list parameters.

Addresses: Source and Destination. (Be careful with the subnets)
Mask.
Protocol.
Ports. Source and Destination.
Template: The corresponding dynamic Template must be mapped.

If you still cannot find the source of the error, check the result of the **LIST ACCESS OUT** monitoring command and check that the *hits* are increasing in the corresponding entry.

b) notif isakmp no proposal chosen. Phase 1

Initiator: `172.24.51.57`

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encryp des)
  (hash sha)(grp desc 1)(auth rsa)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 57(0x39))(HDR 0)
  (notif isakmp no proposal chosen)
```

Origin

The isakmp Template has not been correctly configured.

This message is produced because the device with address 172.24.78.15 has not been able to accept any of the proposals from device 172.24.51.57. In this phase of the negotiation, the proposals received are compared with those configured in the isakmp.

Solution

Check the isakmp Template parameters.

Authentication method: RSA_SIGNATURE, PRE-SHARED...
Encryption system: DES, TDES...
Authentication system: SHA1, MD5...
Type of lifetime: Seconds, Kbytes, both...
Group: 1 or 2.

c) notif isakmp payload malformed. Phase 1

Initiator: `172.24.51.57`

```
172.24.51.57: (HDR 0)(HDR sa)(prop 1 isakmp #1)(trans 1 id=1)(encryp des)
  (hash md5)(grp desc 1)(auth presh)(life sec)(duration 600)(vendor 14)
172.24.78.15:
  (* --------- Creating ISAKMP NEG ---------)(# 67(0x43))
  (* --------- Matching template ---------)(# 20(0x14))(HDR 0)(HDR sa)
  (prop 1 isakmp #1)(trans 1 id=1)(encryp des)(hash md5)(grp desc 1)(auth presh)
  (life sec)(duration 600)
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id none prot=148 port=9841)(# 0x3c068321)(HDR 75 0)
172.24.78.15: (HDR 0)(notif isakmp payload malformed)
```

Origin

The Pre-shared key has not been correctly configured.

This message has been produced because the device with address 172.24.78.15 has not been able to correctly decode the encrypted message sent by device 172.24.51.57. In fact, on analyzing the erroneous message, you can see that some strange parameters have been received: unknown identifier, with protocol and port distinct to those configured, followed by an unknown header, .hdr 75 0.

Solution

Check the Pre-shared key and the ip_address – key, hostname-key associations.

d) notif esp no proposal chosen. Phase 2

Initiator: `172.24.51.57`

```
172.24.51.57: (HDR 53da7bd5)(HDR hash)(HDR sa)(prop 1 esp #2)
  (# -786612676(0xd11d3e3c))(trans 1 id=des)(life sec)(duration 300)
  (life kbytes)(duration 100000)(encap tunnel)(auth alg md5)(trans 2 id=des)
  (life sec)(duration 300)(life kbytes)(duration 100000)(encap tunnel)
  (auth alg sha)(prop 2 ah #2)(# -786612676(0xd11d3e3c))(trans 1 id=md5)
  (life sec)(duration 300)(life kbytes)(duration 100000)(encap tunnel)
  (auth alg md5)(trans 2 id=sha)(life sec)(duration 300)(life kbytes)
  (duration 100000)(encap tunnel)(auth alg sha)(HDR nonce)
  (id addr4 prot=0 port=0)(# 0xac183339)(id addr4 prot=0 port=0)(# 0xac184e0f)
172.24.78.15:
  (* --------- Creating ISAKMP SA id ---------)(# -583852704(0xdd331d60))
  (HDR dd331d60)(HDR hash)(notif esp no proposal chosen)
```

Origin

The isakmp Template has not been correctly configured.

This message is produced because the device with address 172.24.78.15 has not been able to accept any of the proposals from device 172.24.51.57. In this phase of the negotiation, the proposals received are compared with those configured in the dynamic Template associated with the corresponding access control list.

Solution

Check the dynamic Template parameters.

Type of encapsulation: Tunnel or Transport.
Encryption system: DES, TDES...
Authentication system: SHA1, MD5...
Type of lifetime: Seconds, Kbytes, both...
PFS: Check that the remote device admits PFS.

e) notif esp invalid id inform. Phase 2

Initiator: `172.24.51.57`

```
172.24.78.15: (HDR 0)(id addr4 prot=17 port=500)(# 0xac184e0f)(HDR hash)
  (* --------- Creating ISAKMP SA id ---------)(# 785093687(0x2ecb9437))
172.24.51.57: (HDR 2ecb9437)(HDR hash)(HDR sa)(prop 1 esp #2)
  (# 291357516(0x115dc34c))(trans 1 id=des)(life sec)(duration 300)(life kbytes)
  (duration 100000)(encap tunnel)(auth alg md5)(trans 2 id=des)(life sec)
  (duration 300)(life kbytes)(duration 100000)(encap tunnel)(auth alg sha)
  (prop 2 ah #2)(# 291357516(0x115dc34c))(trans 1 id=md5)(life sec)
  (duration 300)(life kbytes)(duration 100000)(encap tunnel)(auth alg md5)
  (trans 2 id=sha)(life sec)(duration 300)(life kbytes)(duration 100000)
  (encap tunnel)(auth alg sha)(HDR nonce)(id addr4 prot=0 port=0)(# 0xac183339)
  (id addr4 prot=16 port=0)(# 0xac184e0f)
172.24.78.15:
  (* --------- Creating ISAKMP SA id ---------)(# 1537079449(0x5b9df899))
  (HDR 5b9df899)(HDR hash)(notif esp invalid id inform)
```

Origin

The access control list has not been correctly configured.

This message is produced when the device with address 172.24.78.15 has not been able to accept the client identifier from device 172.24.51.57 *(id addr4 prot=0 port=0)(# 0xac183339) (id addr4 prot=16 port=0)(# 0xac184e0f).* In this phase of the negotiation, the proposals of the received identifiers are compared with those configured in the access control list.

Solution

Check the access control list parameters.

Addresses: Source and Destination. (Be careful with the subnets)
Mask.
Protocol.
Ports. Source and Destination.
Template: The corresponding dynamic Template must be mapped.

f) notif isakmp invalid cert authority. Phase 1. Initiator A

Initiator: `172.24.78.15`

```
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 8)
172.24.78.15:
  (* --------- Creating ISAKMP SA ---------)(HDR 0)
  (notif isakmp invalid cert authority)
```

Origin

The isakmp Template has not been correctly configured.

This message is produced because the device with address 172.24.78.15 has not been able to find the CA configured in the corresponding isakmp Template.

Solution

Check the isakmp Template parameters.

Name of the CA.
Check that the CA name corresponds to a file in the device:

```
Router CERTIFICATES config>LIST EXIST
```

g) notif isakmp invalid cert authority. Phase 1. Initiator B

Initiator: `172.24.51.57`

```
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 6)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 7)(cert x509sig CERTREG 8)
  (HDR sig)(certreq x509sig CERTREG 9)
172.24.78.15: (HDR 0)(notif isakmp invalid cert authority)
```

Origin

The isakmp Template has not been correctly configured.

This message is produced because the device with address 172.24.78.15 has not been able to find a CA configured in any isakmp Template that corresponds to that of the received certificate, in the example CERTREG 9

Solution

Check the isakmp Template parameters and compare them with the command execution result.

```
Router IPSec>LIST CERTIFICATE_NUMBER 9
```

Name of the CA.
Check that the CA name corresponds to a file in the device:

```
Router CERTIFICATES config>LIST EXIST
```

h) notif isakmp invalid cert. Phase 1

Initiator: `172.24.51.57`

```
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 14)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 15)(cert x509sig CERTREG 16)
  (HDR sig)(certreq x509sig CERTREG 17)
172.24.78.15: (HDR 0)(notif isakmp invalid cert)
```

Origin

The received certificate is invalid.

Solution

Check that the received certificate is correct with the command:

```
Router IPSec>LIST CERTIFICATE_NUMBER 16
```

Check the parameters for:
Validity period.
The Issuer corresponds with the required CA.

```
Router IPSec>LIST CERTIFICATE_NUMBER 14
```

The certificate may be incorrectly signed.

i) notif isakmp cert unavailable. Phase 1

Initiator: `172.24.51.57`

```
172.24.51.57: (HDR 0)(HDR keyx)(HDR nonce)
172.24.78.15: (HDR 0)(HDR keyx)(HDR nonce)(certreq x509sig CERTREG 0)
  (* --------- Creating ISAKMP SA ---------)
172.24.51.57: (HDR 0)(id der_dn port=0 CERTREG 1)(cert x509sig CERTREG 2)
  (HDR sig)(certreq x509sig CERTREG 3)
172.24.78.15: (HDR 0)(notif isakmp cert unavailable)
```

Origin

There is no user certificate loaded for device 172.24.78.15 to send to the 172.24.51.57 end.

Solution

Check that there is a loaded certificate for the required CA.

First of all check which CA is required.

```
Router IPSec>LIST CERTIFICATE_NUMBER 3
```

If the required CA coincides with that sent. Execute a list of the isakmp Templates and check the result. This should indicate what the problem is.

If the required CA does not coincide with that sent, search in the CERTIFICATES menu to ensure there does exist a loaded certificate pertaining to this CA.

```
Router CERTIFICATES config>LIST LOADED PRINT ISSUER <certificate_name>
```