Virtual Private Networks

Administration Guide

Version R70

Check Point®
SOFTWARE TECHNOLOGIES LTD.

703333 April 23, 2009

TRADEMARKS:

Contents

Preface

Introduction to VPN Technology

Chapter 1 **Overview**

Chapter 2 **IPSEC & IKE**

Chapter 4

Introduction to Site to Site VPN

Chapter 5

Public Key Infrastructure

Site-to-Site VPN

Chapter 8 Route Injection Mechanism

Chapter 9 Wire Mode

Chapter 10 Directional VPN Enforcement

Chapter 11 Link Selection

Remote Access VPN

Chapter 14 Introduction to Remote Access VPN

Chapter 15 Office Mode

Chapter 17 Endpoint Connect

Chapter 18 Endpoint Connect API

Chapter 19 SecureClient Mobile

Chapter 29

Userc.C and Product.ini Configuration Files

Chapter 30

SSL Network Extender

Appendix

Appendix B

Converting a Traditional Policy to a Community Based Policy

Appendix C

VPN Shell

Preface

In This Chapter

Who Should Use This Guide

This guide is intended for administrators responsible for maintaining network security within an enterprise, including policy management and user support.

This guide assumes a basic understanding of

- System administration.
- The underlying operating system.
- Internet protocols (IP, TCP, UDP etc.).

Summary of Contents

This guide describes the VPN components of a Check Point Security Gateway. It contains the following sections and chapters:

Section 1: Introduction to VPN Technology

This section describes the basic components of a VPN and provides the background for the technology that comprises the VPN infrastructure.

Chapter	Description
Chapter 1, “Overview”	Provides an overview of Check Point’s solution for VPN.
Chapter 2, “IPSEC & IKE”	Description of encryption modes used to transport packets securely using VPN tunnels.
Chapter 5, “Public Key Infrastructure”	Public Key Infrastructure is a system of certificate authorities that verify and authenticate the validity of each party exchanging information.

Section 2: Site-to-Site VPN

This section explains how to ensure secure communication between gateway modules.

Chapter	Description
Chapter 4, “Introduction to Site to Site VPN”	An introduction to the basics of VPN’s between gateways and VPN communities.
Chapter 5, “Domain Based VPN”	Domain Based VPN is a method of controlling how VPN traffic is routed between gateway modules and remote access clients within a community.
Chapter 6, “Route Based VPN”	Route Based VPN is a method of controlling how VPN traffic is routed between gateways using VPN Tunnel Interfaces.
Chapter 7, “Tunnel Management”	Tunnel Management descibes the various aspects of VPN sharing and Permanent Tunnels.

Chapter	Description
Chapter 8, "Route Injection Mechanism"	Route Injection Mechanism (RIM) enables a Check Point Security Gateway to use dynamic routing protocols to propagate the encryption domain of a Check Point Security Gateway to the internal network and then initiate back connections.
Chapter 9, "Wire Mode"	Describes how Wire Mode improves connectivity by allowing existing connections to fail over successfully by bypassing firewall enforcement.
Chapter 10, "Directional VPN Enforcement"	Explains how to control the direction of VPN traffic between gateways.
Chapter 11, "Link Selection"	Explanation of how the Link Selection feature is used to determine which interface is used for incoming and outgoing VPN traffic as well as the best possible path between gateway modules.
Chapter 12, "Multiple Entry Point VPNs"	Description of how the Multiple Entry Point (MEP) feature provides a high availability and load sharing solution for VPN connections between peer gateways.
Chapter 13, "Traditional Mode VPNs"	Explanation of Traditional Mode VPNs and how to configure.

Section 3: Remote Access VPN

This section explains how to ensure secure communication between gateway modules and remote access clients.

Chapter	Description
Chapter 14, "Introduction to Remote Access VPN"	Introduction to VPN connections between gateways and remote users.
Chapter 15, "Office Mode"	Office Mode enables a Check Point Security Gateway to assign a remote client an IP address.
Chapter 16, "SecuRemote/SecureClient"	SecuRemote/SecureClient is a method that allows you to connect to your organization in a secure manner, while at the same time protecting your machine from attacks that originate on the Internet.
Chapter 17, "Endpoint Connect"	Covers Endpoint connect client, server-side configuration
Chapter 18, "Endpoint Connect API"	Covers the Endpoint Application Programming Interface (API)
Chapter 19, "SecureClient Mobile"	SecureClient Mobile is a client for mobile devices that includes a VPN and a firewall. SecureClient Mobile's VPN is based on SSL (HTTPS) tunneling and enables handheld devices to securely access resources behind Check Point gateways.
Chapter 20, "Packaging SecureClient"	Using one of the two available packaging tools, enables the administrator to create pre-configured SecureClient and SecuRemote packages. Users can then install the package without being required to specify configuration details, ensuring that users cannot inadvertently misconfigure their SecureClient and SecuRemote software.
Chapter 21, "Desktop Security"	Description of how SecureClient protects remote clients by enforcing a Desktop Security Policy on the remote client.

Chapter	Description
Chapter 22, "Layer Two Tunneling Protocol (L2TP) Clients"	Check Point Security Gateways can create VPNs with a number of third party IPSec clients. This chapter focuses on the Microsoft IPSec/L2TP client.
Chapter 23, "Secure Configuration Verification"	Secure Configuration Verification (SCV) enables the administrator to monitor the configuration of remote computers, to confirm that the configuration complies with the organization's Security Policy, and to block connectivity for machines that do not comply.
Chapter 24, "VPN Routing - Remote Access"	Understanding how VPN Routing provides a way of controlling how VPN traffic is directed. between gateway modules and remote access clients.
Chapter 25, "Link Selection for Remote Access Clients"	Explanation of how the Link Selection feature is used to determine which interface is used for incoming and outgoing VPN traffic as well as the best possible path between gateway modules and remote access clients.
Chapter 26, "Using Directional VPN for Remote Access"	Explains how to control the direction of VPN traffic between gateways and remote access clients.
Chapter 27, "Remote Access Advanced Configuration"	Understanding more complex remote access scenarios.
Chapter 28, "Multiple Entry Point for Remote Access VPNs"	Description of how the Multiple Entry Point (MEP) feature provides a high availability and load sharing solution for VPN connections between peer gateways and remote access clients.
Chapter 29, "Userc.C and Product.ini Configuration Files"	How to edit the Userc.c and Product.ini files to customize SecuRemote/SecureClient.

Chapter	Description
Chapter 30, “SSL Network Extender”	Contains an introduction of the SSL Network Extender and the advantages it has for remote access clients.
Chapter 31, “Resolving Connectivity Issues”	Provides information of some of the challenges remote access clients face when connecting and various Check Point solutions.
Chapter 32, “Clientless VPN”	Explanation of how Clientless VPN provides secure SSL-based communication between clients when VPN technology is not available.

Appendices

This guide contains the following appendices:

Appendix	Description
Chapter A, “VPN Command Line Interface”	A list of CLI command lines related to VPN.
Chapter B, “Converting a Traditional Policy to a Community Based Policy”	Backround to both traditoinal and simplified modes as well as instructions for converting policies.
Chapter C, “VPN Shell”	Provides all the commands and arguments used for VTI’s using the VPN Shell.

Related Documentation

This release includes the following documentation:

TABLE P-1 Check Point Documentation

Title	Description
Internet Security Installation and Upgrade Guide	Contains detailed installation instructions for Check Point network security products. Explains the available upgrade paths from versions R60-65 to the current version.
High-End Installation and Upgrade Guide	Contains detailed installation instructions for the Provider-1 and VSX products, including hardware and software requirements and licensing requirements. Explains all upgrade paths for Check Point products specifically geared towards upgrading to the current version.
Security Management Server Administration Guide	Explains Security Management solutions. This guide provides solutions for control over configuring, managing, and monitoring security deployments.
Firewall Administration Guide	Describes how to control and secure network access and VoIP traffic; how to use integrated web security capabilities; and how to optimize Application Intelligence with capabilities such as Content Vectoring Protocol (CVP) applications, URL Filtering (UFP) applications.
IPS Administration Guide	Describes how to use IPS to protect against attacks.
Virtual Private Networks Administration Guide	Describes the basic components of a VPN and provides the background for the technology that comprises the VPN infrastructure.

TABLE P-1 Check Point Documentation (continued)

Title	Description
Eventia Reporter Administration Guide	Explains how to monitor and audit traffic, and generate detailed or summarized reports in the format of your choice (list, vertical bar, pie chart etc.) for all events logged by Check Point Security Gateways, SecureClient and IPS.
SecurePlatform/ SecurePlatform Pro Administration Guide	Explains how to install and configure SecurePlatform. This guide will also teach you how to manage your SecurePlatform machine and explains Dynamic Routing (Unicast and Multicast) protocols.
Provider-1/SiteManager-1 Administration Guide	Explains the Provider-1 security management solution. This guide provides details about a three-tier, multi-policy management architecture and a host of Network Operating Center oriented features that automate time-consuming repetitive tasks common in Network Operating Center environments.

More Information

- For additional technical information about Check Point products, consult Check Point's SecureKnowledge at http://support.checkpoint.com.
- To view the latest version of this document in the Check Point User Center, go to: http://support.checkpoint.com.

Feedback

Check Point is engaged in a continuous effort to improve its documentation. Please help us by sending your comments to:

cp_techpub_feedback@checkpoint.com

Introduction to VPN Technology

Chapter 1
Overview

In This Chapter

The Connectivity Challenge

With the explosive growth in computer networks and network users, IT managers are faced with the task of consolidating existing networks, remote sites, and remote users into a single secure structure.

Branch offices require connectivity with other branch offices as well as the central organization. Remote users require enhanced connectivity features to cope with today's changing networking environments. New partnership deals mean business to business connections with external networks.

Typically, consolidation needs to take place using existing infrastructure. For many, this means connectivity established via the Internet as opposed to dedicated leased lines. Remote sites and users must be unified while at the same time maintaining high levels of security. Once connectivity has been established, the connections must *remain* secure, offer high levels of privacy, authentication, and integrity while keeping costs low.

In addition, only legitimate traffic must be allowed to enter the internal network. Possibly harmful traffic must be inspected for content. Within the internal network, different levels of access must also exist so that sensitive data is only available to the right people.

The Basic Check Point VPN Solution

In This Section:

Virtual Private Networking technology leverages existing infrastructure (the Internet) as a way of building and enhancing existing connectivity in a secure manner. Based on standard Internet secure protocols, VPN implementation enables secure links between special types of network nodes: Check Point Security Gateways. Site to Site VPN ensures secure links between gateways. Remote Access VPN ensures secure links between gateways and remote access clients.

Check Point's Security gateway is an integrated software solution that provides connectivity to corporate networks, remote and mobile users, branch offices and business partners on a wide range of open platforms and security appliances. shows

Check Point Security Gateways integrate access control, authentication, and encryption to guarantee the security of network connections over the public Internet.

A typical deployment places a Check Point Security Gateway connecting the corporate network (from the Internet), and remote access software on the laptops of mobile users. Other remote sites are guarded by additional Check Point Security Gateways and communication between all components regulated by a strict security policy.

VPN Components

VPN is composed of:

- *VPN endpoints*, such as gateways, clusters of gateways, or remote client software (for mobile users) which negotiate the VPN link.
- *VPN trust entities*, for example the Check Point Internal Certificate Authority. The ICA is part of the Check Point suite used for establishing trust for SIC connections between gateways, authenticating administrators and third party servers. The ICA provides certificates for internal gateways and remote access clients which negotiate the VPN link.

- *VPN Management tools*. Security Management server and SmartDashboard. SmartDashboard is the SmartConsole used to access the Security Management server. The VPN Manager is part of SmartDashboard. SmartDashboard enables organizations to define and deploy Intranet, and remote Access VPNs.

Understanding the Terminology

A number of terms are used widely in Secure VPN implementation, namely:

- **VPN**. A private network configured within a public network, such as the Internet
- **VPN Tunnel**. An exclusive channel or encrypted link between gateways.
- **VPN Topology**. The basic element of VPN is the link or encrypted tunnel. Links are created between gateways. A collection of links is a *topology*. The topology shows the layout of the VPN. Two basic topologies found in VPN are *Mesh* and *Star*.
- **VPN Gateway**. The endpoint for the encrypted connection, which can be any peer that supports the IPSec protocol framework. Gateways can be single standalone modules or arranged into clusters for "high availability" and "load sharing".
- **VPN Domain**. A group that specifies the hosts or networks for which encryption of IP datagrams is performed. A VPN gateway provides an entrance point to the VPN Domain.
- **Site to Site VPN**. Refers to a VPN tunnel between gateways.
- **Remote Access VPN**. Refers to remote users accessing the network with client software such as SecuRemote/SecureClient or third party IPSec clients. The Check Point Security Gateway provides a *Remote Access Service* to the remote clients.
- **Encryption algorithm**. A set of mathematically expressed processes for rendering information into a meaningless form, the mathematical transformations and conversions controlled by a special key. In VPN, various encryption algorithms such as 3DES and AES ensure that only the communicating peers are able to understand the message.
- **Integrity**. Integrity checks (via hash functions) ensure that the message has not been intercepted and altered during transmission.
- **Trust**. Public key infrastructure (PKI), certificates and certificate authorities are employed to establish trust between gateways. (In the absence of PKI, gateways employ a pre-shared secret.)

- **IKE & IPSec**. Secure VPN protocols used to manage encryption keys, and exchange encrypted packets. IPSec is an encryption technology framework which supports several standards to provide authentication and encryption services of data on a private or public network. IKE (Internet Key Exchange) is a key management protocol standard. IKE enhances IPSec by providing additional features, flexibility, and ease of configuration.

Site to Site VPN

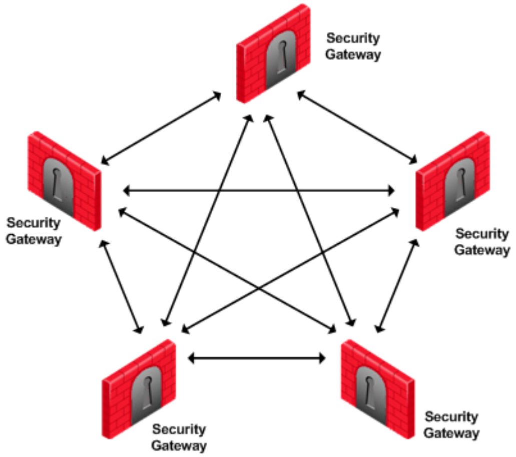
At the center of VPN is the encrypted tunnel (or VPN link) created using the IKE/IPSec protocols. The two parties are either Check Point Security Gateways or remote access clients. The peers negotiating a link first create a trust between them. This trust is established using certificate authorities, PKI or pre-shared secrets. Methods are exchanged and keys created. The encrypted tunnel is established and then maintained for multiple connections, exchanging key material to refresh the keys when needed. A single gateway maintains multiple tunnels simultaneously with its VPN peers. Traffic in each tunnel is encrypted and authenticated between the VPN peers, ensuring integrity and privacy. Data is transferred in bulk via these virtual-physical links.

VPN Communities

There are two basic community types - Mesh and Star. A topology is the collection of enabled VPN links in a system of gateways, their VPN domains, hosts located behind each gateway and the remote clients external to them.

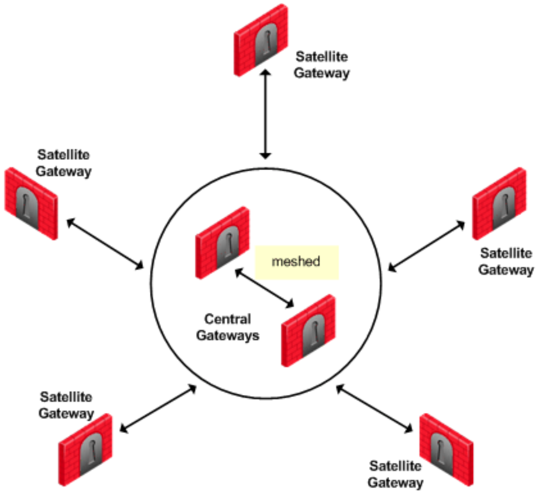
In a Mesh community, every gateway has a link to every other gateway, as shown in Figure 1-1:

Figure 1-1 Check Point Security Gateways in a Mesh community



In a Star community, only gateways defined as Satellites (or “spokes”) are allowed to communicate with a central gateway (or “Hub”) but not with each other:

Figure 1-2 Check Point Security Gateways in a Star community



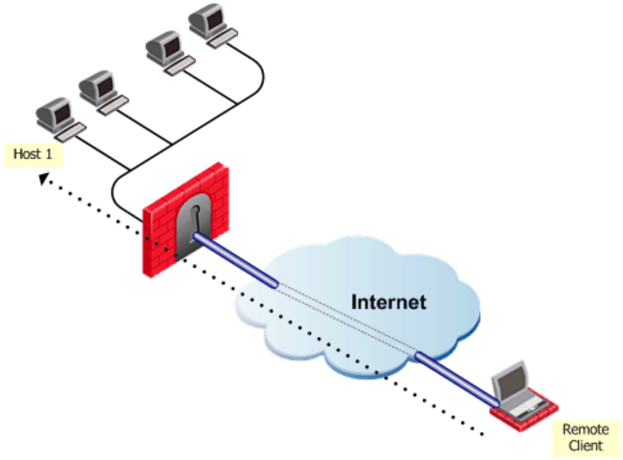
As shown in Figure 1-2, it is possible to further enhance connectivity by meshing central gateways. This kind of topology is suitable for deployments involving Extranets that include networks belonging to business partners.

Remote Access VPN

Whenever users access the organization from remote locations, it is essential that the usual requirements of secure connectivity be met but also the special demands of remote clients.

SecuRemote/SecureClient extends VPN functionality to remote users, enabling users to securely communicate sensitive information to networks and servers over the VPN tunnel, using both dial-up (including broadband connections), and LAN (and wireless LAN) connections. Users are managed either in the internal database of the Check Point Security Gateway or via an external LDAP server.

Figure 1-3 Remote Client to Host behind Gateway



In Figure 1-3, the remote user initiates a connection to the gateway. Authentication takes place during the IKE negotiation. Once the user's existence is verified, the gateway then authenticates the user, for example by validating the user's certificate. Once IKE is successfully completed, a tunnel is created; the remote client connects to Host 1.

Chapter 2

IPSEC & IKE

In This Chapter

Overview

In symmetric cryptographic systems, both communicating parties use the same key for encryption and decryption. The material used to build these keys must be exchanged in a secure fashion. Information can be securely exchanged only if the key belongs exclusively to the communicating parties.

The goal of the *Internet Key Exchange* (IKE) is for both sides to independently produce the same symmetrical key. This key then encrypts and decrypts the regular IP packets used in the bulk transfer of data between VPN peers. IKE builds the VPN tunnel by authenticating both sides and reaching an agreement on methods of encryption and integrity. The outcome of an IKE negotiation is a *Security Association* (SA).

This agreement upon keys and methods of encryption must also be performed securely. For this reason IKE is composed of two phases. The first phase lays the foundations for the second.

Diffie-Hellman (DH) is that part of the IKE protocol used for exchanging the material from which the symmetrical keys are built. The Diffie-Hellman algorithm builds an encryption key known as a “shared secret” from the private key of one party and the public key of the other. Since the IPSec symmetrical keys are derived from this DH key shared between the peers, at no point are symmetric keys actually exchanged.

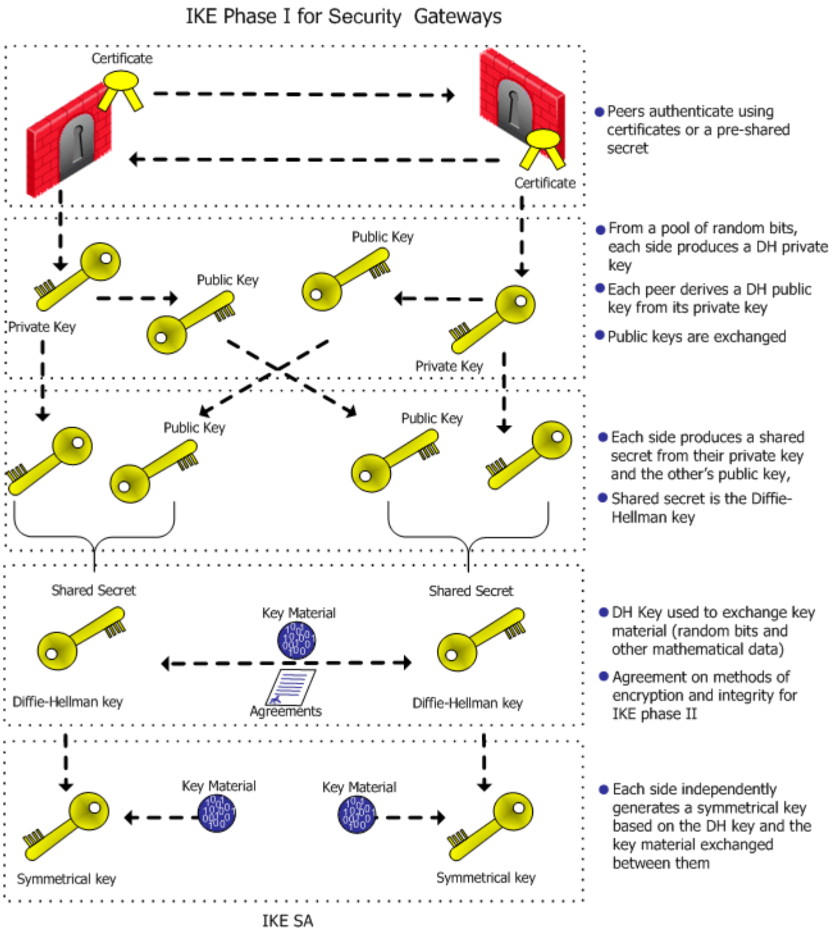
IKE Phase I

During IKE Phase I:

- The peers authenticate, either by certificates or via a pre-shared secret. (More authentication methods are available when one of the peers is a remote access client.)
- A Diffie-Hellman key is created. The nature of the Diffie-Hellman protocol means that both sides can independently create the shared secret, a key which is known only to the peers.
- Key material (random bits and other mathematical data) as well as an agreement on methods for IKE phase II are exchanged between the peers.

In terms of performance, the generation of the Diffie Hellman Key is slow and heavy. The outcome of this phase is the IKE SA, an agreement on keys and methods for IKE phase II. Figure 2-1 illustrates the process that takes place during IKE phase I but does not necessarily reflect the actual order of events.

Figure 2-1 IKE phase I

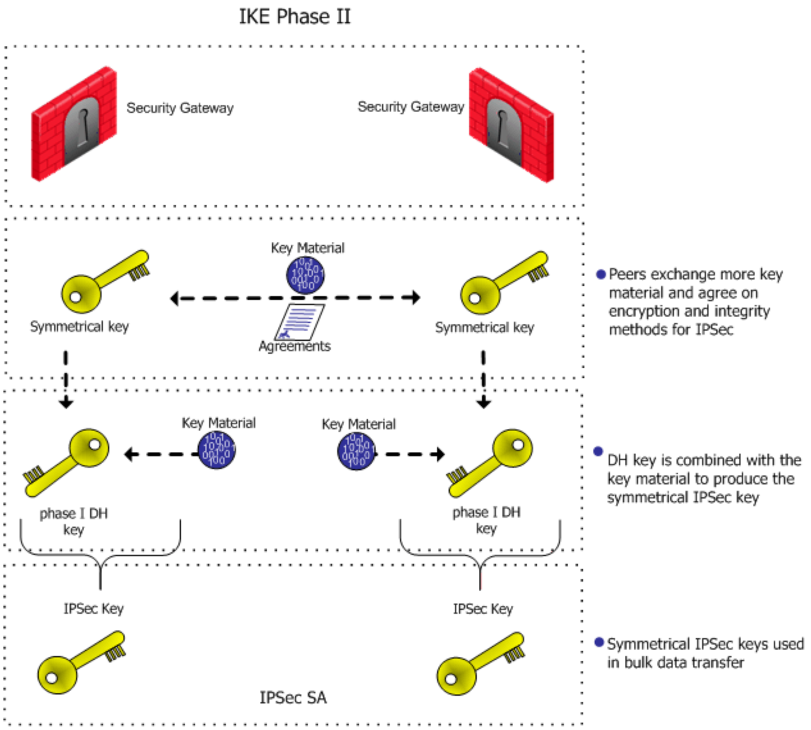


- Peers authenticate using certificates or a pre-shared secret
- From a pool of random bits, each side produces a DH private key
- Each peer derives a DH public key from its private key
- Public keys are exchanged
- Each side produces a shared secret from their private key and the other's public key,
- Shared secret is the Diffie-Hellman key
- DH Key used to exchange key material (random bits and other mathematical data)
- Agreement on methods of encryption and integrity for IKE phase II
- Each side independently generates a symmetrical key based on the DH key and the key material exchanged between them

IKE Phase II (Quick mode or IPSec Phase)

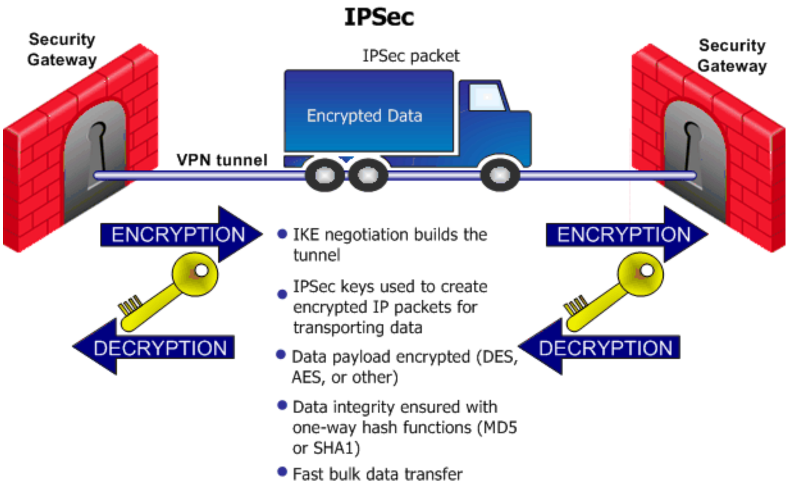
IKE phase II is encrypted according to the keys and methods agreed upon in IKE phase I. The key material exchanged during IKE phase II is used for building the IPSec keys. The outcome of phase II is the IPSec Security Association. The IPSec SA is an agreement on keys and methods for IPSec, thus IPSec takes place according to the keys and methods agreed upon in IKE phase II.

Figure 2-2 IKE Phase II



- Peers exchange more key material and agree on encryption and integrity methods for IPSec
- DH key is combined with the key material to produce the symmetrical IPSec key
- Symmetrical IPSec keys used in bulk data transfer

Once the IPSec keys are created, bulk data transfer takes place:



- IKE negotiation builds the tunnel
- IPSec keys used to create encrypted IP packets for transporting data
- Data payload encrypted (DES, AES, or other)
- Data integrity ensured with one-way hash functions (MD5 or SHA1)
- Fast bulk data transfer

Methods of Encryption and Integrity

Two parameters are decided during the negotiation:

- Encryption algorithm
- Hash algorithm

Table 2-1 displays the number of methods for encryption and integrity supported by security gateways:

Table 2-1 Methods of Encryption/integrity for IKE

Parameter	IKE Phase I (IKE SA)	IKE Phase II (IPSec SA)
Encryption	AES -256(default) 3DES DES CAST	3DEA AES -128 (default) AES - 256 DES CAST DES - 40CP CAST -40 NULL
Integrity	MD5 SHA1 (default)	MD5 (default) SHA1

NULL means perform an integrity check only; *packets are not encrypted*.

Diffie Hellman Groups

The Diffie-Hellman key computation (also known as exponential key agreement) is based on the Diffie Hellman (DH) mathematical groups. Table 2-2 shows the DH groups supported by a security gateway during the two phases of IKE:

Table 2-2 DH groups

Parameter	IKE Phase I (IKE SA)	IKE Phase II (IPSec SA)
Diffie Hellman Groups	Group2 (1024 bits) (default) Group1 (768 bits) Group5 (1536 bits) Group14 (2048 bits)	Group2 (1024 bits) (default) Group1 (768 bits) Group5 (1536 bits) Group14 (2048 bits)

A group with more bits ensures a key that is harder to break, but carries a heavy cost in terms of performance, since the computation requires more CPU cycles.

Phase I modes

Between security gateways, there are two modes for IKE phase I:

- Main Mode
- Aggressive Mode

If aggressive mode is *not* selected, the gateway defaults to main mode, performing the IKE negotiation using six packets; aggressive mode performs the IKE negotiation with three packets.

Main mode is preferred because:

- Main mode is partially encrypted, from the point at which the shared DH key is known to both peers.
- Main mode is less susceptible to *Denial of Service* (DoS) attacks. In main mode, the DH computation is performed *after* authentication. In aggressive mode, the DH computation is performed parallel to authentication. A peer that is not yet authenticated can force processor intensive Diffie-Hellman computations on the other peer.

Note - Aggressive mode is provided for backwards compatibility with pre-*NG* remote access clients. Also use aggressive mode when a Check Point Security Gateway needs to negotiate with third party VPN solutions that do not support main mode.

When dealing with remote access, IKE has additional modes:

- *Hybrid mode*. Hybrid mode provides an alternative to IKE phase I, where the gateway is allowed to authenticate using certificates and the client via some other means, such as SecurID. For more information on Hybrid mode, see: "Introduction to Remote Access VPN" on page 269.
- *Office mode*. Office mode is an extension to the IKE protocol. Office Mode is used to resolve routing issues between remote access clients and the VPN domain. During the IKE negotiation, a special mode called *config mode* is inserted between phases I and II. During config mode, the remote access client requests an IP address from the gateway. After the gateway assigns the IP address, the client creates a virtual adapter in the Operating System. The virtual adapter uses the assigned IP address. For further information, see: "Office Mode" on page 299.

Renegotiating IKE & IPSec Lifetimes

IKE phase I is more processor intensive than IKE phase II, since the Diffie-Hellman keys have to be produced and the peers authenticated each time. For this reason, IKE phase I is performed less frequently. However, the IKE SA is only valid for a certain period, after which the IKE SA must be renegotiated. The IPSec SA is valid for an even shorter period, meaning many IKE phase II's take place.

The period between each renegotiation is known as the *lifetime*. Generally, the shorter the lifetime, the more secure the IPSec tunnel (at the cost of more processor intensive IKE negotiations). With longer lifetimes, future VPN connections can be set up more quickly. By default, IKE phase I occurs once a day; IKE phase II occurs every hour but the time-out for each phase is configurable.

The IPSec lifetime can also be configured according to Kilo Bytes by using **DBedit** to edit the `objects_5_0.c` file. The relevant properties are under the community set:

- **ike_p2_use_rekey_kbytes**. Change from **false** (default) to **true**.
- **ike_p2_rekey_kbytes**. Modify to include the required rekeying value (default 50000).

Perfect Forward Secrecy

The keys created by peers during IKE phase II and used for IPsec are based on a sequence of random binary digits exchanged between peers, and on the DH key computed during IKE phase I.

The DH key is computed once, then used a number of times during IKE phase II. Since the keys used during IKE phase II are based on the DH key computed during IKE phase I, there exists a mathematical relationship between them. For this reason, the use of a single DH key may weaken the strength of subsequent keys. If one key is compromised, subsequent keys can be compromised with less effort.

In cryptography, *Perfect Forward Secrecy* (PFS) refers to the condition in which the compromise of a current session key or long-term private key does *not* cause the compromise of earlier or subsequent keys. Security gateways meet this requirement with a PFS mode. When PFS is enabled, a fresh DH key is generated during IKE phase II, and renewed for each key exchange.

However, because a new DH key is generated during each IKE phase I, no dependency exists between these keys and those produced in subsequent IKE Phase I negotiations. Enable PFS in IKE phase II only in situations where extreme security is required.

The DH group used during PFS mode is configurable between groups 1, 2, 5 and 14, with group 2 (1042 bits) being the default.

Note - PFS mode is supported only between gateways, not between gateways and remote access clients.

IP Compression

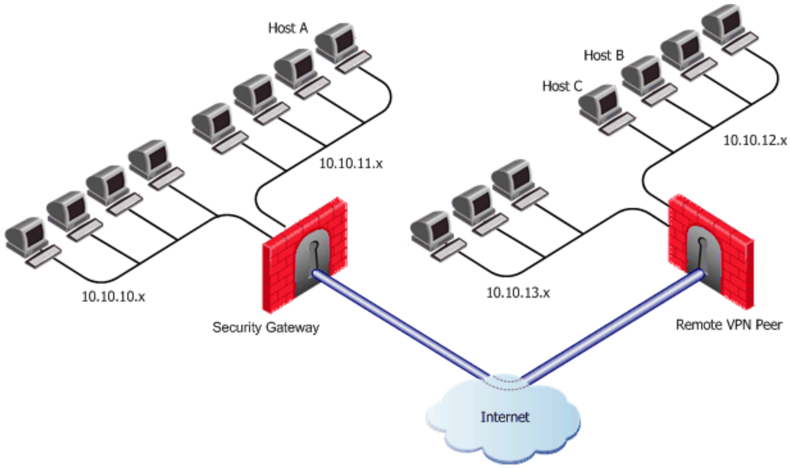
IP compression is a process that reduces the size of the data portion of the TCP/IP packet. Such a reduction can cause significant improvement in performance. IPsec supports the *Flate/Deflate* IP compression algorithm. Deflate is a smart algorithm that adapts the way it compresses data to the actual data itself. Whether to use IP compression is decided during IKE phase II. IP compression is not enabled by default.

IP compression is important for SecuRemote/SecureClient users with slow links. For Example, dialup modems do compression as a way of speeding up the link. Security gateway encryption makes TCP/IP packets appear “mixed up”. This kind of data cannot be compressed and bandwidth is lost as a result. If IP compression is enabled, packets are compressed *before* encryption. This has the effect of recovering the lost bandwidth.

Subnets and Security Associations

By default, a VPN tunnel is never created just for the hosts machines involved in the communication, but for the complete subnets the hosts reside on.

Figure 2-3 VPN per subnet



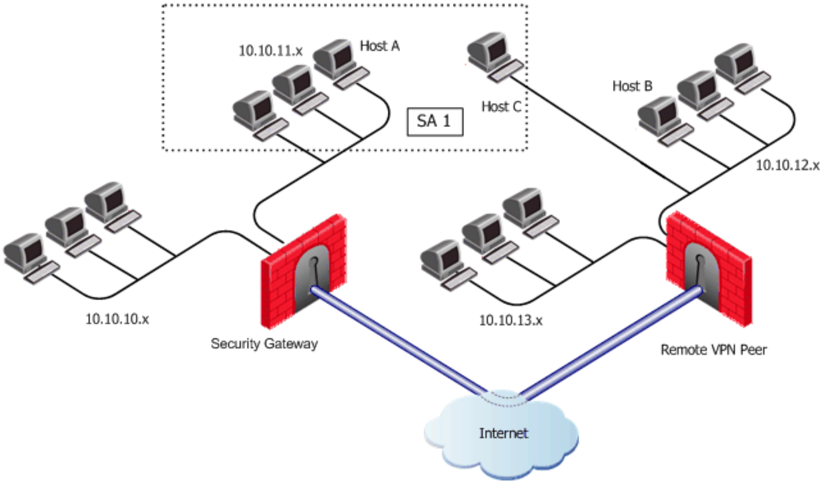
In Figure 2-3, a gateway protects a network consisting of two subnets (10.10.10.x, and 10.10.11.x, with netmask 255.255.255.0 for both). A second gateway, the remote peer, protects subnets 10.10.12.x and 10.10.13.x, with netmask 255.255.255.0.

Because a VPN tunnel is created by default for complete subnets, four SA's exist between the gateway and the peer gateway. When Host A communicates with Host B, an SA is created between Host A's subnet and Host B's subnet.

Unique SA Per Pair of Peers

By disabling the **Support Key exchange for subnets** option on each gateway, it is possible to create a *unique* Security Association per pair of peers.

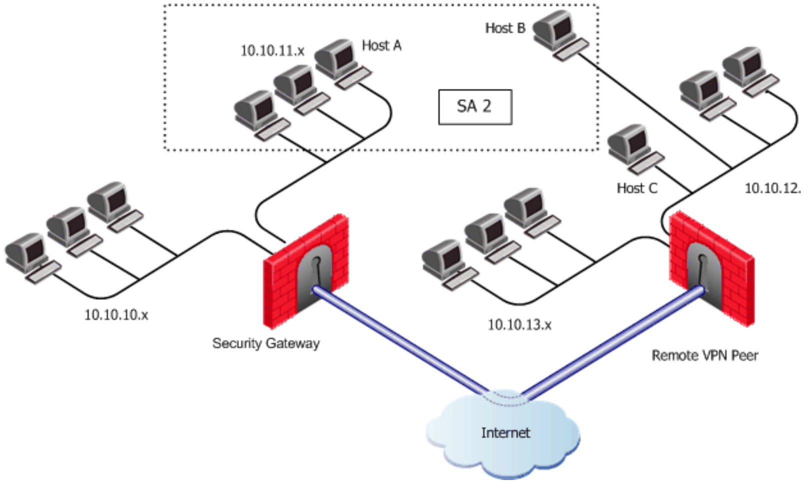
Figure 2-4 SA per IP address



In Figure 2-4, if the gateway is configured to **Support key exchange for subnets** and the option remains unsupported on the remote peer, when host A communicates with host C, a Security Association (SA 1) will be negotiated between host A's subnet and host C's IP address. The same SA is then used between any host on the 10.10.11.x subnet and Host C.

When host A communicates with host B, a separate Security Association (SA 2) is negotiated between host A's subnet and host B. As before, the same SA is then used between any host in 10.10.11.x subnet and Host B.

Figure 2-5 SA per host



In other words, when **Support Key exchange for subnets** is not enabled on communicating gateways, then a security association is negotiated between individual IP addresses; in effect, a unique SA per host.

IKE DOS Protection

In This Section

Understanding DoS Attacks

Denial of Service (DoS) attacks are intended to reduce performance, block legitimate users from using a service, or even bring down a service. They are not direct security threats in the sense that no confidential data is exposed, and no user gains unauthorized privileges. However, they consume computer resources such as memory or CPU.

Generally, there are two kinds of DoS attack. One kind consists of sending malformed (garbage) packets in the hope of exploiting a bug and crashing the service. In the other kind of DoS attack, an attacker attempts to exploit a vulnerability of the service or protocol by sending well-formed packets. IKE DoS attack protection deals with the second kind of attack.

IKE DoS Attacks

The IKE protocol requires that the receiving gateway allocates memory for the first IKE Phase 1 request packet that it receives. The gateway replies, and receives another packet, which it then processes using the information gathered from the first packet.

An attacker can send many IKE first packets, while forging a different source IP address for each. The receiving gateway is obliged to reply to each, and assign memory for each. This can consume all CPU resources, thereby preventing connections from legitimate users.

The attacker sending IKE packets can pretend to be a machine that is allowed to initiate IKE negotiations, such as a Check Point Security Gateway. This is known as an identified source. The attacker can also pretend to have an IP address that the receiving gateway does not know about, such as a SecuRemote/SecureClient, or a Check Point Security Gateway with a dynamic IP address. This is known as an unidentified source.

Defense Against IKE DoS Attacks

When the number of simultaneously IKE negotiations handled exceeds the accespted threshold, it concludes that it is either under load or experiencing a Denial of Service attack. In such a case, the gateway can filter out peers that are the probable source of a potential Denial of Service attack. There are two kinds of protection:

Stateless Protection Against IKE DoS Attacks

A security gateway prevents IKE DoS Attacks by delaying allocation of gateway resources until the peer proves itself to be legitimate. The following process is called stateless protection:

1. If the gateway concludes that it is either under load or experiencing a Denial of Service attack, and it receives an IKE request, it replies to the alleged source with a packet that contains a number that only the gateway can generate. The gateway then “forgets” about the IKE request. In other words, it does not need to store the IKE request in its memory (which is why the protection is called “Stateless”).
2. The machine that receives the packet is required to reinitiate the IKE request by sending an IKE request that includes this number.
3. If the gateway receives an IKE request that contains this number, the gateway will recognize the number as being one that only it can generate, and will only then continue with the IKE negotiation, despite being under load.

If the Check Point Security Gateway receives IKE requests from many IP addresses, each address is sent a different unique number, and each address is required to reinitiate the IKE negotiation with a packet that includes that number. If the peer does not reside at these IP addresses, this unique number will never reach the peer. This will thwart an attacker who is pretending to send IKE requests from many IP addresses.

IKE DoS attack protection is not available for third party gateways. Under heavy load, third party gateways and clients (such as Microsoft IPSec/L2TP clients) may be unable to connect.

Using Puzzles to Protect Against IKE DoS Attacks

Stateless protection is appropriate where the IKE packet appears to come from an identified source, that is, a machine that is allowed to initiate IKE negotiations, such as a Check Point Security Gateway.

An unidentified source is an IP address that the receiving gateway does not recognize, such as a SecuRemote/SecureClient, or a Check Point Security Gateway with a dynamic IP address. An attacker may well have control of many unidentified IP addresses, and may be able to reply to stateless packets from all these addresses. Therefore, if an attack comes from an unidentified source, another approach is required.

The gateway can require that the source of the IKE request solves a computationally intensive puzzle. Most computers can solve only a very few of these puzzles per second, so that an attacker would only be able to send very few IKE packets per second. This renders a DoS attack ineffective.

IKE DoS attack protection is not available for Third party gateways. Under heavy load, they may be unable to connect.

SmartDashboard IKE Dos Attack Protection Settings

To protect against IKE Dos attacks, edit the SmartDashboard **IKE Denial of Service Protection** settings, in the **VPN >Advanced** page of the **Global Properties**.

- **Support IKE DoS protection from identified source** — The default setting for identified sources is **Stateless**. If the gateway is under load, this setting requires the peer to respond to an IKE notification in a way that proves that the IP address of the peer is not spoofed. If the peer cannot prove this, the gateway does not begin the IKE negotiation.

  If the source is identified, protecting using **Puzzles** is over cautious, and may affect performance. A third possible setting is **None**, which means no DoS protection.

- **Support IKE DoS protection from unidentified source** — The default setting for unidentified sources is **Puzzles**. If the gateway is under load, this setting requires the peer to solve a mathematical puzzle. Solving this puzzle consumes peer CPU resources in a way that makes it difficult to initiate multiple IKE negotiations simultaneously.

  For unidentified sources, **Stateless** protection may not be sufficient because an attacker may well control all the IP addresses from which the IKE requests appear to be sent. A third possible setting is **None**, which means no DoS protection.

Advanced IKE Dos Attack Protection Settings

Advanced IKE DoS attack protection can be configured on the Security Management server using the **Dbedit** command line or using the graphical Database Tool. Configure the protection by means of the following Global Properties.

ike_dos_threshold

Values: 0-100. Default: 70. Determines the percentage of maximum concurrent ongoing negotiations, above which the gateway will request DoS protection. If the threshold is set to 0, the gateway will always request DoS protection.

ike_dos_puzzle_level_identified_initiator

Values: 0-32. Default: 19. Determines the level of the puzzles sent to known peer gateways. This attribute also determines the maximum puzzle level a gateway is willing to solve.

ike_dos_puzzle_level_unidentified_initiator

Values: 0-32. Default: 19. Determines the level of the puzzles sent to unknown peers (such as SecuRemote/SecureClients and DAIP gateways). This attribute also determines the maximum puzzle level that DAIP gateways and SecuRemote/SecureClients are willing to solve.

ike_dos_max_puzzle_time_gw

Values: 0-30000. Default: 500. Determines the maximum time in milliseconds a gateway is willing to spend solving a DoS protection puzzle.

ike_dos_max_puzzle_time_daip

Values: 0-30000. Default: 500. Determines the maximum time in milliseconds a DAIP gateway is willing to spend solving a DoS protection puzzle.

ike_dos_max_puzzle_time_sr

Values: 0-30000. Default: 5000. Determines the maximum time in milliseconds a SecuRemote is willing to spend solving a DoS protection puzzle.

ike_dos_supported_protection_sr

Values: None, Stateless, Puzzles. Default: Puzzles. When downloaded to SecuRemote/SecureClient, it controls the level of protection the client is willing to support.

gateways use the ike_dos_protection_unidentified_initiator property (equivalent to the SmartDashboard Global Property: **Support IKE DoS Protection from unidentified Source**) to decide what protection to require from remote clients, but SecuRemote/SecureClient clients use the ike_dos_protection. This same client property is called ike_dos_supported_protection_sr on the gateway.

Client Properties

Some gateway properties change name when they are downloaded to SecuRemote/SecureClient. The modified name appears in the Userc.C file, as follows:

Table 2-3 Property Names

Property Name on Gateway	Userc.C Property Name on SecuRemote/SecureClient
ike_dos_protection_unidentified_initiator (Equivalent to the SmartDashboard Global Property: **Support IKE DoS Protection from unidentified Source**)	ike_dos_protection or ike_support_dos_protection
ike_dos_supported_protection_sr	ike_dos_protection
ike_dos_puzzle_level_unidentified_initiator	ike_dos_acceptable_puzzle_level
ike_dos_max_puzzle_time_sr	ike_dos_max_puzzle_time

Configuring Advanced IKE Properties

IKE is configured in two places:

- On the VPN community network object (for IKE properties).
- On the gateway network object (for subnet key exchange).

On the VPN Community Network Object

1. **VPN Properties** page, select:
   - Encryption methods for IKE phase I and II
   - Integrity methods for IKE phase I and II
2. On the **Advanced Settings > Advanced VPN Properties** page, select:
   - Which Diffie-Hellman group to use.
   - When to renegotiate the IKE Security Associations.
   - Whether to use **aggressive mode** (Main mode is the default).
   - Whether to use **Perfect Forward Secrecy**, and with which Diffie-Hellman group.
   - When to renegotiate the IPSec security associations.
   - Whether to use **Support IP compression**.

On the Gateway Network Object

On the **VPN Advanced** page, *deselect* **Support Key exchange for subnets** if you want the SA to be calculated per host. The default is to support key exchange for subnets.

Chapter 4

Introduction to Site to Site VPN

In This Chapter:

The Need for Virtual Private Networks

Communicating parties need a connectivity platform that is not only fast, scalable, and resilient but also provides:

- Confidentiality
- Integrity
- Authentication

Confidentiality

Only the communicating parties must be able to read the private information exchanged between them.

Authentication

The communicating parties must be sure they are connecting with the intended party.

Integrity

The sensitive data passed between the communicating parties is unchanged, and this can be proved with an integrity check.

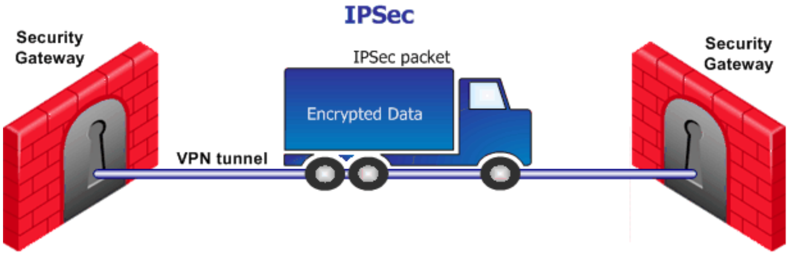
The Check Point Solution for VPN

A *Virtual Private Network* (VPN) is a secure connectivity platform that both *connects* networks and *protects* the data passing between them. For example, an organization may have geographically spaced networks connected via the Internet; the company has connectivity but no privacy. Check Point Security Gateways provide privacy by encrypting those connections that need to be secure. Another company may connect all parts of its geographically spaced network through the use of dedicated leased lines; this company has achieved connectivity and privacy but at great expense. The Check Point Product Suite offers a cheaper connectivity solution by connecting the different parts of the network via the public Internet.

A Virtual Private Network is a network that employs encrypted tunnels to exchange securely protected data. Check Point Security gateways create encrypted tunnels by using the *Internet Key Exchange* (**IKE**) and *IP Security* (**IPSec**) protocols. IKE creates the VPN tunnel, and this tunnel is used to transfer IPSec encoded data.

Think of IKE as the process that builds a tunnel, and IPSec packets as trucks that carry the encrypted data along the tunnel.

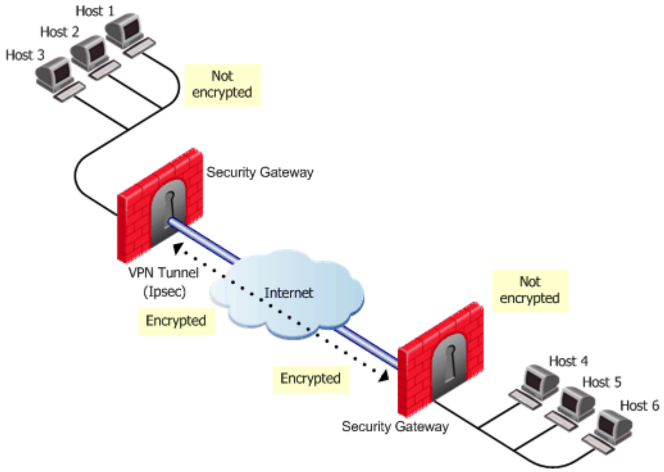
Figure 4-1 Simplified VPN tunnel



How it Works

In Figure 4-2, host 1 and host 6 need to communicate. The connection passes in the clear between host 1 and the local gateway. From the source and destination addresses of the packet, the gateway determines that this should be an encrypted connection. If this is the first time the connection is made, the local gateway initiates an IKE negotiation with the peer gateway in front of host 6. During the negotiation, both gateways authenticate each other, and agree on encryption methods and keys. After a successful IKE negotiation, a VPN tunnel is created. From now on, every packet that passes between the gateways is encrypted according to the IPSec protocol. IKE supplies authenticity (gateways are sure they are communicating with each other) and creates the foundation for IPSec. Once the tunnel is created, IPSec provides privacy (through encryption) and integrity (via one-way hash functions).

Figure 4-2 Confidentiality, integrity, and authentication via IPSec.



After a VPN tunnel has been established (Figure 4-2), packets are dealt with in the following way:

- A packet leaves the source host and reaches the gateway.
- The gateway encrypts the packet.
- The packet goes down the VPN tunnel to the second gateway. In actual fact, the packets are standard IP packets passing through the Internet. However, because the packets are encrypted, they can be considered as passing through a private "virtual" tunnel.
- The second gateway decrypts the packet.
- The packet is delivered in the clear to the destination host. From the hosts perspective, they are connecting directly.

For more information regarding the IKE negotiation, see: "IPSEC & IKE".

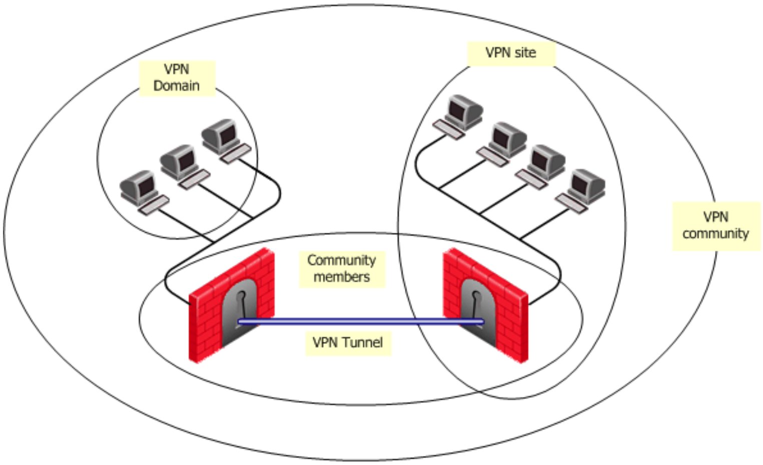
VPN Communities

Creating VPN tunnels between gateways is made easier through the configuration of VPN communities. A VPN community is a collection of VPN enabled gateways capable of communicating via VPN tunnels.

To understand VPN Communities, a number of terms need to be defined:

- *VPN Community member*. Refers to the gateway that resides at one end of a VPN tunnel.
- *VPN domain*. Refers to the hosts behind the gateway. The VPN domain can be the whole network that lies behind the gateway or just a section of that network. For example a gateway might protect the corporate LAN and the DMZ. Only the corporate LAN needs to be defined as the VPN domain.
- *VPN Site*. Community member plus VPN domain. A typical VPN site would be the branch office of a bank.
- *VPN Community*. The collection of VPN tunnels/links and their attributes.
- *Domain Based VPN*. Routing VPN traffic based on the encryption domain behind each gateway in the community. In a star community, this allows satellite gateways to communicate with each other through center gateways.
- *Route Based VPN*. Traffic is routed within the VPN community based on the routing information, static or dynamic, configured on the Operating Systems of the gateways.

Figure 4-3 VPN Terminology



The methods used for encryption and ensuring data integrity determine the type of tunnel created between the gateways, which in turn is considered a characteristic of that particular VPN community.

Security Management server can manage multiple VPN communities, which means communities can be created and organized according to specific needs.

Note - Defining services in the clear in the community (available in gateway-to-gateway communities) is not supported if one of the internally managed members is of version earlier than NG FP3.

Remote Access Community

A Remote Access Community is a type of VPN community created specifically for users that usually work from remote locations, outside of the corporate LAN. This type of community ensures secure communication between users and the corporate LAN. For more information, see: "Introduction to Remote Access VPN" on page 269.

VPN Topologies

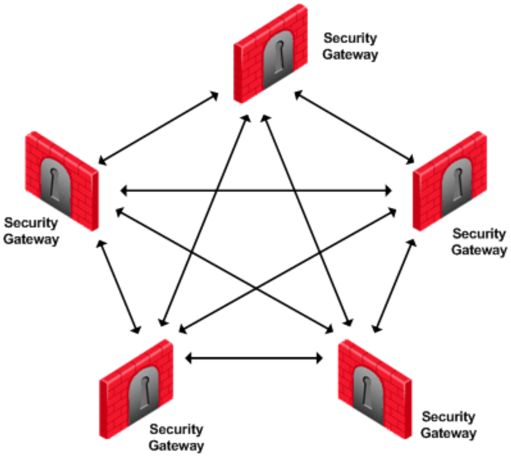
The most basic topology consists of two gateways capable of creating a VPN tunnel between them. Security Management server's support of more complex topologies enables VPN communities to be created according to the particular needs of an organization. Security Management server supports two main VPN topologies:

- Meshed
- Star

Meshed VPN Community

A Mesh is a VPN community in which a VPN site can create a VPN tunnel with any other VPN site in the community:

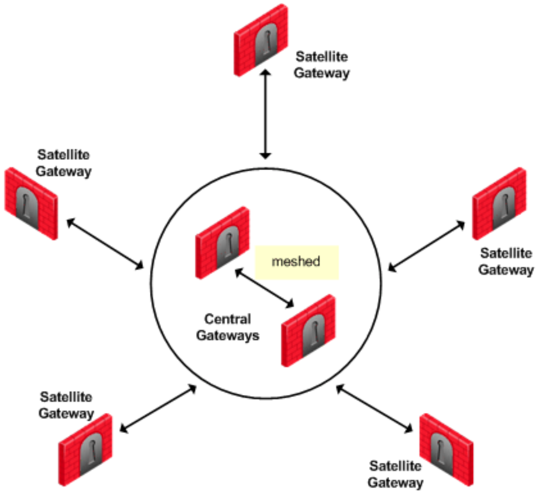
Figure 4-4 Basic Meshed communityp



Star VPN Community

A star is a VPN community consisting of central gateways (or “hubs”) and satellite gateways (or “spokes”). In this type of community, a satellite can create a tunnel only with other sites whose gateways are defined as central.

Figure 4-5 Star VPN community



A satellite gateway cannot create a VPN tunnel with a gateway that is also defined as a satellite gateway.

Central gateways can create VPN tunnels with other Central gateways only if the **Mesh center gateways** option has been selected on the **Central Gateways** page of the **Star Community Properties** window.

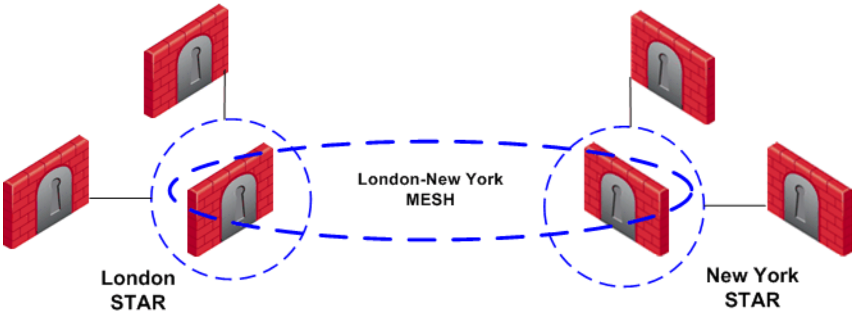
Choosing a Topology

Which topology to choose for a VPN community depends on the overall policy of the the organization. For example, a meshed community is usually appropriate for an Intranet in which only gateways which are part of the internally managed network are allowed to participate; gateways belonging to company partners are not.

A Star VPN community is usually appropriate when an organization needs to exchange information with networks belonging to external partners. These partners need to communicate with the organization but not with each other. The organization's gateway is defined as a "central" gateway; the partner gateways are defined as "satellites."

For more complex scenarios, consider a company with headquarters in two countries, London and New York. Each headquarters has a number of branch offices. The branch offices only need to communicate with the HQ in their country, not with each other; only the HQ's in New York and London need to communicate directly. To comply with this policy, define two star communities, London and New York. Configure the London and New York gateways as "central" gateways. Configure the gateways of New York and London branch offices as "satellites." This allows the branch offices to communicate with the HQ in their country. Now create a third VPN community, a VPN mesh consisting of the London and New York gateways.

Figure 4-6 Two stars and mesh

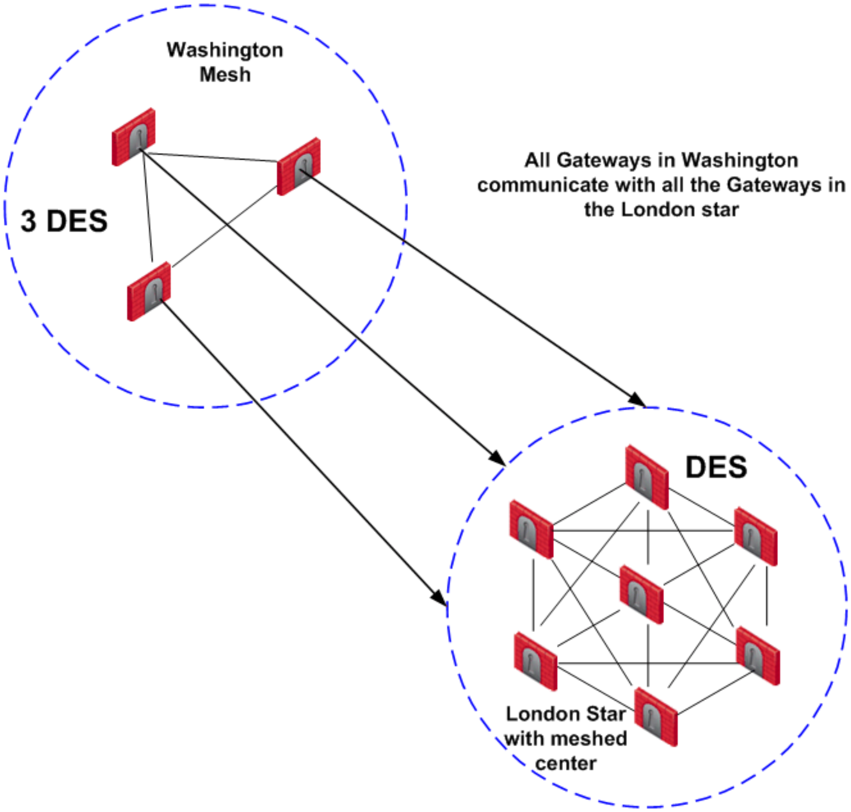


Topology and Encryption Issues

Issues involving topology and encryption can arise as a result of an organization's policy on security, for example the country in which a branch of the organization resides may have a national policy regarding encryption strength. For example, policy says the Washington gateways should communicate using 3DES for encryption. Policy also states the London gateways must communicate uses DES as the encryption algorithm.

In addition, the Washington and London gateways (as shown in Figure 4-7) need to communicate with each other using the weaker DES. Consider the solution in Figure 4-7:

Figure 4-7 Different means of encryption in separate Mesh communities

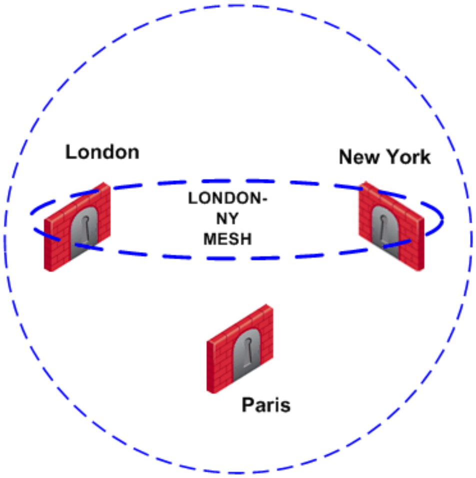

Washington Mesh
3 DES
All Gateways in Washington communicate with all the Gateways in the London star
DES
London Star with meshed center

In this solution, gateways in the Washington mesh are also defined as satellites in the London star. In the London star, the central gateways are *meshed*. Gateways in Washington build VPN tunnels with the London gateways using DES. Internally, the Washington gateways build VPN tunnels using 3DES.

Special Condition for VPN Gateways

Individually, gateways can appear in many VPN communities; however, two gateways that can create a VPN link between them in one community cannot appear in another VPN community in which they can *also* create a link. For example:

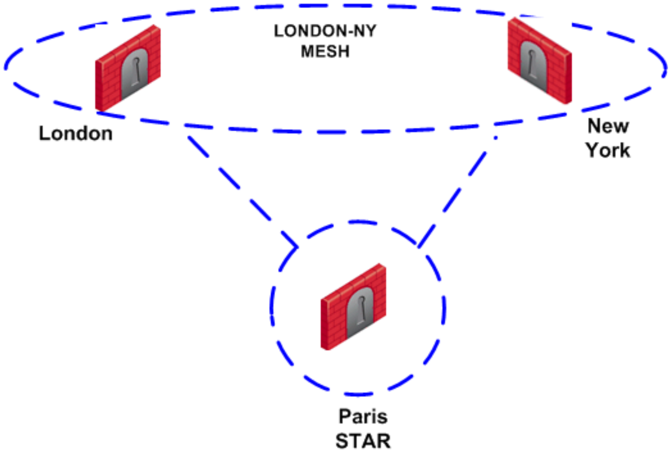
Figure 4-8 Special condition



The London and New York gateways belong to the London-NY Mesh VPN community. To create an additional VPN community which includes London, New York, and Paris is not allowed. The London and New York gateways cannot appear "together" in more than one VPN community.

Two gateways that can create a VPN link between them in one community can appear in another VPN community provided that they are *incapable* of creating a link between them in the second community. For example:

Figure 4-9 Three VPN communities



In Figure 4-9, The London and New York gateways appear in the London-NY mesh. These two gateways also appear as Satellite gateways in the Paris Star VPN community. In the Paris Star, satellite gateways (London and NY) can *only* communicate with the central Paris gateway. Since the London and New York satellite gateways *cannot* open a VPN link between them, this is a valid configuration.

Authentication Between Community Members

Before gateways can exchange encryption keys and build VPN tunnels, they first need to authenticate to each other. Gateways authenticate to each other by presenting one of two types of "credentials":

- **Certificates**. Each gateway presents a certificate which contains identifying information of the gateway itself, and the gateway's public key, both of which are signed by the trusted CA. For convenience, the Check Point product suite installs its own Internal CA that automatically issues certificates for all internally managed gateways, requiring no configuration by the user. In addition, the Check Point Product Suite supports other PKI solutions. For more information, see: "Public Key Infrastructure" on page 91.
- **Pre-shared secret**. A pre-shared is defined for a pair of gateways. Each gateway proves that it knows the agreed upon pre-shared secret. The pre-shared secret can be a mixture of letters and numbers, a password of some kind.

Considered more secure, certificates are the preferred means. In addition, since the Internal CA on the Security Management server automatically provides a certificate to each Check Point Security Gateway it manages, it is more convenient to use this type of authentication.

However, if a VPN tunnel needs to be created with an externally managed gateway (a gateway managed by a different Security Management server) the externally managed gateway:

- Might support certificates, but certificates issued by an external CA, in which case both gateways need to trust the other's CA. (For more information, see: "Configuring a VPN with External Gateways Using PKI" on page 81.)
- May not support certificates; in which case, VPN supports the use of a "pre-shared secret." For more information, see: "Configuring a VPN with External Gateways Using a Pre-Shared Secret" on page 85.

  A "secret" is defined per external gateway. If there are five internal gateways and two externally managed gateways, then there are two pre-shared secrets. The two pre-shared secrets are used by the five internally managed gateways. In other words, all the internally managed gateways use the same pre-shared secret when communicating with a particular externally managed gateway.

Dynamically Assigned IP Gateways

A Dynamically Assigned IP (DAIP) gateway is a gateway where the external interface's IP address is assigned dynamically by the ISP. Creating VPN tunnels with DAIP gateways are only supported by using certificate authentication. Peer gateways identify internally managed DAIP gateways using the DN of the certificate. Peer gateways identify externally managed DAIP gateways and 3rd party DAIP gateways using the *Matching Criteria* configuration

DAIP gateways may initiate a VPN tunnel with non-DAIP gateways. However, since a DAIP gateway's external IP address is always changing, peer gateways cannot know in advance which IP address to use to connect to the DAIP gateway. As a result, a peer gateway cannot initiate a VPN tunnel with a DAIP gateway unless DNS Resolving is configured on the DAIP gateway. For more information, see "Link Selection" on page 207.

If the IP on the DAIP gateway changes during a session, it will renegotiate IKE using the newly assigned IP address.

In a star community when VPN routing is configured, DAIP gateways cannot initiate connections from their external IP through the center gateway(s) to other DAIP gateways or through the center to the Internet. In this configuration, connections from the encryption domain of the DAIP are supported.

Routing Traffic within a VPN Community

VPN routing provides a way of controlling how VPN traffic is directed. There are two methods for VPN routing:

- Domain Based VPN
- Route Based VPN

Domain Based VPN

This method routes VPN traffic based on the encryption domain behind each gateway in the community. In a star community, this allows satellite gateways to communicate with each other through center gateways. Configuration for Domain Based VPN is performed directly through SmartDashboard. For more information, see "Domain Based VPN" on page 119.

Route Based VPN

Traffic is routed within the VPN community based on the routing information, static or dynamic, configured on the Operating Systems of the gateways. For more information, see "Route Based VPN" on page 129.

Note - If both Domain Based VPN and Route Based VPN are configured, then Domain Based VPN will take precedence.

Access Control and VPN Communities

Configuring gateways into a VPN community does not create a de facto access control policy between the gateways. The fact that two gateways belong to the same VPN community does not mean the gateways have access to each other.

The configuration of the gateways into a VPN community means that *if* these gateways are allowed to communicate via an access control policy, then that communication is encrypted. Access control is configured in the Security Policy Rule Base.

Using the VPN column of the Security Policy Rule Base, it is possible to create access control rules that apply *only* to members of a VPN community, for example:

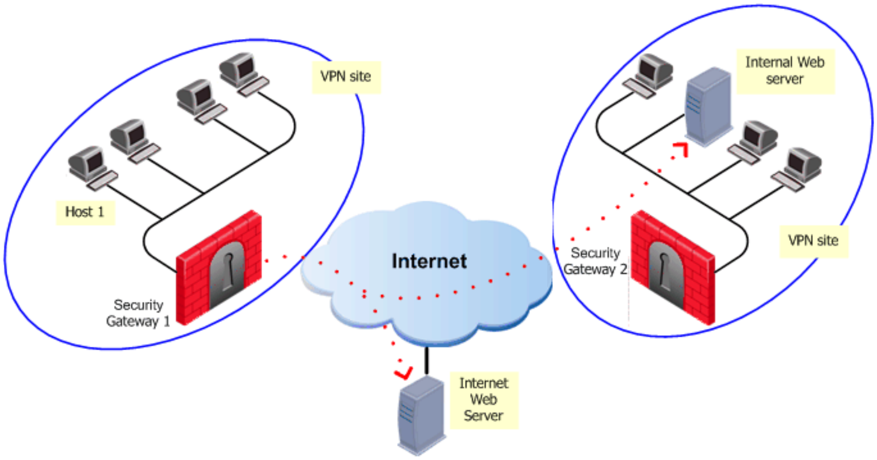
Table 4-1

Source	Destination	VPN	Service	Action
Any	Any	Community_A	HTTP	Accept

The connection is matched only if all the conditions of the rule are true, that is - it must be an HTTP connection between a source and destination IP address within VPN Community A. If any one of these conditions is not true, the rule is not matched. If all conditions of the rule are met, the rule is matched and the connection allowed.

It is also possible for a rule in the Security Policy Rule Base to be relevant for both VPN communities and host machines *not* in the community. For example:

Figure 4-10 Access control in VPN communities



The rule in the Security Policy Rule base allows an HTTP connection between any internal IP with any IP:

Table 4-2

Source	Destination	VP N	Service	Action
Any_internal_machine	Any	Any	HTTP	Accept

In Figure 4-10, an HTTP connection between host 1 and the Internal web server behind gateway 2 matches this rule. A connection between the host 1 and the web server on the Internet also matches this rule; however, the connection between host 1 and the internal web server is a connection between members of a VPN community and passes encrypted; the connection between host 1 and the Internet web server passes in the clear.

In both cases, the connection is simply matched to the Security Policy Rule; whether or not the connection is encrypted is dealt with on the VPN level. *VPN is another level of security separate from the access control level*.

Accepting all Encrypted Traffic

If you select **Accept all encrypted traffic** on the **General** page of the VPN community **Properties** window, a new rule is added to the Security Policy Rule Base. This rule is neither a regular rule or an implied rule, but an *automatic community rule*, and can be distinguished by its “beige” colored background.

Excluded Services

In the VPN **Communities Properties** window **Excluded Services** page, you can select services that are *not* to be encrypted, for example Firewall control connections. Services in the clear means “do not make a VPN tunnel for this connection”. For further information regarding control connections, see: “How to Authorize Firewall Control Connections in VPN Communities” on page 88. Note that *Excluded Services* is not supported when using *Route Based VPN*.

Special Considerations for Planning a VPN Topology

When planning a VPN topology it is important to ask a number of questions:

1. Who needs secure/private access?
2. From a VPN point of view, what will be the structure of the organization?
3. Internally managed gateways authenticate each other using certificates, but how will externally managed gateways authenticate?
    - Do these externally managed gateways support PKI?
    - Which CA should be trusted?

Configuring Site to Site VPNs

VPN communities can be configured in either traditional or simplified mode. In *Traditional mode*, one of the actions available in the Security Policy Rule Base is **Encrypt**. When encrypt is selected, all traffic between the gateways is encrypted. Check Point Security gateways are more easily configured through the use of VPN communities — otherwise known as working in *Simplified Mode*. For more information regarding traditional mode, see: "Traditional Mode VPNs" on page 251.

Migrating from Traditional Mode to Simplified Mode

To switch from Traditional mode to Simplified mode (For more information, see "Converting a Traditional Policy to a Community Based Policy" on page 717):

1. On the **Global Properties > VPN** page, select either **Simplified mode to all new Security Policies**, or **Traditional or Simplified per new Security Policy. File > Save**. If you do not save, you are prompted to do so.
2. **File > New...** The **New Policy Package** window opens.
3. Create a name for the new security policy package and select **Security and Address Translation.**
4. For the **VPN configuration method**, select **Simplified mode** (if you selected **Traditional or Simplified per new Security policy** in **Global Properties**). Click **OK**.

In the Security Policy Rule base, a new column marked **VPN** appears and the **Encrypt** option is *no longer available* in the **Action** column. You are now working in Simplified Mode.

Configuring a Meshed Community Between Internally Managed Gateways

Internally managed VPN communities have one of two possible topologies; meshed or star. To configure an internally managed VPN meshed community, create the network objects (gateways) first and then add them to the community:

1. In the **Network Objects** tree, right click **Network Objects** > **New** > **Check Point** > **Gateway...**Select **Simple mode (wizard)** or **Classic mode**. The **Check Point Gateway properties** window opens.
   a. On the **General Properties** page, after naming the object and supplying an IP address, select **VPN** and establish SIC communication.
   b. On the **Topology** page, click **Add** to add interfaces. Once an interface appears in the table, clicking **Edit...** opens the **Interface Properties** window.
   c. In the **Interface Properties** window, define the general properties of the interface and the topology of the network behind it.
   d. Still on the **Topology** page, **VPN Domain** section, define the VPN domain as either all the machines behind the gateway based on the topology information or manually defined:
      i. As an address range.
      ii. As a network.
      iii. As a group, which can be a combination of address ranges, networks, and even other groups.

         (There are instances where the VPN domain is a group which contains only the gateway itself, for example where the gateway is acting as a backup to a primary gateway in a MEPed environment.)

The network gateway objects are now configured, and need to be added to a VPN community.

Note - There is nothing to configure on the **VPN** page, regarding certificates, since internally managed gateways automatically receive a certificate from the internal CA.

2. On the **Network objects** tree, select the **VPN Communities** tab. 🔒
   a. Right-click **Site to Site**.
   b. From the short-cut menu, select **New Site To Site... > Meshed**. The **Meshed Communities Properties** window opens.

c. On the **General** page, select **Accept all encrypted traffic** if you need all traffic between the gateways to be encrypted. If not, then create appropriate rules in the Security Policy Rule Base that allows encrypted traffic between community members.

d. On the **Participating Gateways** page, add the gateways created in step 1.

A VPN tunnel is now configured. For more information on other options, such as **VPN Properties**, **Advanced Properties**, and **Shared Secret**, see: "IPSEC & IKE" on page 43.

3. If you did not select **Accept all encrypted traffic** in the community, build an access control policy, for example:

Table 4-3

Source	Destination	VPN	Service	Action
Any	Any	Meshed community	Any	Accept

Where "Meshed community" is the VPN community you have just defined.

Configuring a Star VPN Community

A star VPN community is configured in much the same way as a meshed community, the difference being the options presented on the **Star Community Properties** window:

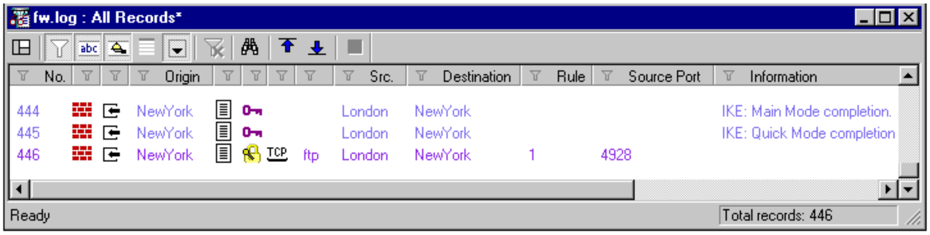
- On the **General** page, **Enable VPN routing for satellites** section, select **To center only**.
- On the **Central Gateways** page, **Add...** the central gateways.
- On the **Central Gateways** page, select **Mesh central gateways** if you want the central gateways to communicate.
- On the **Satellite Gateways** page, click **Add...** to add the satellite gateways.

Confirming a VPN Tunnel Successfully Opens

To confirm a VPN tunnel has successfully opened:

1. Edit a rule in the Security Policy Rule base that encrypts a specific service between Member gateways of a VPN community, for example FTP.
2. Select **log** as the tracking option.
3. Open an appropriate connection, in this example FTP session from a host behind the first gateway to an FTP server behind the second.
4. Open SmartView Tracker and examine the logs. The connection appears as encrypted, as in Figure 4-11.

Figure 4-11 Sample log



fw.log : All Records*

No.			Origin					Src.	Destination	Rule	Source Port	Information
444			NewYork					London	NewYork			IKE: Main Mode completion.
445			NewYork					London	NewYork			IKE: Quick Mode completion
446			NewYork			TCP	ftp	London	NewYork	1	4928	

Ready

Total records: 446

Configuring a VPN with External Gateways Using PKI

Configuring a VPN with external gateways (those managed by a different Security Management server) is more complicated than configuring a VPN with internal gateways (managed by the same Security Management server). This is because:

- Configuration is done separately in two distinct systems.
- All details must be agreed and coordinated between the administrators. Details such as the IP address or the VPN domain topology cannot be detected automatically but have to be supplied manually by the administrator of the peer VPN gateways.
- The gateways are likely to be using different Certificate Authorities (CAs). Even if the peer VPN gateways use the Internal CA (ICA), it is still a different CA.

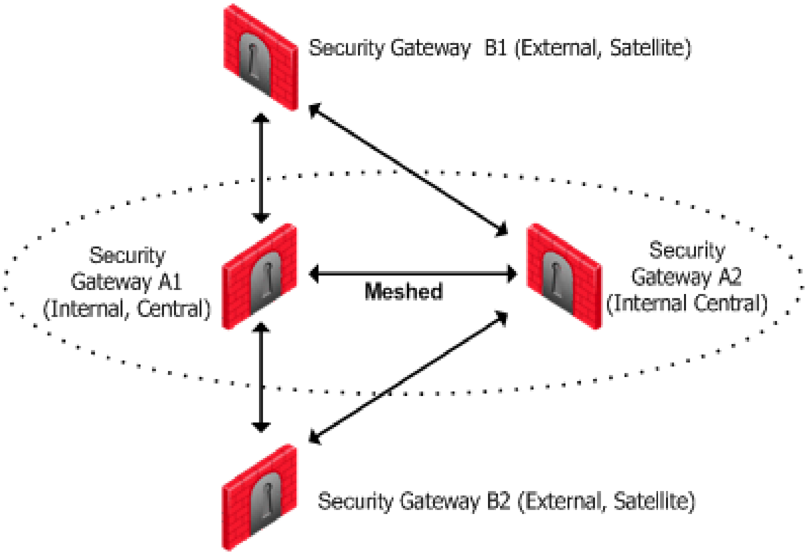
There are various scenarios when dealing with externally managed gateways. The following description tries to address typical cases and assumes that the peers work with certificates. If this is not the case refer to "Configuring a VPN with External Gateways Using a Pre-Shared Secret" on page 85.

Note - Configuring a VPN using PKI and certificates is considered more secure than using pre-shared secrets.

Although an administrator may choose which community type to use, the Star Community is more natural for a VPN with externally managed gateways. The Internal gateways will be defined as the central gateways while the external ones will be defined as the satellites. The decision whether to mesh the central, internal gateways or not depends on the requirements of the organization. The diagram below shows this typical topology.

Note that this is the Topology from the point of view of the administrator of gateways A1 and A2. The Administrator of gateways B1 and B2 may well also define a Star Topology, but with B1 and B2 as his central gateways, and A1 and A2 as satellites.

Figure 4-12 External Gateways as Satellites in a Star VPN Community


Security Gateway B1 (External, Satellite)
Security Gateway A1 (Internal, Central)
Meshed
Security Gateway A2 (Internal Central)
Security Gateway B2 (External, Satellite)

The configuration instructions require an understanding of how to build a VPN. The details can be found in: "Introduction to Site to Site VPN" on page 61.

You also need to understand how to configure PKI. See "Public Key Infrastructure" on page 91.

To configure VPN using certificates, with the external gateways as satellites in a star VPN Community:

1. Obtain the certificate of the CA that issued the certificate for the peer VPN gateways, from the peer administrator. If the peer gateway is using the ICA, you can obtain the CA certificate using a web browser from:

   ```
   http://<IP address of peer gateway or Management Server>:18264
   ```

2. In SmartDashboard, define the CA object for the CA that issued the certificate for the peer. See "Enrolling with a Certificate Authority" on page 107.
3. Define the CA that will issue certificates for your side if the Certificate issued by ICA is not appropriate for the required VPN tunnel.

   You may have to export the CA certificate and supply it to the peer administrator.

4. Define the Network Object(s) of the gateway(s) that are internally managed. In particular, be sure to do the following:
   - In the **General Properties** page of the gateway object, select **VPN**.

- In the **Topology** page, define the **Topology**, and the **VPN Domain**. If the VPN Domain does not contain all the IP addresses behind the gateway, define the VPN domain manually by defining a group or network of machines and setting them as the VPN Domain.

5. If the ICA certificate is not appropriate for this VPN tunnel, then in the **VPN** page, generate a certificate from the relevant CA (see "Enrolling with a Certificate Authority" on page 107).

6. Define the Network Object(s) of the externally managed gateway(s).

   - If it is not a Check Point gateway, define an Interoperable Device object from: **Manage > Network Objects... > New... > Interoperable Device...**
   - If it is a Check Point gateway, In the **Network Objects** tree, right click and select **New > Check Point > Externally Managed Gateway...**.

7. Set the various attributes of the peer gateway. In particular, be sure to do the following:

   - In the **General Properties** page of the gateway object, select **VPN** (for an Externally Managed Check Point gateway object only).
   - in the **Topology** page, define the **Topology** and the **VPN Domain** using the VPN Domain information obtained from the peer administrator. If the VPN Domain does not contain all the IP addresses behind the gateway, define the VPN domain manually by defining a group or network of machines and setting them as the VPN Domain.
   - In the **VPN** page, define the **Matching Criteria**. specify that the peer must present a certificate signed by its own CA. If feasible, enforce details that appear in the certificate as well.

8. Define the Community. The following details assume that a Star Community was chosen, but a Meshed Community is an option as well. If working with a Meshed community ignore the difference between the Central gateways and the Satellite gateways.

   - Agree with the peer administrator about the various IKE properties and set them in the **VPN Properties** page and the **Advanced Properties** page of the community object.
   - Define the Central gateways. These will usually be the internally managed ones. If there is no another Community defined for them, decide whether or not to mesh the central gateways. If they are already in a Community, do not mesh the central gateways.
   - Define the Satellite gateways. These will usually be the external ones.

9. Define the relevant access rules in the Security Policy. Add the Community in the **VPN** column, the services in the **Service** column, the desired **Action**, and the appropriate **Track** option.
10. Install the Security Policy.

Configuring a VPN with External Gateways Using a Pre-Shared Secret

Configuring VPN with external gateways (those managed by a different Security Management server) is more involved than configuring VPN with internal gateways (managed by the same Security Management server) because:

- Configuration is done separately in two distinct systems.
- All details must be agreed and coordinated between the administrators. Details such as the IP address or the VPN domain topology cannot be detected automatically but have to be supplied manually by the administrator of the peer VPN gateways.

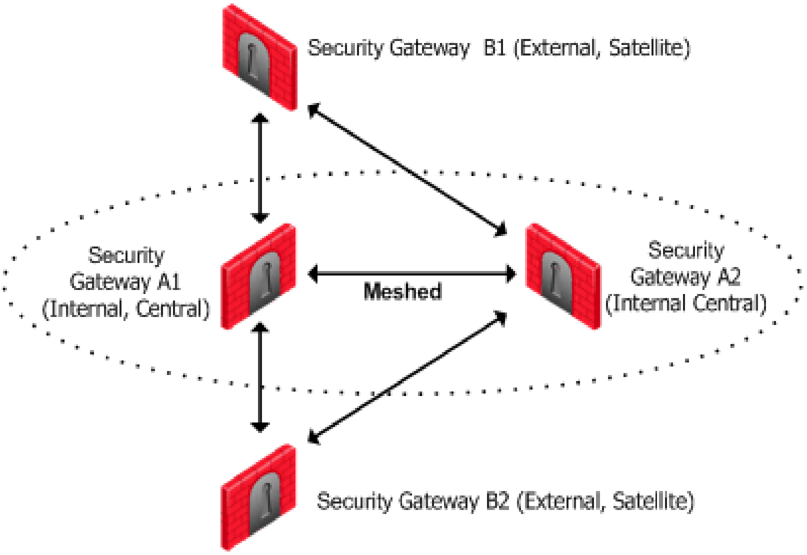
There are various scenarios when dealing with externally managed gateways. The following description tries to address typical cases but assumes that the peers work with pre-shared secrets. If this is not the case refer to "Configuring a VPN with External Gateways Using PKI" on page 81.

Note - Configuring a VPN using PKI and certificates is considered more secure than using pre-shared secrets.

Although an administrator may choose which community type to use, the Star Community is more natural for a VPN with externally managed gateways. The Internal gateways will be defined as the central gateways while the external ones will be defined as the satellites. The decision whether to mesh the central, internal gateways or not depends on the requirements of the organization. The diagram below shows this typical topology.

Note that this is the Topology from the point of view of the administrator of gateways A1 and A2. The administrator of gateways B1 and B2 may well also define a Star Topology, but with B1 and B2 as his central gateways, and A1 and A2 as satellites.

Figure 4-13 External Gateways as Satellites in a Star VPN Community


Security Gateway B1 (External, Satellite)
Security Gateway A1 (Internal, Central)
Meshed
Security Gateway A2 (Internal Central)
Security Gateway B2 (External, Satellite)

The configuration instructions require an understanding of how to build a VPN. The details can be found in: "Introduction to Site to Site VPN" on page 61.

To configure a VPN using pre-shared secrets, with the external gateways as satellites in a star VPN Community, proceed as follows:

1. Define the Network Object(s) of the gateway(s) that are internally managed. In particular, be sure to do the following:
   - In the **General Properties** page of the gateway object, select **VPN**.
   - In the **Topology** page, define the **Topology**, and the **VPN Domain**. If the VPN Domain does not contain all the IP addresses behind the gateway, define the VPN domain manually by defining a group or network of machines and setting them as the VPN Domain.
2. Define the Network Object(s) of the externally managed gateway(s).
   - If it is not a Check Point gateway, define an Interoperable Device object from: **Manage > Network Objects... > New... > Interoperable Device...**
   - If it is a Check Point gateway, In the **Network Objects** tree, right click and select **New > Check Point > Externally Managed Gateway...**.
3. Set the various attributes of the peer gateway. In particular, be sure to do the following:
   - In the **General Properties** page of the gateway object, select **VPN** (for an Externally Managed Check Point gateway object only).

- in the **Topology** page, define the **Topology** and the **VPN Domain** using the VPN Domain information obtained from the peer administrator. If the VPN Domain does not contain all the IP addresses behind the gateway, define the VPN domain manually by defining a group or network of machines and setting them as the VPN Domain.

4. Define the Community. The following details assume that a Star Community was chosen, but a Meshed Community is an option as well. If working with a Mesh community ignore the difference between the Central gateways and the Satellite gateways.
    - Agree with the peer administrator about the various IKE properties and set them in the **VPN Properties** page and the **Advanced Properties** page of the community object.
    - Define the Central gateways. These will usually be the internally managed ones. If there is no another Community defined for them, decide whether or not to mesh the central gateways. If they are already in a Community, do not mesh the central gateways.
    - Define the Satellite gateways. These will usually be the external ones.
5. Agree on a pre-shared secret with the administrator of the external Community members. Then, in the **Shared Secret** page of the community, select **Use Only Shared Secret for all External Members**. For each external peer, enter the pre-shared secret.
6. Define the relevant access rules in the Security Policy. Add the Community in the **VPN** column, the services in the **Service** column, the desired **Action**, and the appropriate **Track** option.
7. Install the Security Policy.

How to Authorize Firewall Control Connections in VPN Communities

Check Point Nodes communicate with other Check Point Nodes by means of control connections. For example, a control connection is used when the Security Policy is installed from the Security Management server to a Security Gateway. Also, logs are sent from security gateways to the Security Management server across control connections. Control connections use Secure Internal Communication (SIC).

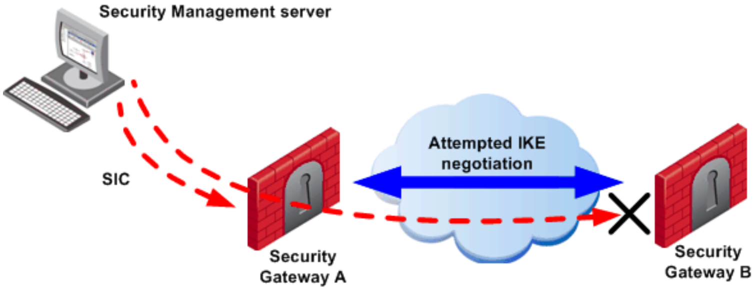
Control connections are allowed using Implied Rules in the Security Rule Base. Implied Rules are added to or removed from the Security Rule Base, by checking or unchecking options in the **FireWall Implied Rules** page of the SmartDashboard Global Properties.

Some administrators prefer not to rely on implied rules, and instead prefer to define explicit rules in the Security Rule Base.

Why Turning off FireWall Implied Rules Blocks Control Connections

If you turn off implicit rules, you may not be able to install a Policy on a remote security gateway. Even if you define explicit rules in place of the implied rules, you may still not be able to install the policy. Figure 4-14 and the following explanation illustrate the problem.

Figure 4-14 Turning off control connections can cause Policy installation to fail



The administrator wishes to configure a VPN between gateways A and B by configuring SmartDashboard. To do this, the administrator must install a Policy from the Security Management server to the gateways.

1. The Security Management server successfully installs the Policy on gateway A. As far as gateway A is concerned, gateways A and B now belong to the same VPN Community. However, B does not yet have this Policy.

2. The Security Management server tries to open a connection to gateway B in order to install the Policy.
3. Gateway A allows the connection because of the explicit rules allowing the control connections, and starts IKE negotiation with gateway B to build a VPN tunnel for the control connection.
4. Gateway B does not know how to negotiate with A because it does not yet have the Policy. Therefore Policy installation on gateway B fails.

The solution for this is to make sure that control connections do not have to pass through a VPN tunnel.

Allowing Firewall Control Connections Inside a VPN

If you turn off implied rules, you must make sure that control connections are not changed by the security gateways. To do this, add the services that are used for control connections to the **Excluded Services** page of the Community object.

Note - Even though control connections between the Security Management server and the gateway are not encrypted by the community, they are nevertheless encrypted and authenticated using Secure Internal Communication (SIC).

Discovering Which Services are Used for Control Connections

1. In the main menu, select **View > Implied Rules**.
2. In the Global Properties **FireWall** page, very that ‘control connections’ are accepted.
3. Examine the Security Rule Base to see what Implied Rules are visible. Note the services used in the Implied Rules.

Chapter 5

Public Key Infrastructure

In This Chapter:

Need for Integration with Different PKI Solutions

X.509-based PKI solutions provide the infrastructure that enables entities to establish trust relationships between each other based on their mutual trust of the Certificate Authority (CA). The trusted CA issues a certificate for an entity, which includes the entity's public key. Peer entities that trust the CA can trust the certificate — because they can verify the CA's signature — and rely on the information in the certificate, the most important of which is the association of the entity with the public key.

IKE standards recommend the use of PKI in VPN environments, where strong authentication is required.

A security gateway taking part in VPN tunnel establishment must have an RSA key pair and a certificate issued by a trusted CA. The certificate contains details about the module's identity, its public key, CRL retrieval details, and is signed by the CA.

When two entities try to establish a VPN tunnel, each side supplies its peer with random information signed by its private key and with the certificate that contains the public key. The certificate enables the establishment of a trust relationship between the gateways; each gateway uses the peer gateway's public key to verify the source of the signed information and the CA's public key to validate the certificate's authenticity. In other words, the validated certificate is used to authenticate the peer.

Every deployment of Check Point Security Management server includes an Internal Certificate Authority (ICA) that issues VPN certificates for the VPN modules it manages. These VPN certificates simplify the definition of VPNs between these modules. For more information about the ICA, see the *Security Management server Administration* Guide.

Situations can arise when integration with other PKI solutions is required, for example:

- A VPN must be established with a security gateway managed by an external Security Management server. For example, the peer gateway belongs to another organization which utilizes Check Point products, and its certificate is signed by its own Security Management server's ICA.
- A VPN must be established with a non-Check Point VPN entity. In this case, the peer's certificate is signed by a third-party CA.
- An organization may decide, for whatever reason, to use a third party CA to generate certificates for its gateways.

Supporting a Wide Variety of PKI Solutions

Check Point Security Gateways support many different scenarios for integrating PKI in VPN environments.

- **Multiple CA Support for Single VPN Tunnel** – Two gateways present a certificate signed by different ICAs.
- **Support for non-ICA CAs** – In addition to ICA, security gateways support the following Certificate Authorities:
  - External ICA - The ICA of another Security Management server
  - Other OPSEC certified PKI solutions
- **CA Hierarchy** – CAs are often arranged in an hierarchical structure where multiple CAs are subordinate to root authority or root CA. A subordinate CA is a Certificate Authority certified by another Certificate Authority. Subordinate CA's can issue certificates to other, more subordinate CAs, forming a certification chain or hierarchy.

PKI and Remote Access Users

The Check Point Suite supports certificates not only for gateways but for users as well. For more information, see "Introduction to Remote Access VPN" on page 269 for information about user certificates.

PKI Deployments and VPN

Following are some sample CA deployments:

- Simple Deployment - internal CA
- CA of an external Security Management server
- CA services provided over the Internet
- CA on the LAN

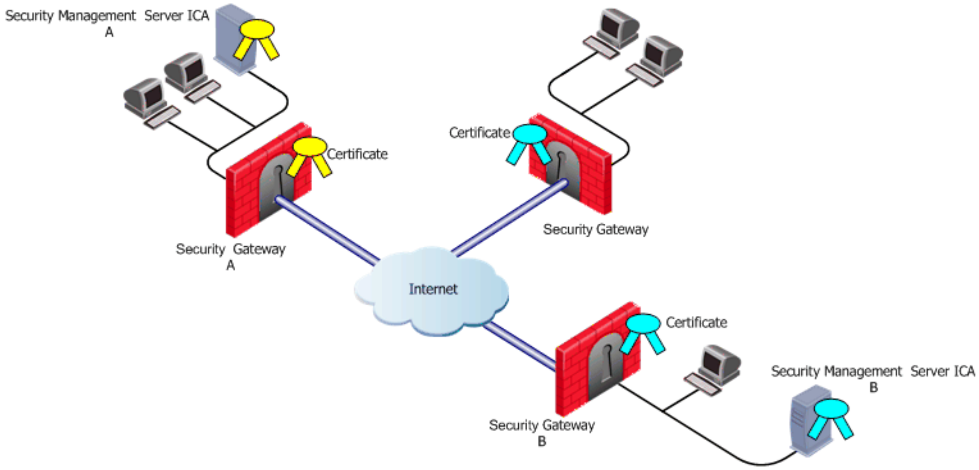
Simple Deployment – Internal CA

When the VPN tunnel is established between gateways managed by the same Security Management server, each peer has a certificate issued by the Security Management server's ICA.

CA of An External Security Management server

If a Check Point Security Gateway is managed by an external Security Management server (for example, when establishing a VPN tunnel with another organization's VPN modules), each peer has a certificate signed by its own Security Management server's ICA.

Figure 5-1 Two gateways managed by different Security Management servers

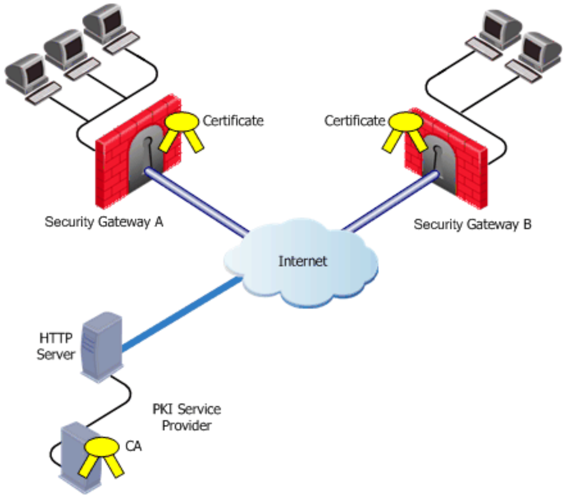


In Figure 5-1, Security Management server A issues certificates for security gateway A, while Security Management server B issues certificates for security gateway B.

CA Services Over the Internet

If a Check Point Security Gateway's certificate is issued by a third party CA accessible over the Internet, CA operations such as registration or revocation are usually performed through HTTP forms. CRLs are retrieved from an HTTP server functioning as a CRL repository. Figure 5-2 depicts a CA and CRL repository accessible over the Internet.

Figure 5-2 CA services are on the Internet

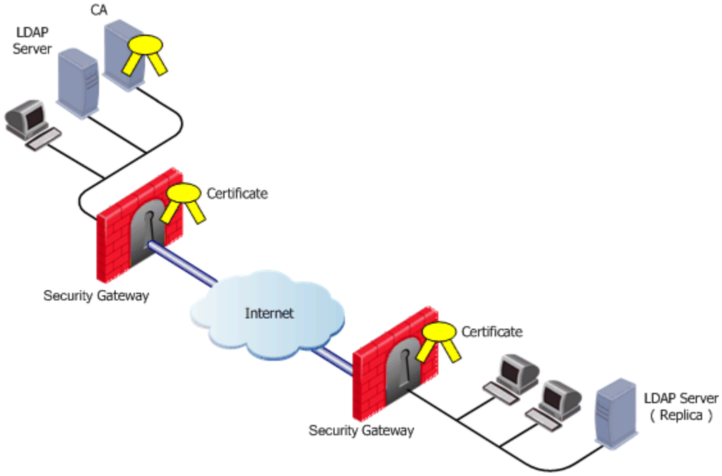


In Figure 5-3, gateways A and B receive their certificates from a PKI service provider accessible via the web. Certificates issued by external CA's may be used by gateways managed by the same Security Management server to verification.

CA is Located on the LAN

If the peer VPN gateway's certificate is issued by a third party CA on the LAN, the CRL is usually retrieved from an internal LDAP server, as shown in Figure 5-3.

Figure 5-3 Third Party CA deployed locally



Trusting An External CA

A trust relationship is a crucial prerequisite for establishing a VPN tunnel. However, a trust relationship is possible only if the CA that signs the peer's certificate is "trusted." Trusting a CA means obtaining and validating the CA's own certificate. Once the CA's Certificate has been validated, the details on the CA's certificate and its public key can be used to both obtain and validate other certificates issued by the CA.

The Internal CA (ICA) is automatically trusted by all modules managed by the Security Management server that employs it. External CAs (even the ICA of another Check Point Security Management server) are not automatically trusted, so a module must first obtain and validate an external CA's certificate. The external CA must provide a way for its certificate to be imported into the Security Management server.

If the external CA is:

- The ICA of an external Security Management server, see the *Security Management Server Administration Guide* for further information
- An OPSEC Certified CA, use the CA options on the **Servers and OSPEC Applications** tab to define the CA and obtain its certificate

Subordinate Certificate Authorities

A subordinate CA is a Certificate Authority certified by another Certificate Authority. Subordinate CAs can issue certificates to other, more subordinate CAs, in this way forming a certification chain or hierarchy. The CA at the top of the hierarchy is the root authority or root CA. Child Certificate Authorities of the root CA are referred to as Subordinate Certificate Authorities.

With the CA options on the **Servers and OSPEC Applications** tab, you can define either a Certificate Authority as either Trusted or Subordinate. Subordinate CAs are of the type OPSEC, and not trusted.

Enrolling a Managed Entity

Enrollment means obtaining a certificate from a CA, that is, requesting that the CA issue a certificate for an entity.

The process of enrollment begins with the generation of a key pair. A certificate request is then created out of the public key and additional information about the module. The type of the certificate request and the rest of the enrollment process depends on the CA type.

The case of an internally managed gateway is the simplest, because the ICA is located on the Security Management server machine. The enrollment process is completed automatically.

To obtain a certificate from an OPSEC Certified CA, Security Management server takes the module details and the public key and encodes a PKCS#10 request. The request (which can include *SubjectAltName* for OPSEC certificates and Extended Key Usage extensions) is delivered to the CA manually by the administrator. Once the CA issues the certificate the administrator can complete the process by importing the certificate to the Security Management server.

A certificate can also be obtained for the gateway using Automatic Enrollment. With Automatic Enrollment, you can automatically issue a request for a certificate from a trusted CA for any gateway in the community. Automatic Enrollment supports the following protocols:

- **SCEP**
- **CMPV1**
- **CMPV2**

Note - During SCEP enrollment, some HTTP requests may be larger than 2K, and may be dropped by the HTTP protocol inspection mechanism if enabled (**Web Intelligence > HTTP Protocol Inspection > HTTP Format Sizes**). A change of the default value will be required to enable these HTTP requests. If enrollment still fails, enrollment must be done manually. For more information, see the *IPS Administraton Guide*.

Validation of a Certificate

When an entity receives a certificate from another entity, it must:

1. Verify the certificate signature, i.e. verify that the certificate was signed by a trusted CA. If the certificate is not signed directly by a trusted CA, but rather by a subsidiary of a trusted CA, the path of CA certificates is verified up to the trusted CA.
2. Verify that the certificate chain has not expired.
3. Verify that the certificate chain is not revoked. A CRL is retrieved to confirm that the serial number of the validated certificate is not included among the revoked certificates.

In addition, VPN verifies the validity of the certificate's use in the given situation, confirming that:

- The certificate is authorized to perform the required action. For example, if the private key is needed to sign data (e.g., for authentication) the **KeyUsage** extension on the certificate – if present – is checked to see if this action is permitted.
- The peer used the correct certificate in the negotiation. When creating a VPN tunnel with an externally managed module, the administrator may decide that only a certificate signed by a specific CA from among the trusted CAs can be accepted. (Acceptance of certificates with specific details such as a *Distinguished Name* is possible as well).

Revocation Checking

There are two available methods useful in determining the status of a certificate:

1. CRL
2. Online Certificate Status Protocol (OCSP)

CRL

VPN can retrieve the CRL from either an HTTP server or an LDAP server. If the CRL repository is an HTTP server, the module uses the URL published in the CRL **Distribution Point** extension on the certificate and opens an HTTP connection to the CRL repository to retrieve the CRL.

If the CRL repository is an LDAP server, VPN attempts to locate the CRL in one of the defined LDAP account units. In this scenario, an LDAP account unit must be defined. If the CRL **Distribution Point** extension exists, it publishes the DN of the

CRL, namely, the entry in the Directory under which the CRL is published or the LDAP URI. If the extension does not exist, VPN attempts to locate the CRL in the entry of the CA itself in the LDAP server.

OCSP

Online Certificate Status Protocol (OCSP) enables applications to identify the state of a certificate. OCSP may be used for more timely revocation information than is possible with CRLs and may also be used to obtain additional status information. When OCSP client issues a status request to an OCSP server, acceptance of the certificate in question is suspended until the server provides a response.

In order to use OCSP, the root CA must be configured to use this method instead of CRL. This setting is inherited by the subordinate CA's.

CRL Prefetch-Cache

Since the retrieval of CRL can take a long time (in comparison to the entire IKE negotiation process), VPN stores the CRLs in a CRL cache so that later IKE negotiations do not require repeated CRL retrievals.

The cache is pre-fetched:

- every two hours
- on policy installation
- when the cache expires

If the pre-fetch fails, the previous cache is not erased.

Note - The ICA requires the use of a CRL cache.

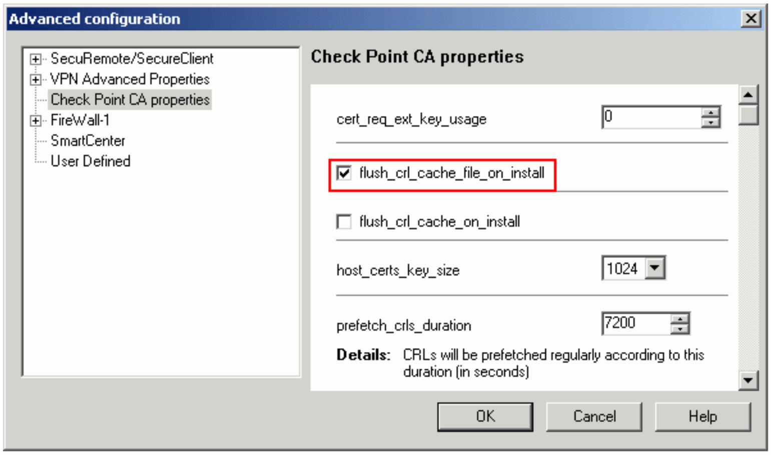
An administrator can shorten the lifetime of a CRL in the cache or even to cancel the use of the cache. If the CRL Cache operation is cancelled, the CRL must be retrieved for each subsequent IKE negotiation, thus considerably slowing the establishment of the VPN tunnel. Because of these performance implications, it is recommend that CRL caching be disabled only when the level of security demands continuous CRL retrieval.

Special Considerations for the CRL Pre-fetch Mechanism

The CRL pre-fetch mechanism makes a “best effort” to obtain the most up to date list of revoked certificates. However, after the `cpstop`, `cpstart` commands have been executed, the cache is no longer updated. The gateway continues to use the old CRL for as long as the old CRL remains valid (even if there is an updated CRL available on the CA). The pre-fetch cache mechanism returns to normal functioning only after the old CRL expires and a new CRL is retrieved from the CA.

In case there is a requirement that after `cpstop`, `cpstart` the CRL’s will be updated immediately, proceed as follows:

- After executing `cprestart`, run `crl_zap` to empty the cache, or:
- In **Global Properties > SmartDashboard Customization > Configure > Check Point CA properties >** select: **flush_crl_cache_file_on_install**.



Check Point CA properties

cert_req_ext_key_usage 0

☑ flush_crl_cache_file_on_install

□ flush_crl_cache_on_install

host_certs_key_size 1024

prefetch_crls_duration 7200

Details: CRLs will be prefetched regularly according to this duration (in seconds)

When a new policy is installed, the cache is flushed and a new CRL will be retrieved on demand.

CRL Grace Period

Temporary loss of connection with the CRL repository or slight differences between clocks on the different machines may cause valid of CRLs to be considered invalid—and thus the certificates to be invalid as well. VPN overcomes this problem by supplying a CRL Grace Period. During this period, a CRL is considered valid even if it is not valid according to the CLR validity time.

Special Considerations for PKI

In This Section

Using the Internal CA vs. Deploying a Third Party CA

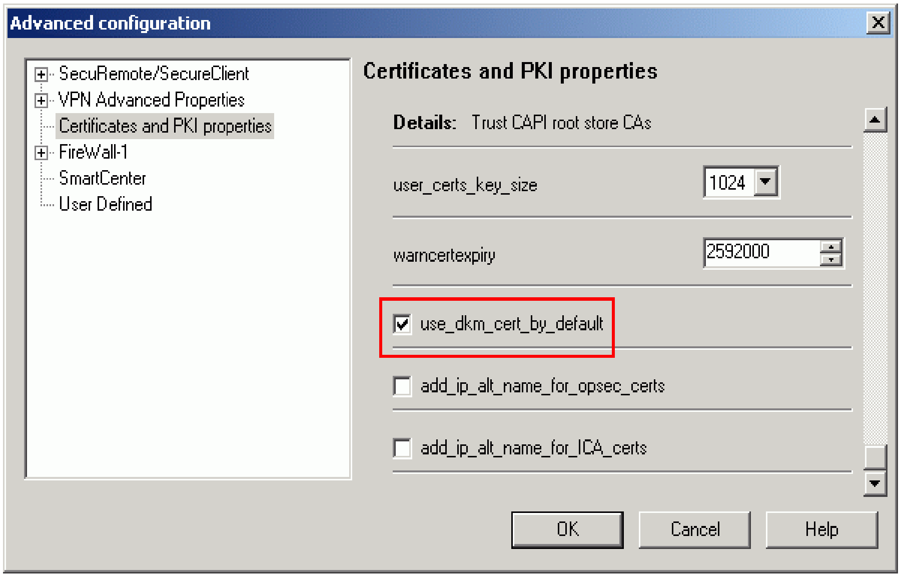
The Internal CA makes it easy to use PKI for Check Point applications such as site-to-site and remote access VPNs. However, an administrator may prefer to continue using a CA that is already functioning within the organization, for example a CA used to provide secure email, and disk encryption.

Distributed Key Management and Storage

Distributed Key Management (DKM) provides an additional layer of security during the key generation phase. Instead of the Security Management server generating both public and private keys and downloading them to the module during a policy installation, the management server instructs the module to create its own public and private keys and send (to the management server) only its public key. The private key is created and stored on the module in either a hardware storage device, or via software that emulates hardware storage. Security Management server then performs certificate enrollment. During a policy installation, the certificate is downloaded to the module. The private key never leaves the module.

Local key storage is supported for all CA types.

DKM is supported for all enrollment methods, and can be configured as a default setting by selecting in **Global Properties > SmartDashboard Customization > Configure > Certificates and PKI properties**, the option: **use_dkm_cert_by_default**



Advanced configuration

- SecuRemote/SecureClient
- VPN Advanced Properties
- Certificates and PKI properties
- FireWall-1
- SmartCenter
- User Defined

Certificates and PKI properties

Details: Trust CAPI root store CAs

user_certs_key_size: 1024

warncertexpiry: 2592000

☑ use_dkm_cert_by_default

□ add_ip_alt_name_for_opsec_certs

□ add_ip_alt_name_for_ICA_certs

Note - Generating certificates for Edge devices does not support DKM and will be generated locally on the management even if **use_dkm_cert_by_default** is configured.

Configuration of PKI Operations

In This Section

Trusting a CA – Step-By-Step

This section describes the procedures for obtaining a CA's own certificate, which is a prerequisite for trusting certificates issued by a CA.

In order to trust a CA, a CA server object has to be defined. The following sections deal with the various configuration steps required in different scenarios.

Trusting an ICA

A VPN module automatically trusts the ICA of the Security Management server that manages it. No further configuration is required.

Trusting an Externally Managed CA

An externally managed CA refers to the ICA of another Security Management server. The CA certificate has to be supplied and saved to disk in advance. To establish trust:

1. Open **Manage > Servers and OPSEC Applications**

   The **Servers and OPSEC Application** window opens.

2. Choose **New > CA**

   Select **Trusted...**

   The **Certificate Authority Properties** window opens.

3. Enter a **Name** for the CA object and in the **Certificate Authority Type** drop-down box select the **External Check Point CA**.
4. Go to the **External Check Point CA** tab and click **Get...**
5. Browse to where you saved the peer CA certificate and select it.

   VPN reads the certificate and displays its details. Verify the certificate's details. Display and validate the SHA-1 and MD5 fingerprints of the CA certificate.

6. Click **OK**.

Trusting an OPSEC Certified CA

The CA certificate has to be supplied and saved to the disk in advance.

Note - In case of SCEP automatic enrollment, you can skip this stage and fetch the CA certificate automatically after configuring the SCEP parameters.

The CA's Certificate must be retrieved either by downloading it using the CA options on the **Servers and OSPEC Applications** tab, or by obtaining the CA's certificate from the peer administrator in advance.

Then define the CA object according to the following steps:

1. Open **Manage > Servers and OPSEC Applications**

   The **Servers and OPSEC Application** window opens.

2. Choose **New > CA**

   Select **Trusted...** or **Subordinate...**

   The **Certificate Authority Properties** window opens.

3. Enter a **Name** for the CA object, in the **Certificate Authority Type** drop-down box select the **OPSEC PKI**.
4. On the **OPSEC PKI** tab:
   - For automatic enrollment, select **automatically enroll certificate**
   - From the **Connect to CA with protocol**, select the protocol used to connect with the certificate authority, either SCEP, CPMV1 or CPMV2.

Note - For entrust 5.0 and later, use CPMV1

5. Click **Properties...**
   - **If you chose SCEP as** the protocol, in the **Properties for SCEP protocol** window, enter the CA identifier (such as example.com) and the Certification Authority/Registration Authority URL.
   - If you chose cmpV1 as the protocol, in the **Properties for CMP protocol - V1** window, enter the appropriate IP address and port number. (The default port is 829).
   - If you chose cmpV2 as the protocol, in the **Properties for CMP protocol -V2** window, decide whether to use direct TCP or HTTP as the transport layer.

Note - If Automatic enrollment is not selected, then enrollment will have to be performed manually.

6. Choose a method for retrieving CRLs from this CA.

   If the CA publishes CRLs on HTTP server choose **HTTP Server(s)**. Certificates issued by the CA must contain the CRL location in an URL in the **CRL Distribution Point** extension.

   If the CA publishes CRL on LDAP server, choose **LDAP Server(s)**. In this case, you must define an LDAP Account Unit as well. See the *Security Management Server Adminstration guide* for more details about defining an LDAP object.

   Make sure that **CRL retrieval** is checked in the **General** tab of the **LDAP Account Unit Properties** window.

   Certificates issued by the CA must contain the LDAP DN on which the CRL resides in the CRL distribution point extension.
7. Click **Get...**

8. If SCEP is configured, it will try to connect to the CA and retrieve the certificate. If not, browse to where you saved the peer CA certificate and select it.

   VPN reads the certificate and displays its details. Verify the certificate's details. Display and validate the SHA-1 and MD5 fingerprints of the CA certificate.

9. Click **OK**.

Enrolling with a Certificate Authority

A certificate is automatically issued by the ICA for all internally managed entities that are VPN capable. That is, after the administrator has checked the **VPN** option in the **Check Point Products** area of a network objects **General Properties** tab.

The process for obtaining a certificate from a OPSEC PKI or External Check Point CA is identical.

Manual Enrollment with OPSEC Certified PKI

To create a PKCS#10 Certificate Request:

1. Create the CA object, as described in "Trusting an OPSEC Certified CA" on page 104
2. Open the **VPN** tab of the relevant Network Object.
3. In the **Certificate List** field click **Add...**

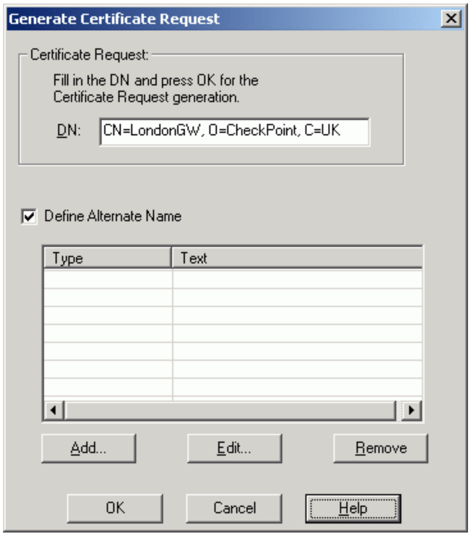
   The **Certificate Properties** window is displayed.
4. Enter the **Certificate Nickname**

   The nickname is only an identifier and has no bearing on the content of the certificate.
5. From the **CA to enroll from** drop-down box, select the direct OPSEC CA/External CheckPoint CA that will issue the certificate.

Note - The list displays only those subordinate CA's that lead directly to a trusted CA and the trusted CAs themselves. If the CA that issues the certificate is a subordinate CA that does not lead directly to a trusted CA, the subordinate CA will not appear in the list.

6. Choose the appropriate method for Key Pair creation and storage. See "Distributed Key Management and Storage" on page 101 for more information.
7. Click **Generate...**

The **Generate Certificate Properties** window is displayed.



Generate Certificate Request

Certificate Request:

Fill in the DN and press OK for the Certificate Request generation.

DN: CN=LondonGW, O=CheckPoint, C=UK

☑ Define Alternate Name

Type	Text

8. Enter the appropriate DN.

   The final DN that appears in the certificate is decided by the CA administrator.

   If a **Subject Alternate Name** extension is required in the certificate, check the **Define Alternate Name** check box.

   Adding the object IP as Alternate name extension can be configured as a default setting by selecting in **Global Properties > SmartDashboard Customization > Configure > Certificates and PKI properties**, the options:

   add_ip_alt_name_for_opsec_certs

   add_ip_alt_name_for_ICA_certs

   The configuration in this step is also applicable for Internal CA’s.

9. Click **OK.**

   The public key and the DN are then used to DER-encode a PKCS#10 Certificate Request.

10. Once the Certificate Request is ready, click **View...**

    The **Certificate Request View** window appears with the encoding.

11. Copy the whole text in the window and deliver it to the CA.

The CA administrator must now complete the task of issuing the certificate. Different CAs provide different ways of doing this, such as an advanced enrollment form (as opposed to the regular form for users). The issued certificate may be delivered in various ways, for example email. Once the certificate has arrived, it needs to be stored:

A. Go to the **Severs and OPSEC Applications** tab of the network object, select the appropriate CA object.

B. On the OPEC PKI tab, click **Get...** and browse to the location in which the certificate was saved.

C. Select the appropriate file and verify the certificate details.

D. Close object and save.

Automatic Enrollment with the Certificate Authority

On the OPSEC PKI tab of the CA object, make sure **Automatically enroll certificate** is selected and SCEP or CMP are chosen as the connecting protocol. Then:

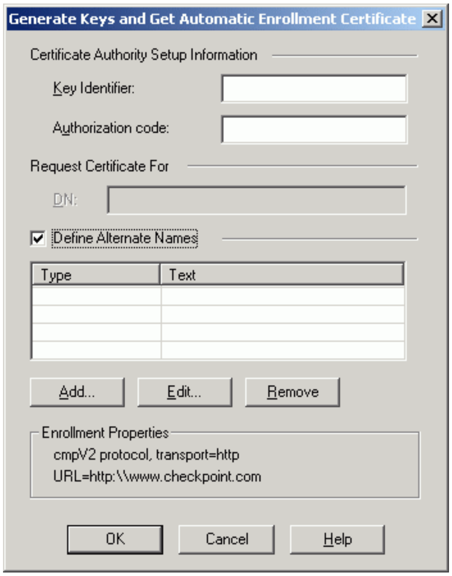
1. On the relevant network object, open the **VPN** tab.
2. In the **Certificates List** section, click **Add...**

   The **Certificate Properties** window opens.
3. Enter a **Certificate Nickname** (any string used as an identifier)
4. From the drop-down list box, select the CA that issues the certificate.

Note - The list displays only those subordinate CA's that lead directly to a trusted CA and the trusted CAs themselves. If the CA that issues the certificate is a subordinate CA that does not lead directly to a trusted CA, the subordinate CA will not appear in the list.

5. Select a method for key pair generation and storage.
6. Click **Generate**, and select **Automatic enrollment.**

   The **Generate Keys and Get Automatic Enrollment Certificate** window opens.



Generate Keys and Get Automatic Enrollment Certificate

Certificate Authority Setup Information

Key Identifier:

Authorization code:

Request Certificate For

DN:

☑ Define Alternate Names

Type	Text

Add... Edit... Remove

Enrollment Properties

cmpV2 protocol, transport=http

URL=http:\\www.checkpoint.com

OK Cancel Help

- Supply the **Key Identifier** and your secret **authorization code**.
- Click **OK.**

7. When the certificate appears in the **Certificates List** on the network objects VPN page, click **View** and either **Copy to Clipboard** or **Save to File** the text in the **Certificate Request View** window.
8. Send the request to CA administrator.

   Different Certificate Authorities provide different means for doing this, for example an advanced enrollment form on their website. The issued certificate can be delivered in various ways, such as email. Once you have received the certificate, save it to disk.
9. On the **VPN** tab of the network object, select the appropriate certificate in the **Certificates List**, and click **Complete...**
10. Browse to the folder where you stored the issued certificate, select the certificate and verify the certificate details.
11. Close the network object and **Save**.

Enrolling through a Subordinate CA

When enrolling through a subordinate CA:

- Supply the password of the subordinate CA which issues the certificate, not the CA at the top of the hierarchy
- The subordinate CA must lead directly to a trusted CA

Certificate Revocation (All CA Types)

A certificate issued by the Internal Certificate Authority it is revoked when the certificate object is removed. Otherwise, certificate revocation is controlled by the CA administrator using the options on the **Advanced** tab of the CA object. In addition, the certificate must be removed from the module.

To remove the certificate proceed as follows:

1. Open the **VPN** tab of the relevant Network Object.
2. In the **Certificate List** field select the appropriate certificate and click **Remove**.

   A certificate cannot be removed if Smart Center server infers from other settings that the certificate is in use, for example, that the module belongs to one or more VPN communities and this is the module’s only certificate.

Certificate Recovery and Renewal

When a certificate is revoked or becomes expired, it is necessary to create another one or to refresh the existing one.

Recovery and Renewal with Internal CA

Removal of a compromised or expired certificate automatically triggers creation of a new certificate, with no intervention required by the administrator. To manually renew a certificate use the **Renew...** button on the VPN page of the gateway object.

Note - A module can have only one certificate signed by a particular CA. Thus, when the new certificate is issued, you will be asked whether to replace any existing certificate signed by the same CA.

Adding Matching Criteria to the Validation Process

While certificates of an externally managed VPN entity are not handled by the local Security Management server, you can still configure a peer to present a particular certificate when creating a VPN tunnel:

1. Open the **VPN** page of the externally managed VPN entity.
2. Click **Matching Criteria...**
3. Choose the desired characteristics of the certificate the peer is expected to present, including:
   - The CA that issued it
   - The exact DN of the certificate
   - The IP address that appears in the **Subject Alternate Name** extension of the certificate. (This IP address is compared to the IP address of the VPN peer itself as it appears to the VPN module during the IKE negotiation.)
   - The e-mail address appearing in the **Subject Alternate Name** extension of the certificate

CRL Cache Usage

To cancel or modify the behavior of the CRL Cache:

1. Open the **Advanced Tab** of the Certificate Authority object.
2. To enable the CRL cache, check **Cache CRL on the module**.

   The cache should not be disabled for the ICA. In general, it is recommended that the cache be enabled for all CA types. The cache should be disabled (for non-ICAs) only if stringent security requirements mandate continual retrieval of the CRL.

Note - The ICA requires the use of a CRL cache, and should never be disabled.

3. If CRL Cache is enabled, choose whether a CRL is deleted from the cache when it expires or after a fixed period of time (unless it expires first). The second option encourages retrieval of a CRL more often as CRLs may be issued more frequently then the expiry time. By default a CRL is deleted from the cache after 24 hours.

See: “CRL Prefetch-Cache” on page 99 for information about CRL caching.

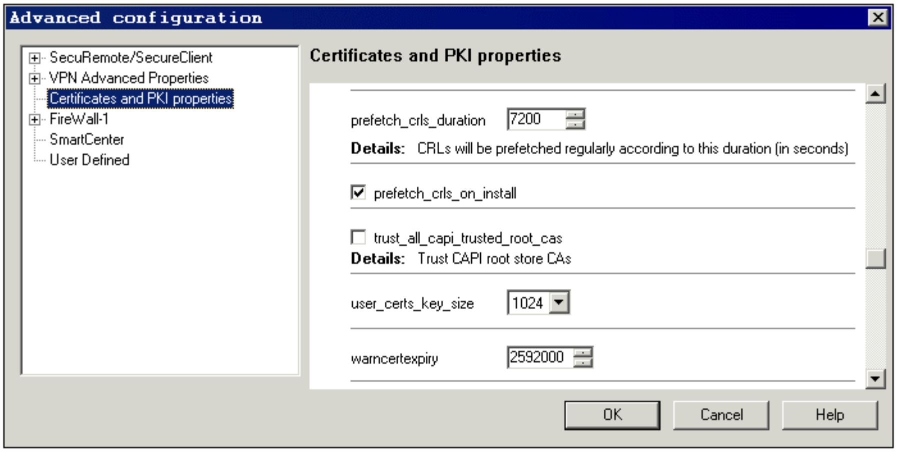
Modifying the CRL Pre-Fetch Cache

The behavior of the Pre-fetch catch can be altered via the Global properties:

1. **Global Properties > SmartDashboard Customization > Configure...** button

   The **Advanced Configuration** window opens.

2. Select Check Point CA Properties:



Certificates and PKI properties

prefetch_crls_duration 7200
Details: CRLs will be prefetched regularly according to this duration (in seconds)

☑ prefetch_crls_on_install

☐ trust_all_capi_trusted_root_cas
Details: Trust CAPI root store CAs

user_certs_key_size 1024

warncertexpiry 2592000

Configuring CRL Grace Period

Set the CRL Grace Period values by selecting **Policy > Global Properties > VPN > Advanced**. The Grace Period can be defined for both the periods before and after the specified CRL validity period.

Configuring OCSP

In order to use OCSP, the CA object must be configured to the OCSP revocation checking method instead of CRL's.

Using **Dbedit**, modify the field `oscp_validation` to **true**. Set to true, this CA will check the validation of the certificate using OCSP. This is configured on the root CA and is inherited by the subordinate CA's.

To configure a trusted OCSP server using **Dbedit** of `objectc.c`:

1. Create a new server object of the type `oscp_server`.
2. Configure the OCSP servers URL and the certificate.
3. In the CA object, configure oscp_server. Add a reference to the OCSP server object created and install policy.

Site-to-Site VPN

Chapter 5

Domain Based VPN

In This Chapter

Overview

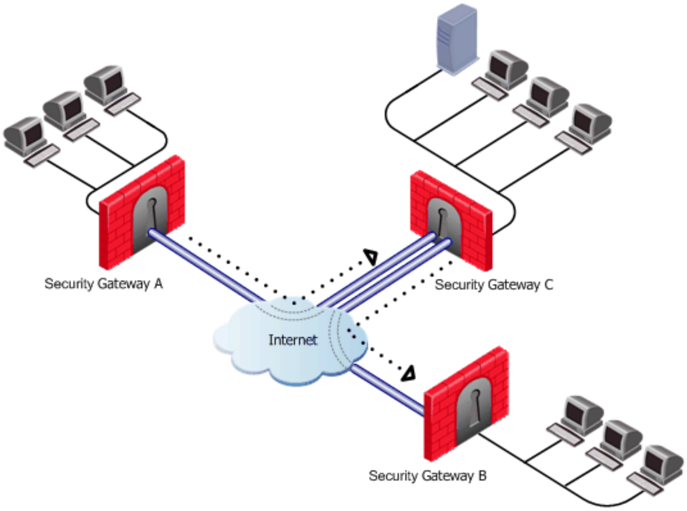
Domain Based VPN is a method of controlling how VPN traffic is routed between gateway modules and remote access clients within a community.

To route traffic to a host behind a gateway, an encryption domain must be configured for that gateway.

Configuration for VPN routing is performed either directly through SmartDashboard or by editing the VPN routing configuration files on the gateways.

In Figure 5-1, one of the host machines behind gateway A initiates a connection with a host machine behind gateway B. For either technical or policy reasons, gateway A cannot establish a VPN tunnel with gateway B. Using VPN Routing, both gateways A and B can establish VPN tunnels with gateway C, so the connection is routed through gateway C.

Figure 5-1 Simple VPN routing



VPN Routing and Access Control

VPN routing connections are subject to the same access control rules as any other connection. If VPN routing is correctly configured but a Security Policy rule exists that does not allow the connection, the connection is dropped. For example: a gateway has a rule which forbids all FTP traffic from inside the internal network to anywhere outside. When a peer gateway opens an FTP connection with this gateway, the connection is dropped.

For VPN routing to succeed, a single rule in the Security Policy Rule base must cover traffic in both directions, inbound and outbound, and on the central gateway. To configure this rule, see "Configuring the 'Accept VPN Traffic Rule'" on page 125.

Configuring Domain Based VPN

In This Section

Common VPN routing scenarios can be configured through a VPN star community, but not all VPN routing configuration is handled through SmartDashboard. VPN routing between gateways (star or mesh) can be also be configured by editing the configuration file `$FWDIR\conf\vpn_route.conf`.

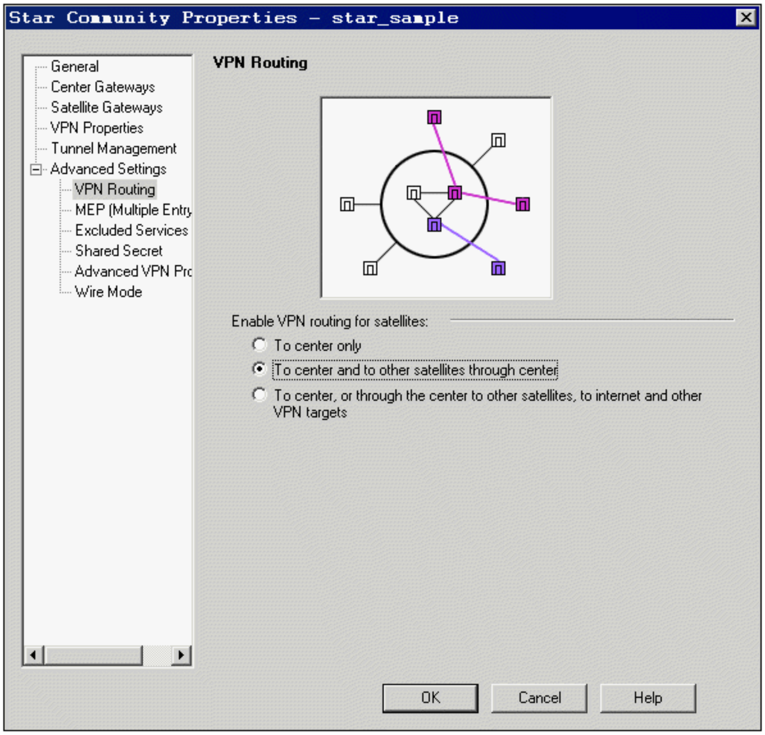
VPN routing cannot be configured between gateways that do not belong to a VPN community.

Configuring VPN Routing for Gateways via SmartDashboard

For simple hubs and spokes (or situations in which there is only one Hub) the easiest way is to configure a VPN star community in SmartDashboard:

1. On the **Star Community properties** window, **Central Gateways page**, select the gateway that functions as the “Hub”.
2. On the **Satellite Gateways** page, select gateways as the “spokes”, or satellites.
3. On the **VPN Routing** page, **Enable VPN routing for satellites** section, select one of these options:
   - **To center and to other Satellites through center**. This allows connectivity between the gateways, for example if the spoke gateways are DAIP gateways, and the Hub is a gateway with a static IP address.
   - **To center, or through the center to other satellites, to internet and other VPN targets**. This allows connectivity between the gateways as well as the ability to inspect all communication passing through the Hub to the Internet.

Figure 5-2 Satellites communicating through the center



VPN Routing

Enable VPN routing for satellites:

○ To center only

● To center and to other satellites through center

○ To center, or through the center to other satellites, to internet and other VPN targets

4. Create an appropriate access control rule in the Security Policy Rule Base. Remember: *one rule must cover traffic in both directions*.
5. NAT the satellite gateways on the Hub if the Hub is used to route connections from Satellites to the Internet.

Note - Disabling NAT in the community (available in Star and Meshed communities in the **Advanced VPN Properties** tab) is not supported if one of the internally managed members is of version earlier than NG FP3.

The two DAIP gateways can securely route communication through the gateway with the static IP address.

To configure the VPN routing option **To center and to other satellites through center** with SmartLSM security gateways:

1. Create a network object that contains the VPN domains of all the gateways managed by SmartProvisioning.
2. Edit the `vpn_route.conf` file, so that this network object appears in the "router" column (the center gateway of the star community).

3. Install this `vpn_route.conf` file on all LSM profiles that participate in the VPN community.

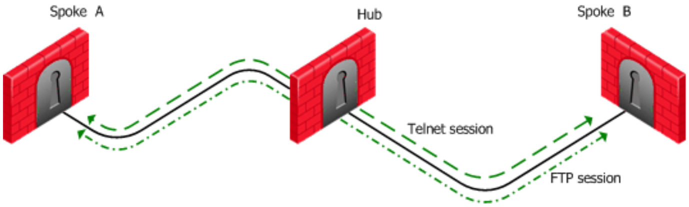
Configuration via Editing the VPN Configuration File

For more granular control over VPN routing, edit the `vpn_route.conf` file in the `conf` directory of the Security Management server.

The configuration file, `vpn_route.conf`, is a text file that contains the name of network objects. The format is: **Destination**, **Next hop**, **Install on Gateway** (with tabbed spaces separating the elements).

Consider a simple VPN routing scenario consisting of Hub and two Spokes (Figure 5-3). All machines are controlled from the same Security Management server, and all the security gateways are members of the same VPN community. Only Telnet and FTP services are to be encrypted between the Spokes and routed through the Hub:

Figure 5-3 Filtering Telnet and FTP



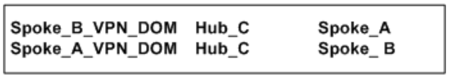
Although this could be done easily by configuring a VPN star community, the same goal can be achieved by editing `vpn_route.conf`:

Table 5-1

Destination	Next hop router interface	Install On
Spoke_B_VPN_Dom	Hub_C	Spoke_A
Spoke_A_VPN_Dom	Hub_C	Spoke_B

In this instance, Spoke_B_VPN_Dom is the name of the network object group that contains spoke B's VPN domain. Hub C is the name of the security gateway enabled for VPN routing. Spoke_A_VPN_Dom is the name of the network object that represents Spoke A's encryption domain. See Figure 5-4 for an example of how the file appears:

Figure 5-4 vpn_route.conf



```
Spoke_B_VPN_DOM   Hub_C          Spoke_A
Spoke_A_VPN_DOM   Hub_C          Spoke_ B
```

Configuring the ‘Accept VPN Traffic Rule’

In SmartDashboard:

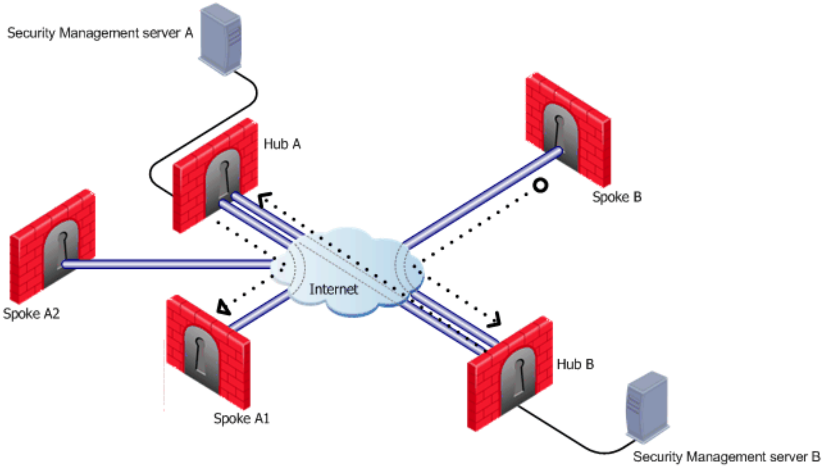
1. Double click on a Star or Meshed community.
2. On the **General** properties page, select the **Accept all encrypted traffic** checkbox.
3. In a Star community, click **Advanced** to choose between accepting encrypted traffic on **Both center and satellite Gateways** or **Satellite Gateways only**.
4. Click **OK**.

A rule will appear in the Rule Base that will accept VPN traffic between the selected gateways.

Configuring Multiple Hubs

Figure 5-5 shows two Hubs, A and B. Hub A has two spokes, spoke_A1, and spoke_A2. Hub B has a single spoke, spoke_B. In addition, Hub A is managed from Security Management server A, while Hub B is managed via Security Management server B:

Figure 5-5 Configuring multiple vpn_route.conf files



For the two VPN star communities, based around Hubs A and B:

- Spokes A1 and A2 need to route all traffic going outside of the VPN community through Hub A
- Spokes A1 and A2 also need to route all traffic to one another through Hub A, the center of their star community
- Spoke B needs to route all traffic outside of its star community through Hub B

A_community is the VPN community of A plus the spokes belonging to A. B_community is the VPN community. Hubs_community is the VPN community of Hub_A and Hub_B.

Configuring VPN Routing and Access Control on Security Management server A

The `vpn_route.conf` file on Security Management server 1 looks like this:

Table 5-2

Destination	Next hop router interface	Install On
Spoke_B_VPN_Dom	Hub_A	A_Spokes
Spoke_A1_VPN_Dom	Hub_A	Spoke_A2
Spoke_A2_VPN_Dom	Hub_A	Spoke _A1
Spoke_B_VPN_Dom	Hub_B	Hub_A

Spokes A1 and A2 are combined into the network group object “A_spokes”. The appropriate rule in the Security Policy Rule Base looks like this:

Table 5-3

Source	Destination	VPN	Service	Action
Any	Any	A_Community B_Community Hubs_Community	Any	Accept

Configuring VPN Routing and Access Control on Security Management server B

The vpn_route.conf file on Security Management server 2 looks like this:

Table 5-4

Destination	Next hop router interface	Install On
Spoke_A1_VPN_Dom	Hub_B	Spoke_B
Spoke_A2_VPN_Dom	Hub_B	Spoke_B
Spoke_A1_VPN_Dom	Hub_A	Hub_B
Spoke_A2_VPN_Dom	Hub_A	Hub_B

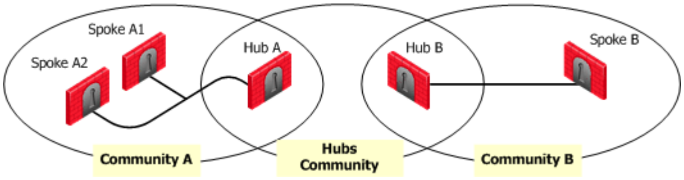
The appropriate rule in the Security Policy Rule Base looks like this:

Table 5-5

Source	Destination	VPN	Service	Action
Any	Any	B_Community A_Community Hubs_Community	Any	Accept

For both vpn_route.conf files:

- “A_Community” is a star VPN community comprised of Hub_A, Spoke_A1, and Spoke_A2
- “B_Community” is a star VPN community comprised of Hub_B and Spoke_B
- “Hubs-Community” is a *meshed* VPN community comprised of Hub_A and Hub_B (it could also be a star community with the central gateways meshed).

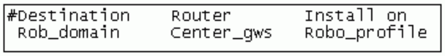


Configuring SmartLSM Security Gateways

If branch office gateways are managed by SmartProvisioning as SmartLSM security gateways, enable VPN routing for a hub and spoke configuration by editing the `vpn_route.conf` file on the Security Management server.

In SmartDashboard:

1. Generate a group that contains the encryption domains of all the satellite SmartLSM security gateways and call it **Robo_domain**
2. Generate a group that contains all the central gateways and call it **Center_gws**
3. In vpn_route.conf, add the rule:



```
#Destination    Router       Install on
 Rob_domain     Center_gws   Robo_profile
```

If access to the SmartLSM security gateway through the VPN tunnel is required, the gateway's external IP address should be included in the ROBO_domain.

Multiple router gateways are now supported on condition that:

- the gateways are listed under "install on" in `vpn_route.conf` or
- the satellites gateways are selected in SmartDashboard

Chapter 6
Route Based VPN

In This Chapter

Overview

The use of VPN Tunnel Interfaces (VTI) introduces a new method of configuring VPNs called *Route Based VPN*. This method is based on the notion that setting up a VTI between peer gateways is much like connecting them directly.

A VTI is an operating system level virtual interface that can be used as a gateway to the encryption domain of the peer gateway. Each VTI is associated with a single tunnel to a security gateway. The tunnel itself with all its properties is defined, as before, by a VPN Community linking the two gateways. The peer gateway should also be configured with a corresponding VTI. The native IP routing mechanism on each gateway can then direct traffic into the tunnel just as it would for any other type of interface.

All traffic destined to the encryption domain of a peer gateway, will be routed through the “associated” VTI. This infrastructure allows dynamic routing protocols to use VTIs. A dynamic routing protocol daemon running on the security gateway can exchange routing information with a neighboring routing daemon running on the other end of an IPSec tunnel, which appears to be a single hop away.

Route Based VPN is supported using SecurePlatform and Nokia IPSO 3.9 platforms only and can only be implemented between two gateways within the same community.

VPN Tunnel Interface (VTI)

A VPN Tunnel Interface is a virtual interface on a security gateway that is associated with an existing VPN tunnel, and is used by IP routing as a point to point interface directly connected to a VPN peer gateway.

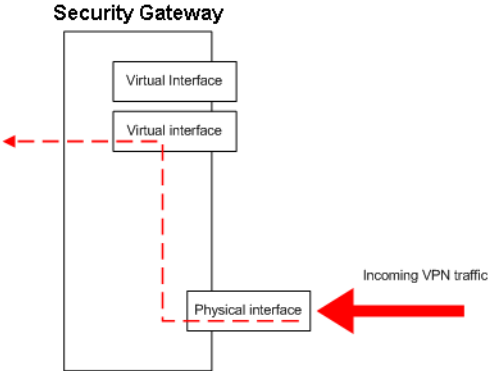
The VPN routing process of an outbound packet can be described as follows:

- An IP packet with destination address X is matched against the routing table.
- The routing table indicates that IP address X should be routed through a point to point link, which is the VPN Tunnel Interface that is associated with peer gateway Y.
- The VPN kernel intercepts the packet as it enters the virtual tunnel interface.
- The packet is encrypted using the proper IPsec Security Association parameters with peer gateway Y as defined in the VPN Community, and the new packet receives the peer gateway Y's IP address as the destination IP.
- Based on the new destination IP, the packet is rerouted to the physical interface according to the appropriate routing table entry for Y's address.

The opposite is done for inbound packets:

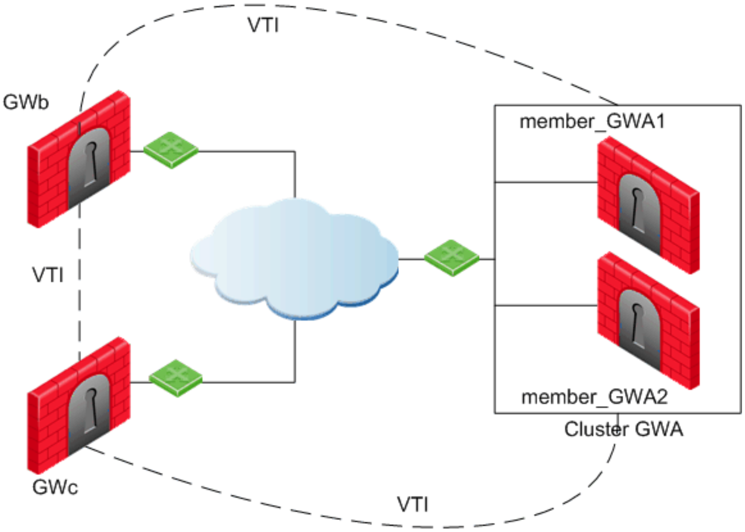
- An IPsec packet enters the machine coming from gateway Y.
- The VPN kernel intercepts the packet on the physical interface.
- The VPN kernel identifies the originating VPN peer gateway.
- The VPN kernel decapsulates the packet, and extracts the original IP packet.
- The VPN kernel detects that a VPN Tunnel Interface exists for the peer VPN gateway, and reroutes the packet from the physical interface to the associated VPN Tunnel Interface.
- The packet enters the IP stack through the VPN Tunnel Interface.

Figure 6-1 Routing to a Virtual Interface



In Route Based VPN, VTIs are created on the local gateway. Each VTI is associated with a corresponding VTI on a remote peer. Traffic routed from the local gateway via the VTI is transferred encrypted to the associated peer gateway.

Figure 6-2 Route Based VPN



In this scenario:

- There is a VTI connecting Cluster GWA and GWb
- There is a VTI connecting Cluster GWA and GWc
- There is a VTI connecting GWb and GWc

A virtual interface behaves like a point-to-point interface directly connected to the remote peer. Traffic between network hosts is routed into the VPN tunnel using the IP routing mechanism of the Operating System. Gateway objects are still required, as well as VPN communities (and access control policies) to define which tunnels are available. However, VPN encryption domains for each peer gateway are no longer necessary. The decision whether or not to encrypt depends on whether the traffic is routed through a virtual interface. The routing changes dynamically if a dynamic routing protocol (OSPF/BGP) is available on the network.

Note - For NGX (R60) and above, the dynamic routing suite has been incorporated into SecurePlatform Pro. The administrator runs a daemon on the gateway to publish the changed routes to the network.

When a connection that originates on GWb is routed through a VTI to GWc (or servers behind GWc) and is accepted by the implied rules, the connection leaves GWb in the clear with the local IP address of the VTI as the source IP address. If this IP address is not routable, return packets will be lost.

The solution for this issue is:

- configure a static route on GWb that redirects packets destined to GWc from being routed through the VTI.
- not including it in any published route
- adding route maps that filter out GWc's IP addresses.

Having excluded those IP addresses from route-based VPN, it is still possible to have other connections encrypted to those addresses (i.e. when not passing on implied rules) by using domain based VPN definitions.

The VTI may be configured in two ways:

- Numbered
- Unnumbered

Numbered VTI

If the VPN Tunnel Interface is numbered, the interface is assigned a local IP Address and a remote IP Address. The local IP Address will be the source IP for the connections originating from the gateway and going through the VTI. VTIs may share an IP Address but cannot use an already existing physical interface IP address. Numbered interfaces are only supported using the SecurePlatform Operating System.

Unnumbered VTI

If the VTI is unnumbered, local and remote IP addresses are not configured. Unnumbered VTIs must be assigned a proxy interface. The proxy interface is used as the source IP for outbound traffic. Unnumbered interfaces eliminate the need to allocate and manage an IP address per interface. Unnumbered interfaces are only supported on the Nokia IPSO 3.9 platform.

Nokia IPSO interfaces may be physical or loopback.

Using Dynamic Routing Protocols

VTIs allow the ability to use Dynamic Routing Protocols to exchange routing information between gateways. The Dynamic Routing Protocols supported are:

1. BGP4
2. OSPF
3. RIPv1 (SecurePlatform Pro only)
4. RIPv2 (SecurePlatform Pro only)

Configuring Numbered VTIs

Route Based VPN is supported using SecurePlatform and Nokia IPSO 3.9 platforms only and can only be implemented between two gateways within the same community.

Enabling Route Based VPN

If both Domain Based VPN and Route Based VPN are configured, then Domain Based VPN will take precedence. For Route Based VPN to take priority, an empty group should be created and assigned as the VPN domain.

In SmartDashboard, proceed as follows:

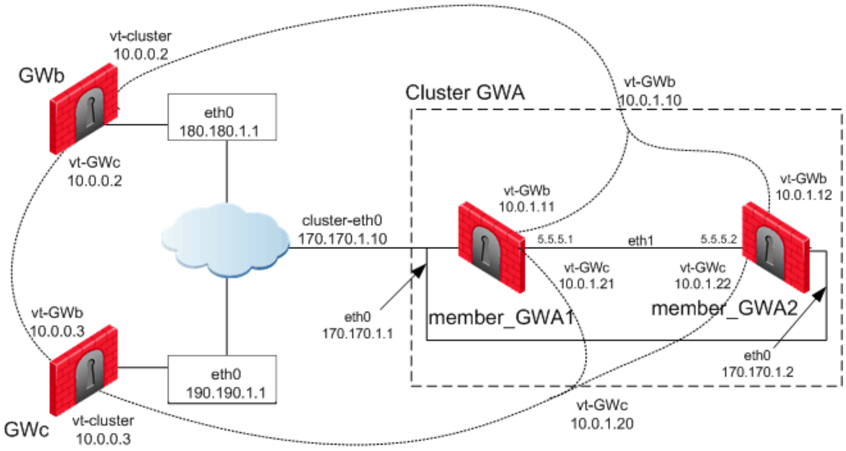
1. Select **Manage > Network Objects**.
2. Select the Check Point gateway and right click **Edit**.
3. In the Properties list, click **Topology**.
4. In the **VPN Domain** section, select **Manually define**.
5. Click **New > Group > Simple Group**.
6. Enter a name in the **Name** field and click **OK**.

Numbered VTIs

Using the new VPN Command Line Interface (VPN Shell), the administrator creates a VPN Tunnel Interface on the enforcement module for each peer gateway, and “associates” the interface with a peer gateway. The VPN Tunnel Interface may be numbered or unnumbered. For more information on the VPN Shell, see “VPN Shell” on page 729.

Every numbered VTI is assigned a local IP Address and a remote IP Address. Prior to configuration, a range of IP Addresses must be configured to assign to the VTIs.

Figure 6-3



In Figure 6-3:

- There is a VTI connecting Cluster GWA and GWb
- There is a VTI connecting Cluster GWA and GWc
- There is a VTI connecting GWb and GWc

The devices in this scenario are:

ClusterXL:

- Cluster GWA
  - member_GWA1
  - member_GWA2

VPN Modules:

- GWb
- GWc

IP Configurations:

- Cluster GWA
  - member_GWA1
    - External Unique IP eth0: 170.170.1.1/24
    - External VIP eth0: 170.170.1.10/24
    - Sync Interface eth1: 5.5.5.1/24

  - IP of VTI vt-GWb: Local: 10.0.1.11, Remote: 10.0.0.2
  - VIP of VTI vt-GWb: 10.0.1.10
  - IP of VTI vt-GWc: Local: 10.0.1.21, Remote: 10.0.0.3
  - VIP of VTI vt-GWc: 10.0.1.20
- member_GWA2
  - External Unique IP eth0: 170.170.1.2/24
  - External VIP eth0: 170.170.1.10/24
  - Sync Interface eth1: 5.5.5.1/24
  - IP of VTI vt-GWb: Local: 10.0.1.12, Remote: 10.0.0.2
  - VIP of VTI vt-GWb: 10.0.1.10
  - IP of VTI vt-GWc: Local: 10.0.1.22, Remote: 10.0.0.3
  - VIP of VTI vt-GWc: 10.0.1.20
- GWb
  - External Unique IP eth0: 180.180.1.1/24
  - IP of VTI vt-ClusterGWa: Local: 10.0.0.2, Remote: 10.0.1.10
  - IP of VTI vt-GWc: Local: 10.0.0.2, Remote: 10.0.0.3
- GWc
  - External Unique IP eth0: 190.190.1.1/24
  - IP of VTI vt-ClusterGWa: Local: 10.0.0.3, Remote: 10.0.1.20
  - IP of VTI vt-GWb: Local: 10.0.0.3, Remote: 10.0.0.2

VTIs in a Clustered Environment

When configuring numbered VTIs in a clustered environment, a number of issues need to be considered:

- Each member must have a unique source IP address.
- Every interface on each member requires a unique IP address.
- All VTIs going to the same remote peer must have the same name.
- Cluster IP addresses are required.

Configuring VTIs in a Clustered Environment

The following sample configurations use the same gateway names and IP addresses used in Figure 6-3 on page 137.

Table 6-1 Configuring member_GWA1

```
--------- Access the VPN shell Command Line Interface
[member_GWa1]# vpn shell
 ?             - This help
 ..            - Go up one level
 quit          - Quit
[interface   ] - Manipulate tunnel interfaces
[show        ] - Show internal data
[tunnels     ] - Manipulate tunnel data
--------- Add vt-GWb
VPN shell:[/] > /interface/add/numbered 10.0.1.11 10.0.0.2 GWb
Interface 'vt-GWb' was added successfully to the system
--------- Add vt-GWc
VPN shell:[/] > /interface/add/numbered 10.0.1.21 10.0.0.3 GWc
Interface 'vt-GWc' was added successfully to the system
---------- Verify configuration
VPN shell:[/] > /show/interface/detailed all
vt-GWb     Type:numbered  MTU:1500
         inet addr:10.0.1.11  P-t-P:10.0.0.2  Mask:255.255.255.255
           Peer:GWb  Peer ID:180.180.1.1  Status:attached

vt-GWc     Type:numbered  MTU:1500
         inet addr:10.0.1.21  P-t-P:10.0.0.3  Mask:255.255.255.255
           Peer:GWc  Peer ID:190.190.1.1  Status:attached

VPN shell:[/] > /quit
[member_GWa1]# ifconfig vt-GWb
vt-GWb    Link encap:IPIP Tunnel   HWaddr
         inet addr:10.0.1.11  P-t-P:10.0.0.2  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)

[member_GWa1]# ifconfig vt-GWc
vt-GWc    Link encap:IPIP Tunnel   HWaddr
         inet addr:10.0.1.21  P-t-P:10.0.0.3  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)
```

Table 6-2 Configuring member_GWA2

```
--------- Access the VPN shell Command Line Interface
[member_GWa2]# vpn shell
 ?              - This help
 ..             - Go up one level
 quit           - Quit
[interface   ] - Manipulate tunnel interfaces
[show        ] - Show internal data
[tunnels     ] - Manipulate tunnel data
--------- Add vt-GWb
VPN shell:[/] > /interface/add/numbered 10.0.1.12 10.0.0.2 GWb
Interface 'vt-GWb' was added successfully to the system
--------- Add vt-GWc
VPN shell:[/] > /interface/add/numbered 10.0.1.22 10.0.0.3 GWc
Interface 'vt-GWc' was added successfully to the system
---------- Verify configuration
VPN shell:[/] > /show/interface/detailed all
vt-GWb     Type:numbered  MTU:1500
         inet addr:10.0.1.12  P-t-P:10.0.0.2  Mask:255.255.255.255
           Peer:GWb  Peer ID:180.180.1.1  Status:attached

vt-GWc     Type:numbered  MTU:1500
         inet addr:10.0.1.22  P-t-P:10.0.0.3  Mask:255.255.255.255
           Peer:GWc  Peer ID:190.190.1.1  Status:attached

VPN shell:[/] > /quit
[member_GWa2]# ifconfig vt-GWb
vt-GWb    Link encap:IPIP Tunnel  HWaddr
         inet addr:10.0.1.12  P-t-P:10.0.0.2  Mask:255.255.255.255
         UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)

[member_GWa2]# ifconfig vt-GWc
vt-GWc    Link encap:IPIP Tunnel  HWaddr
         inet addr:10.0.1.22  P-t-P:10.0.0.3  Mask:255.255.255.255
         UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)
```

When configuring a VTI in a clustered environment and an interface name is not specified, a name is provided. The default name for a VTI is “vt-[peer gateway name]”. For example, if the peer gateway’s name is Server_2, the default name of the VTI is ‘vt-Server_2’. For peer gateways that have names that are longer than 12 characters, the default interface name is the last five characters plus a 7 byte hash of the peer name calculated to the give the interface a unique name.

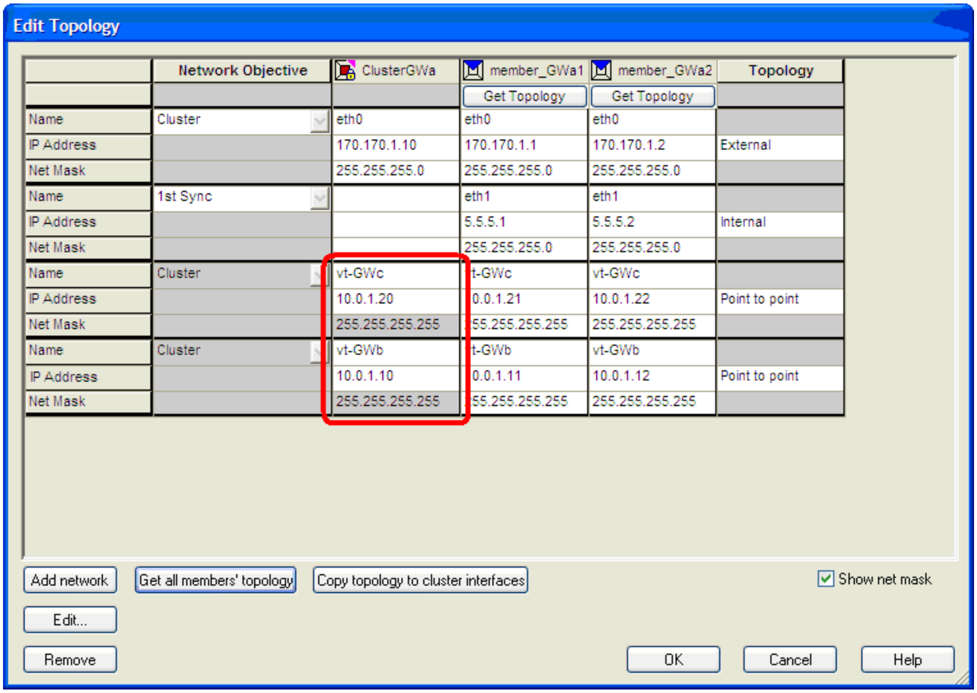
After configuring the VTIs on the cluster members, it is required to configure in the SmartConsole the VIP of these VTIs.

In SmartDashboard:

1. Select **Manage > Network Objects**.
2. Select the Check Point Cluster and right click **Edit**.
3. In **Topology** window, click **Edit Topology**.
4. Click **Get all members’ topology**.

The VTIs now appear in the topology:

Figure 6-4 Edit Topology window



	Network Objective	ClusterGWa	member_GWa1	member_GWa2	Topology
			Get Topology	Get Topology	
Name	Cluster	eth0	eth0	eth0	
IP Address		170.170.1.10	170.170.1.1	170.170.1.2	External
Net Mask		255.255.255.0	255.255.255.0	255.255.255.0	
Name	1st Sync		eth1	eth1	
IP Address			5.5.5.1	5.5.5.2	Internal
Net Mask			255.255.255.0	255.255.255.0	
Name	Cluster	vt-GWc	t-GWc	vt-GWc	
IP Address		10.0.1.20	0.0.1.21	10.0.1.22	Point to point
Net Mask		255.255.255.255	55.255.255.255	255.255.255.255	
Name	Cluster	vt-GWb	t-GWb	vt-GWb	
IP Address		10.0.1.10	0.0.1.11	10.0.1.12	Point to point
Net Mask		255.255.255.255	55.255.255.255	255.255.255.255	

Note that the Edit Topology window, as seen in Figure 6-4, lists the members of a VTI on the same line if the following criteria match:

1. Remote peer name.
2. Remote IP address.

   3. Interface name.

5. Configure the VTI VIP in the **Topology** tab.
6. Click **OK** and install policy.

The sample configurations in Table 6-3 and Table 6-4 use the same gateway names and IP addresses used in Figure 6-3 on page 137.

Table 6-3 Configuring GWb

```
--------- Access the VPN shell Command Line Interface
[GWb]# vpn shell
 ?             - This help
 ..            - Go up one level
 quit          - Quit
[interface   ] - Manipulate tunnel interfaces
[show        ] - Show internal data
[tunnels     ] - Manipulate tunnel data
--------- Add vt-GWa
VPN shell:[/] > /interface/add/numbered 10.0.0.2 10.0.1.10 GWa
Interface 'vt-GWa' was added successfully to the system
--------- Add vt-GWc
VPN shell:[/] > /interface/add/numbered 10.0.0.2 10.0.0.3 GWc
Interface 'vt-GWc' was added successfully to the system
---------- Verify configuration
VPN shell:[/] > /show/interface/detailed all
vt-GWa     Type:numbered  MTU:1500
         inet addr:10.0.0.2  P-t-P:10.0.1.10  Mask:255.255.255.255
           Peer:GWa  Peer ID:170.170.1.10  Status:attached

vt-GWc     Type:numbered  MTU:1500
          inet addr:10.0.0.2  P-t-P:10.0.0.3  Mask:255.255.255.255
           Peer:GWc  Peer ID:190.190.1.1  Status:attached

VPN shell:[/] > /quit
[GWb]# ifconfig vt-GWa
vt-GWa    Link encap:IPIP Tunnel  HWaddr
         inet addr:10.0.0.2  P-t-P:10.0.1.10  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)

[GWb]# ifconfig vt-GWc
vt-GWc    Link encap:IPIP Tunnel  HWaddr
         inet addr:10.0.0.2  P-t-P:10.0.0.3  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)
```

Table 6-4 Configuring GWc

```
--------- Access the VPN shell Command Line Interface
[GWc]# vpn shell
 ?             - This help
 ..            - Go up one level
 quit          - Quit
[interface   ] - Manipulate tunnel interfaces
[show        ] - Show internal data
[tunnels     ] - Manipulate tunnel data
--------- Add vt-GWa
VPN shell:[/] > /interface/add/numbered 10.0.0.3 10.0.1.20 GWa
Interface 'vt-GWa' was added successfully to the system
--------- Add vt-GWb
VPN shell:[/] > /interface/add/numbered 10.0.0.3 10.0.0.2 GWb
Interface 'vt-GWb' was added successfully to the system
---------- Verify configuration
VPN shell:[/] > /show/interface/detailed all
vt-GWa     Type:numbered  MTU:1500
         inet addr:10.0.0.3  P-t-P:10.0.1.20  Mask:255.255.255.255
           Peer:GWa  Peer ID:170.170.1.10  Status:attached

vt-GWb     Type:numbered  MTU:1500
          inet addr:10.0.0.3  P-t-P:10.0.0.2  Mask:255.255.255.255
           Peer:GWb  Peer ID:180.180.1.1  Status:attached

VPN shell:[/] > /quit
[GWc]# ifconfig vt-GWa
vt-GWa    Link encap:IPIP Tunnel  HWaddr
         inet addr:10.0.0.3  P-t-P:10.0.1.20  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)

[GWc]# ifconfig vt-GWb
vt-GWb    Link encap:IPIP Tunnel  HWaddr
          inet addr:10.0.0.3  P-t-P:10.0.0.2  Mask:255.255.255.255
        UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 b)  TX bytes:36 (36.0 b)
```

Enabling Dynamic Routing Protocols on VTIs

Using the example in Figure 6-2 on page 132, Table 6-5 through Table 6-8 illustrate how the OSPF dynamic routing protocol is enabled on VTIs both for single members and for cluster members using SecurePlatform. Note that the network commands for single members and cluster members are not the same.

For more information on advanced routing commands and syntaxes, see the *Check Point Advanced Routing Suite - Command Line Interface* book.

When peering with a Cisco GRE enabled device, a point to point GRE tunnel is required. Use the following command to configure the tunnel interface definition:

```
ip ospf network point-to-point
```

Table 6-5 Dynamic Routing on member_GWA1

```
--------- Launch the Dynamic Routing Module
[member_GWa1]# expert
Enter expert password:

You are in expert mode now.

[Expert@member_GWa1]# cligated
localhost>enable
localhost#configure terminal
--------- Enable OSPF and provide an OSPF router ID
localhost(config)#router ospf 1
localhost(config-router-ospf)#router-id 170.170.1.10
--------- Define interfaces/IP's on which OSPF runs (Use the
cluster IP as defined in topology) and the area ID for the
interface/IP
localhost(config-router-ospf)#network 10.0.1.10 0.0.0.0 area
0.0.0.0
localhost(config-router-ospf)#network 10.0.1.20 0.0.0.0 area
0.0.0.0
--------- Redistribute kernel routes (this is only here as an
example, please see the dynamic routing book for more specific
commands concerning redistribution of routes)
localhost(config-router-ospf)#redistribute kernel
localhost(config-router-ospf)#exit
localhost(config)#exit
-------- Write configuration to disk
localhost#write memory
IU0 999 Configuration written to '/etc/gated.ami'
localhost#quit
```

Table 6-6 Dynamic Routing on member_GWA2

```
--------- Launch the Dynamic Routing Module
[member_GWa2]# expert
Enter expert password:

You are in expert mode now.

[Expert@member_GWa2]# cligated
localhost>enable
localhost#configure terminal
--------- Enable OSPF and provide an OSPF router ID
localhost(config)#router ospf 1
localhost(config-router-ospf)#router-id 170.170.1.10
--------- Define interfaces/IP's on which OSPF runs (Use the
cluster IP as defined in topology) and the area ID for the
interface/IP
localhost(config-router-ospf)#network 10.0.1.10 0.0.0.0 area
0.0.0.0
localhost(config-router-ospf)#network 10.0.1.20 0.0.0.0 area
0.0.0.0
--------- Redistribute kernel routes (this is only here as an
example, please see the dynamic routing book for more specific
commands concerning redistribution of routes)
localhost(config-router-ospf)#redistribute kernel
localhost(config-router-ospf)#exit
localhost(config)#exit
-------- Write configuration to disk
localhost#write memory
IU0 999 Configuration written to '/etc/gated.ami'
localhost#quit
```

Table 6-7 Dynamic Routing on GWb

```
--------- Launch the Dynamic Routing Module
[GWb]# expert
Enter expert password:

You are in expert mode now.

[Expert@GWb]# cligated
localhost>enable
localhost#configure terminal
--------- Enable OSPF and provide an OSPF router ID
localhost(config)#router ospf 1
localhost(config-router-ospf)#router-id 180.180.1.1
--------- Define interfaces/IP's on which OSPF runs (Use the
cluster IP as defined in topology) and the area ID for the
interface/IP
localhost(config-router-ospf)#network 10.0.1.10 0.0.0.0 area
0.0.0.0
localhost(config-router-ospf)#network 10.0.0.3 0.0.0.0 area
0.0.0.0
--------- Redistribute kernel routes (this is only here as an
example, please see the dynamic routing book for more specific
commands concerning redistribution of routes)
localhost(config-router-ospf)#redistribute kernel
localhost(config-router-ospf)#exit
localhost(config)#exit
-------- Write configuration to disk
localhost#write memory
IU0 999 Configuration written to '/etc/gated.ami'
localhost#quit
```

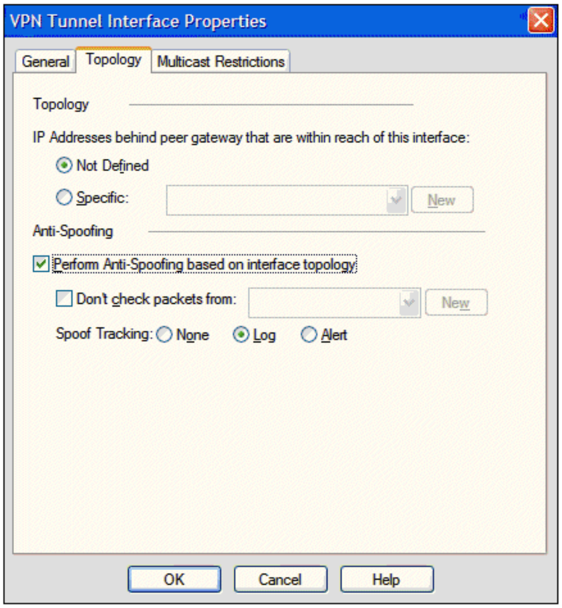
Table 6-8 Dynamic Routing on GWC

```
--------- Launch the Dynamic Routing Module
[GWc]# expert
Enter expert password:

You are in expert mode now.

[Expert@GWc]# cligated
localhost>enable
localhost#configure terminal
--------- Enable OSPF and provide an OSPF router ID
localhost(config)#router ospf 1
localhost(config-router-ospf)#router-id 190.190.1.1
--------- Define interfaces/IP's on which OSPF runs (Use the
cluster IP as defined in topology) and the area ID for the
interface/IP
localhost(config-router-ospf)#network 10.0.1.20 0.0.0.0 area
0.0.0.0
localhost(config-router-ospf)#network 10.0.0.2 0.0.0.0 area
0.0.0.0
--------- Redistribute kernel routes (this is only here as an
example, please see the dynamic routing book for more specific
commands concerning redistribution of routes)
localhost(config-router-ospf)#redistribute kernel
localhost(config-router-ospf)#exit
localhost(config)#exit
-------- Write configuration to disk
localhost#write memory
IU0 999 Configuration written to '/etc/gated.ami'
localhost#quit
```

Configuring Anti-Spoofing on VTIs

In SmartDashboard:

1. Select **Manage > Network Objects**.
2. Select the Check Point gateway and right click **Edit.**
3. In the Properties list, click **Topology.**
4. Click **Get > Interfaces** to read the interface information on the gateway machine.
5. Select an interface click **Edit.**
6. In the Interface Properties window, click **Topology.**



VPN Tunnel Interface Properties

Topology

IP Addresses behind peer gateway that are within reach of this interface:

● Not Defined

○ Specific:

Anti-Spoofing

☑ Perform Anti-Spoofing based on interface topology

□ Don't check packets from:

Spoof Tracking: ○ None ● Log ○ Alert

7. In the **IP Addresses behind peer gateway that are within reach of this interface** section, select:
   - **Not Defined** to accept all traffic.
   - **Specific** to choose a particular network. The IP addresses in this network will be the only addresses accepted by this interface.

8. In the **Perform Anti-Spoofing based on interface topology** section, select **Don't check packets from:** to ensure anti-spoof checks do not take place for addresses from certain internal networks coming into the external interface. Define a network object that represents those internal networks with valid addresses, and from the drop-down list, select that network object.

   Objects selected in the **Don't check packets from:** drop-down menu are disregarded by the anti-spoofing enforcement mechanism.

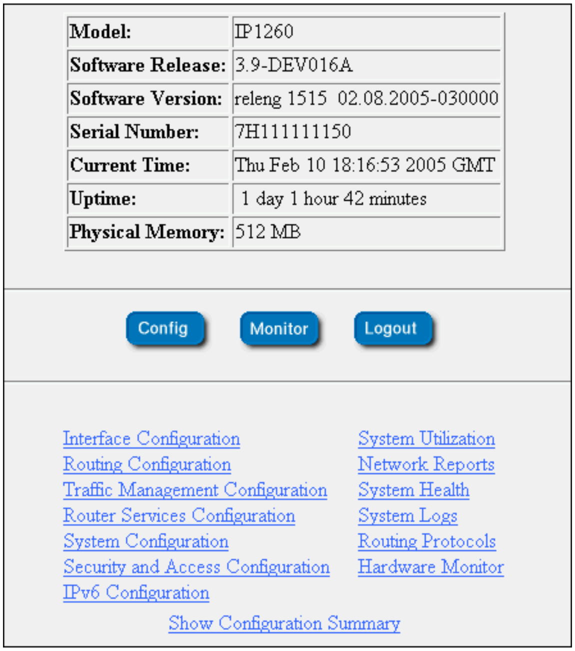
9. Under **Spoof Tracking** select **Log**, and click **OK**.

Configuring a Loopback Interface

When a VTI connects a Nokia machine and a SecurePlatform machine, a loopback interface must be configured and defined in the Topology tab of the gateway.

In Nokia Network Voyager:

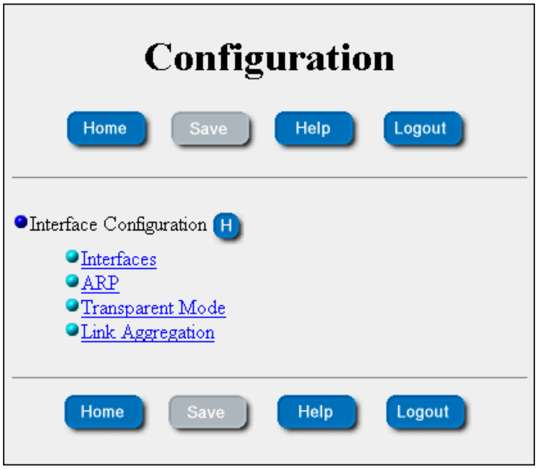
1. Login and the following window appears.



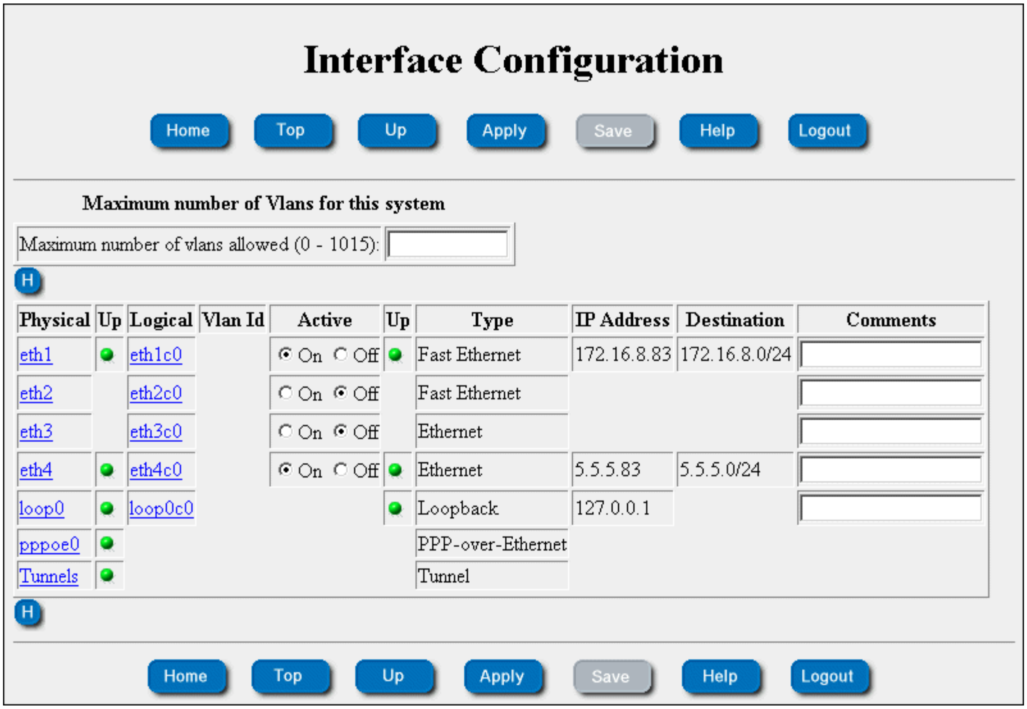
Model:	IP1260
Software Release:	3.9-DEV016A
Software Version:	releng 1515 02.08.2005-030000
Serial Number:	7H111111150
Current Time:	Thu Feb 10 18:16:53 2005 GMT
Uptime:	1 day 1 hour 42 minutes
Physical Memory:	512 MB

2. Click **Interface Configuration**.

3. On the **Configuration** page, click **Interfaces**.



4. On the **Interface Configuration** page, click **loop0**.

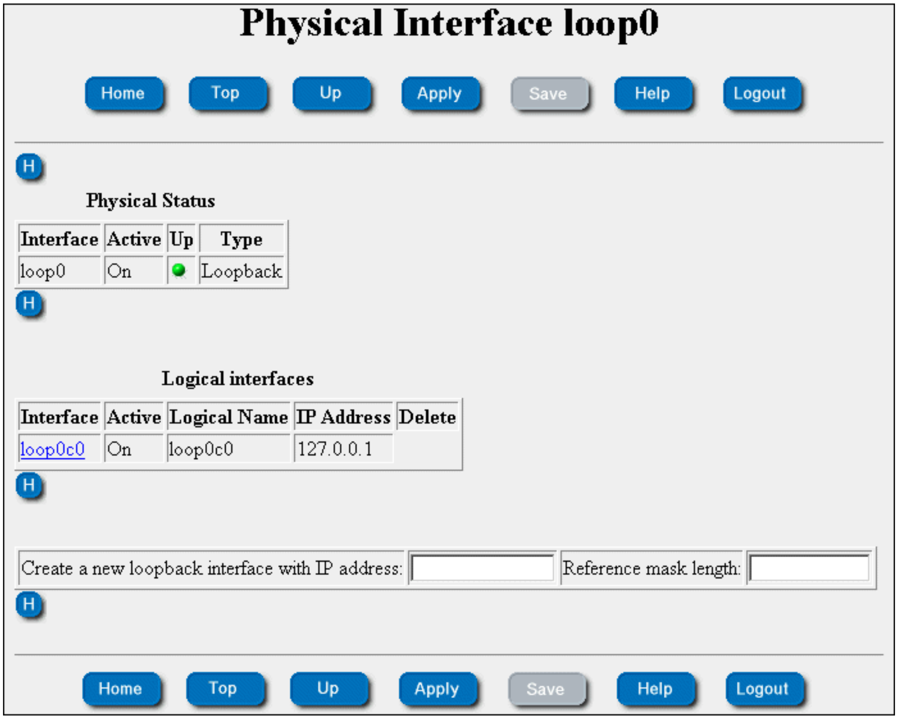


Maximum number of Vlans for this system

Maximum number of vlans allowed (0 - 1015):

Physical	Up	Logical	Vlan Id	Active	Up	Type	IP Address	Destination	Comments
eth1	●	eth1c0		● On ○ Off	●	Fast Ethernet	172.16.8.83	172.16.8.0/24	
eth2		eth2c0		○ On ● Off		Fast Ethernet			
eth3		eth3c0		○ On ● Off		Ethernet			
eth4	●	eth4c0		● On ○ Off	●	Ethernet	5.5.5.83	5.5.5.0/24	
loop0	●	loop0c0			●	Loopback	127.0.0.1		
pppoe0	●					PPP-over-Ethernet			
Tunnels	●					Tunnel			

5. On the **Physical Interface loop0** page, enter an IP address in the **Create a new loopback interface with IP address** field and the value ‘30’ in the **Reference mask length** field.



Physical Interface loop0

Physical Status

Interface	Active	Up	Type
loop0	On	●	Loopback

Logical interfaces

Interface	Active	Logical Name	IP Address	Delete
loop0c0	On	loop0c0	127.0.0.1	

Create a new loopback interface with IP address:		Reference mask length:	

6. Click **Apply**.

   The **Physical Interface loop0** page refreshes and displays the newly configured loopback interface.

7. Click **Save**.

Configuring Unnumbered VTIs

The Nokia IPSO platform supports unnumbered VTIs in a VRRP HA configuration, active-passive mode only.

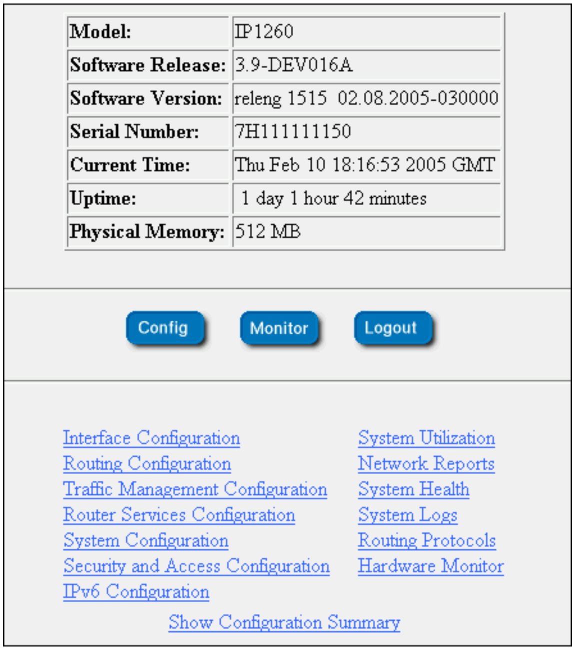
If the VPN Tunnel Interface is unnumbered, local and remote IP addresses are not configured. This interface is associated with a proxy interface from which the virtual interface inherits an IP address. Traffic initiated by the gateway and routed through the virtual interface will have the physical interfaces's IP Address as the source IP.

Working with unnumbered interfaces eliminates the need to assign two IP addresses per interface (the local IP, and the remote IP Address), and the need to synchronize this information among the peers.

Unnumbered interfaces are only supported on the Nokia IPSO 3.9 platform.

In Nokia Network Voyager:

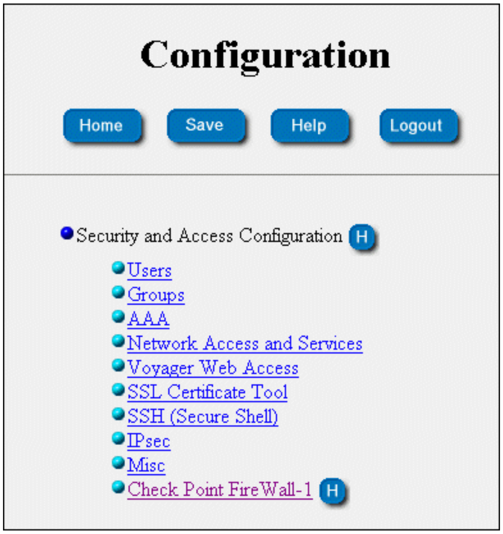
1. Login and the following window appears.



Model:	IP1260
Software Release:	3.9-DEV016A
Software Version:	releng 1515 02.08.2005-030000
Serial Number:	7H111111150
Current Time:	Thu Feb 10 18:16:53 2005 GMT
Uptime:	1 day 1 hour 42 minutes
Physical Memory:	512 MB

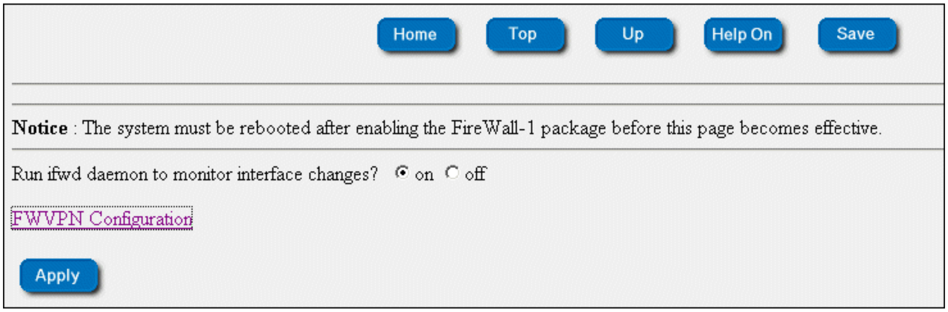
2. Click **Config**.

3. On the **Configuration** page, click **Check Point Firewall-1**.



- Security and Access Configuration
  - Users
  - Groups
  - AAA
  - Network Access and Services
  - Voyager Web Access
  - SSL Certificate Tool
  - SSH (Secure Shell)
  - IPsec
  - Misc
  - Check Point FireWall-1

4. On the next page, click **FWVPN Configuration**.



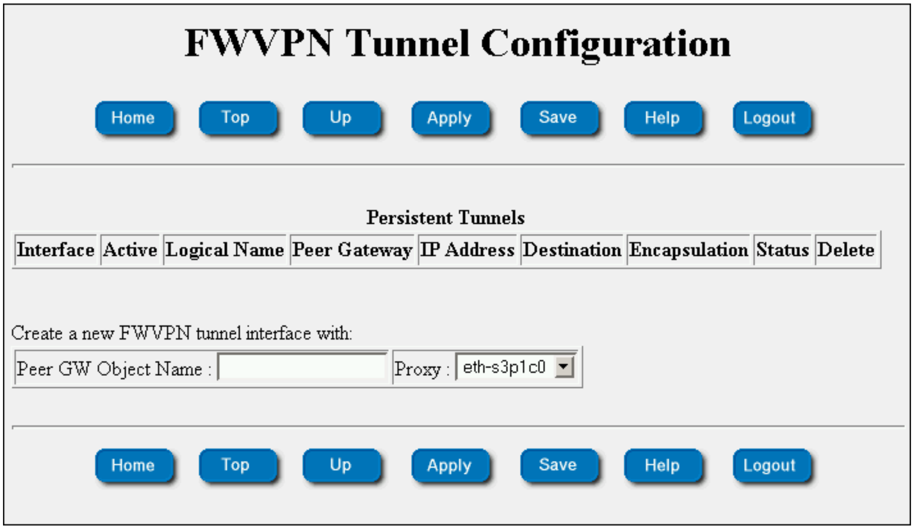
Notice : The system must be rebooted after enabling the FireWall-1 package before this page becomes effective.

Run ifwd daemon to monitor interface changes? ● on ○ off

FWVPN Configuration

5. On the **FWVPN Tunnel Configuration** page, enter the name of the gateway you want to connect to in the **Peer GW Object Name** field.

Select a proxy interface from the **Proxy** drop down menu.



FWVPN Tunnel Configuration

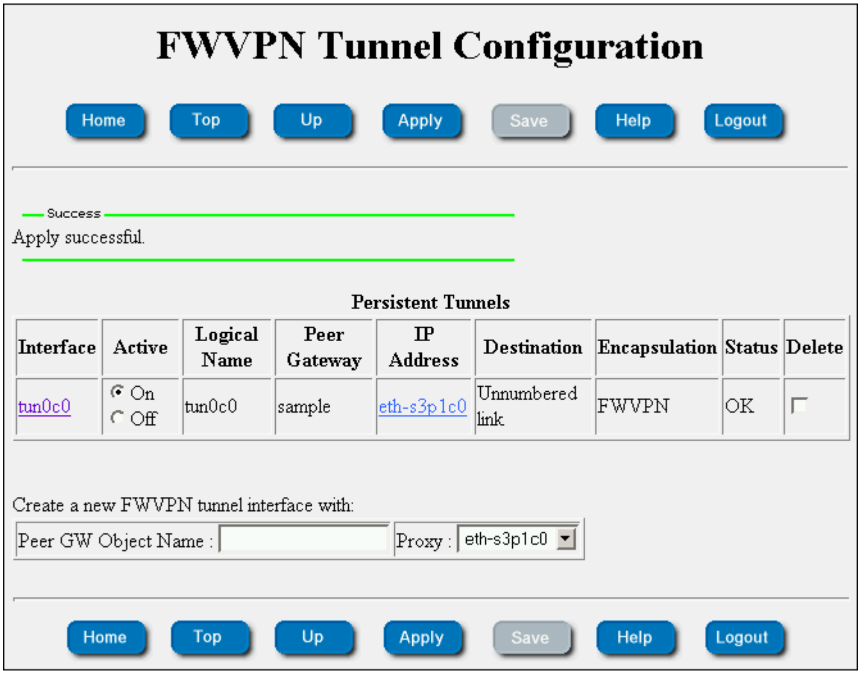
Persistent Tunnels

Interface	Active	Logical Name	Peer Gateway	IP Address	Destination	Encapsulation	Status	Delete

Create a new FWVPN tunnel interface with:

Peer GW Object Name :	Proxy : eth-s3p1c0

6. Click **Apply.**
7. The new interface is now listed on the **FWVPN Tunnel Configuration** page.



FWVPN Tunnel Configuration

Success

Apply successful.

Persistent Tunnels

Interface	Active	Logical Name	Peer Gateway	IP Address	Destination	Encapsulation	Status	Delete
tun0c0	● On ○ Off	tun0c0	sample	eth-s3p1c0	Unnumbered link	FWVPN	OK	□

Create a new FWVPN tunnel interface with:

Peer GW Object Name :	Proxy : eth-s3p1c0

To enable dynamic routing protocols on the Nokia IPSO platform using Nokia Network Voyager, see the *Nokia Network Voyager Reference Guide*.

Routing Multicast Packets Through VPN Tunnels

Multicast is used to transmit a single message to a select group of recipients. IP Multicasting applications send one copy of each datagram (IP packet) and address it to a group of computers that want to receive it. This technique addresses datagrams to a group of receivers (at the multicast address) rather than to a single receiver (at a unicast address). The network is responsible for forwarding the datagrams to only those networks that need to receive them. For more information on Multicasting, see *"Multicast Access Control"* in the *Firewall Administration Guide*.

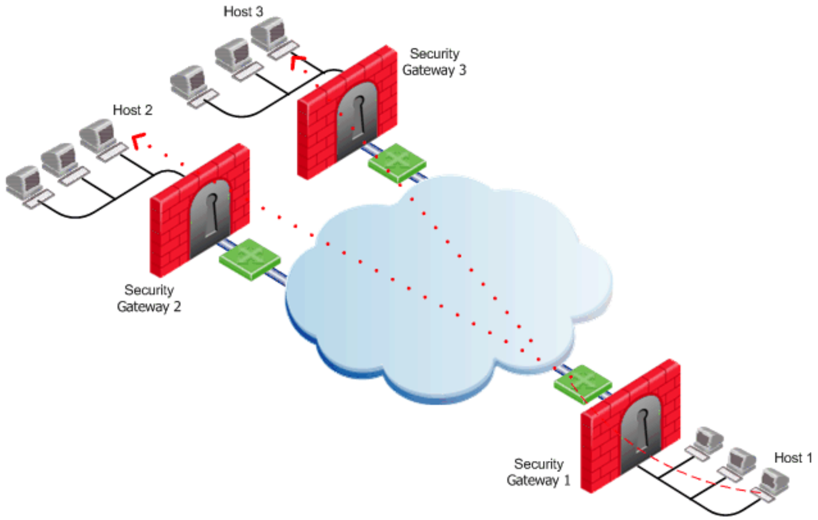
Multicast traffic can be encrypted and forwarded across VPN tunnels that were configured using VPN tunnel interfaces (virtual interfaces associated with the same physical interface). All participant gateways, both on the sending and receiving ends, must have a virtual interface for each VPN tunnel and a multicast routing protocol must be enabled on all participant gateways.

For more information on virtual interfaces, see "Configuring a Virtual Interface Using the VPN Shell" on page 730.

In Figure 6-5:

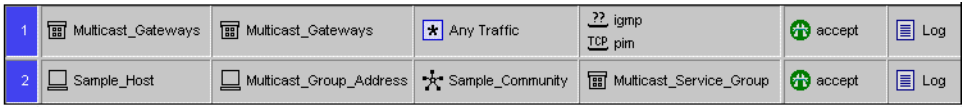
- Gateway 1 has a virtual interface configured for the VPN tunnel linked with gateway 2 and another virtual interface for the VPN tunnel linked with gateway 3.
- Host 1 behind gateway 1 initiates a multicast session destined to the multicast group address which consists of Host 2 behind gateway 2 and to Host 3 behind gateway 3.

Figure 6-5 MultiCasting



To enable multicast service on a security gateway functioning as a rendezvous point, add a rule to the security policy of that gateway to allow only the specific multicast service to be accepted unencrypted, and to accept all other services only through the community. Corresponding access rules enabling multicast protocols and services should be created on all participating gateways. For example:

Figure 6-6 Sample Rules



1	Multicast_Gateways	Multicast_Gateways	Any Traffic	igmp pim	accept	Log
2	Sample_Host	Multicast_Group_Address	Sample_Community	Multicast_Service_Group	accept	Log

Chapter 7
Tunnel Management

In This Chapter

Overview

A Virtual Private Network (VPN) provides a secure connection, typically over the Internet. VPNs accomplish this by creating an encrypted tunnel that provides the same security available as in a private network. This allows workers who are in the field or working at home to securely connect to a remote corporate server and also allows companies to securely connect to branch offices and other companies over the Internet. The VPN tunnel guarantees:

- authenticity, by using standard authentication methods.
- privacy, by encrypting data.
- integrity, by using standard integrity assurance methods.

Types of tunnels and the number of tunnels can be managed with the following features:

- *Permanent Tunnels* - This feature keeps VPN tunnels active allowing real-time monitoring capabilities.
- *VPN Tunnel Sharing* - This feature provides greater interoperability and scalability between gateways. It also controls the number of VPN tunnels created between peer gateways.

The status of all VPN tunnels can be viewed in SmartView Monitor. For more information on monitoring see the *Monitoring Tunnels* chapter in the *SmartView Monitor User Guide*.

Permanent Tunnels

As companies have become more dependent on VPNs for communication to other sites, uninterrupted connectivity has become more crucial than ever before. Therefore it is essential to make sure that the VPN tunnels are kept up and running. Permanent Tunnels are constantly kept active and as a result, make it easier to recognize malfunctions and connectivity problems. Administrators can monitor the two sides of a VPN tunnel and identify problems without delay.

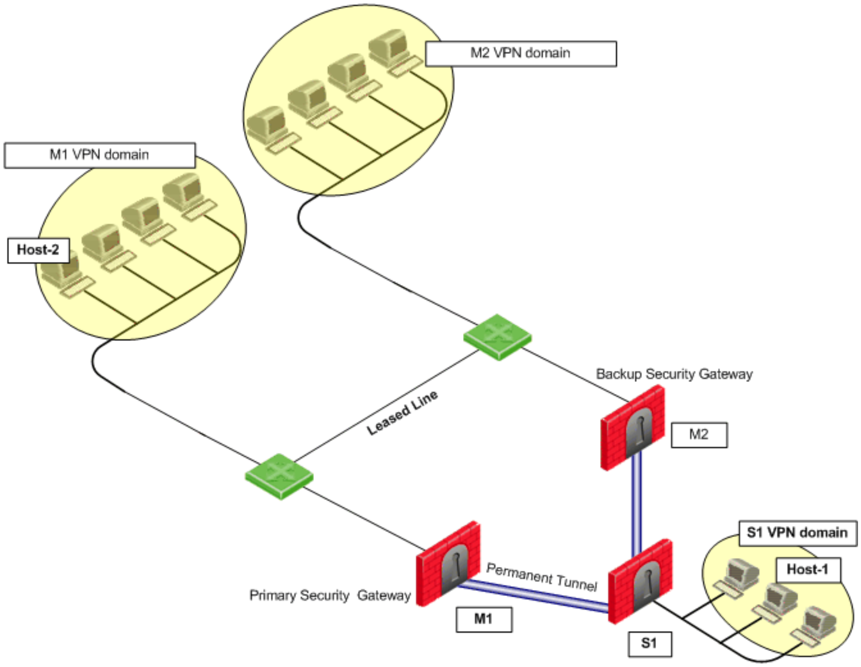
Each VPN tunnel in the community may be set to be a Permanent Tunnel. Since Permanent Tunnels are constantly monitored, if the VPN tunnel is down, then a log, alert, or user defined action, can be issued. A VPN tunnel is monitored by periodically sending “tunnel test” packets. As long as responses to the packets are received the VPN tunnel is considered “up.” If no response is received within a given time period, the VPN tunnel is considered “down.” Permanent Tunnels can only be established between Check Point gateways. The configuration of Permanent Tunnels takes place on the community level and:

- Can be specified for an entire community. This option sets every VPN tunnel in the community as permanent.
- Can be specified for a specific gateway. Use this option to configure specific gateways to have permanent tunnels.
- Can be specified for a single VPN tunnel. This feature allows configuring specific tunnels between specific gateways as permanent.

Permanent Tunnels in a MEP Environment

In a *Multiple Entry Point* (MEP) environment, VPN tunnels that are active are rerouted from the predefined primary gateway to the backup gateway if the primary gateway becomes unavailable. When a Permanent Tunnel is configured between gateways in a MEPed environment where RIM is enabled, the satellite gateways see the center gateways as "unified." As a result, the connection will not fail but will fail over to another center gateway on a newly created permanent tunnel. For more information on MEP see "Multiple Entry Point VPNs" on page 229.

Figure 7-1 Permanent Tunnel in MEP environment



In this scenario:

- Host 1, residing behind gateway S1, is communicating through a Permanent Tunnel with Host 2, residing behind gateway M1.

- M1 and M2 are in a MEPed environment.
- M1 and M2 are in a MEPed environment with Route Injection Mechanism (RIM) enabled.
- M1 is the Primary gateway and M2 is the Backup gateway.

In this case, should gateway M1 become unavailable, the connection would continue though a newly created permanent tunnel between S1 and M2.

Tunnel Testing for Permanent Tunnels

Tunnel test is a proprietary Check Point protocol that is used to test if VPN tunnels are active. A packet has an arbitrary length, with only the first byte containing meaningful data - this is the 'type' field.

The 'type' field can take any of the following values:

1 - Test

2 - Reply

3 - Connect

4 - Connected

Tunnel testing requires two gateways - one configured as a pinger and one as a responder. A pinger gateway uses the VPN daemon to send encrypted “tunnel testing” packets to gateways configured to listen for them. A responder gateway is configured to listen on port 18234 for the special tunnel testing packets.

The pinger sends type 1 or 3. The responder sends a packet of identical length with type 2 or 4 respectively. During the 'connect' phase, “tunnel test” is used in two ways:

1. A 'connect' message is sent to the gateway. Receipt of a 'connected' message is the indication that the connection succeeded. The 'connect' messages are retransmitted for up to 10 seconds after the IKE negotiation is over if no response is received.
2. A series of 'test' messages with various lengths is sent so as to discover the PMTU (Path Maximum Transmission Unit) of the connection. This may also take up to 10 seconds. This test is executed to ensure that TCP packets that are too large are not sent. TCP packets that are too large will be fragmented and slow down performance.

Gateways with version R54 and forward can be either a pinger or responder. In a MEP environment, center gateways can only be responders.

Gateways with Embedded NG 5.0 and forward can be pingers or responders. Older versions of this software can only be responders.

3rd party gateways cannot be a pinger or responder.

VPN Tunnel Sharing

Since various vendors implement IPSec tunnels using a number of different methods, administrators need to cope with different means of implementation of the IPSec framework.

VPN Tunnel Sharing provides interoperability and scalability by controlling the number of VPN tunnels created between peer gateways. There are three available settings:

- **One VPN tunnel per each pair of hosts**
- **One VPN tunnel per subnet pair**
- **One VPN tunnel per Gateway pair**

Configuring Tunnel Features

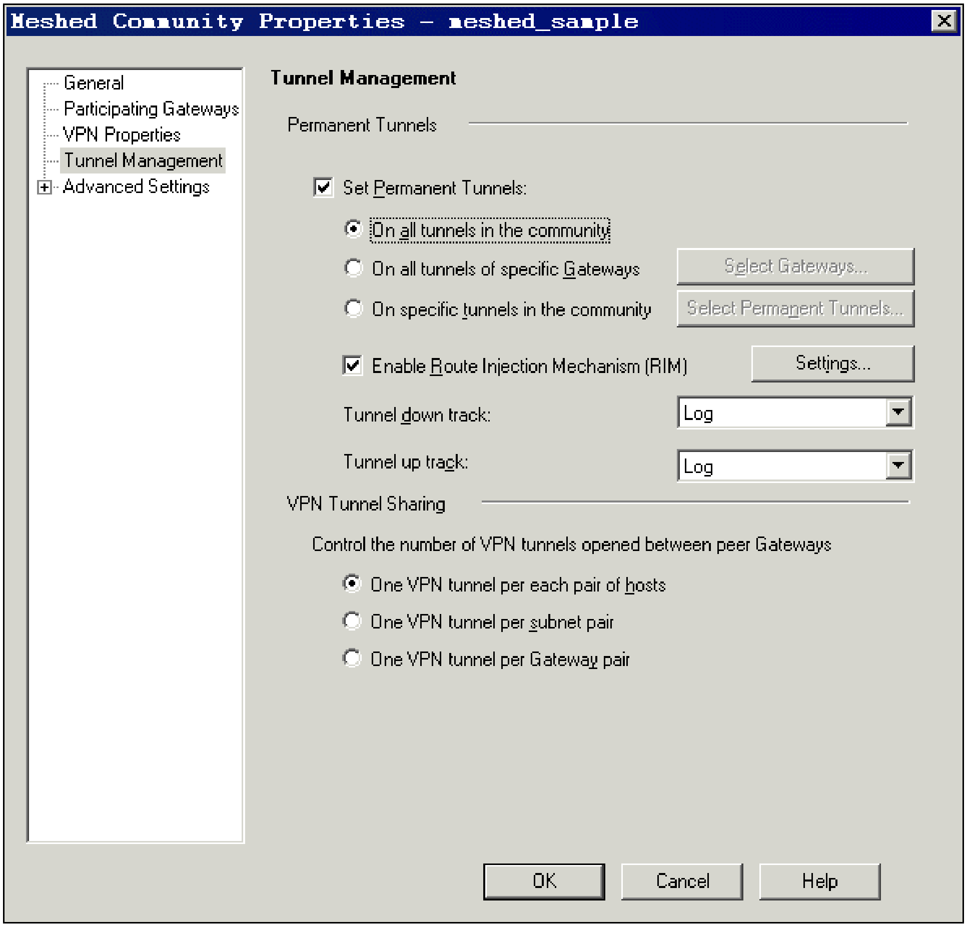
In This Section

To configure Tunnel Management options, proceed as follows:

1. In SmartDashboard, click **Manage** > **VPN Communities**. The **VPN Communities** window will appear.
2. Select the community (star or meshed) to be configured and click **Edit...**
3. Click **Tunnel Management.**

   The **Tunnel Management** window is displayed.

Figure 7-2 Meshed Communities Properties - Tunnel Management page



Meshed Community Properties – meshed_sample

Tunnel Management

Permanent Tunnels

☑ Set Permanent Tunnels:

● On all tunnels in the community

○ On all tunnels of specific Gateways

○ On specific tunnels in the community

☑ Enable Route Injection Mechanism (RIM)

Tunnel down track: Log

Tunnel up track: Log

VPN Tunnel Sharing

Control the number of VPN tunnels opened between peer Gateways

● One VPN tunnel per each pair of hosts

○ One VPN tunnel per subnet pair

○ One VPN tunnel per Gateway pair

- for Permanent Tunnels see “Permanent Tunnels” on page 170.
- for Tracking see “Tracking Options” on page 174.
- for VPN Tunnel Sharing see “VPN Tunnel Sharing” on page 174.

Permanent Tunnels

In the **Community Properties** window on the **Tunnel Management** page, select **Set Permanent Tunnels** and the following Permanent Tunnel modes are then made available:

- **On all tunnels in the community**
- **On all tunnels of specific gateways**
- **On specific tunnels in the community**

To configure all tunnels as permanent, select **On all tunnels in the community**. Deselect this option to terminate all Permanent Tunnels in the community.

To configure **On all tunnels of specific gateways:**

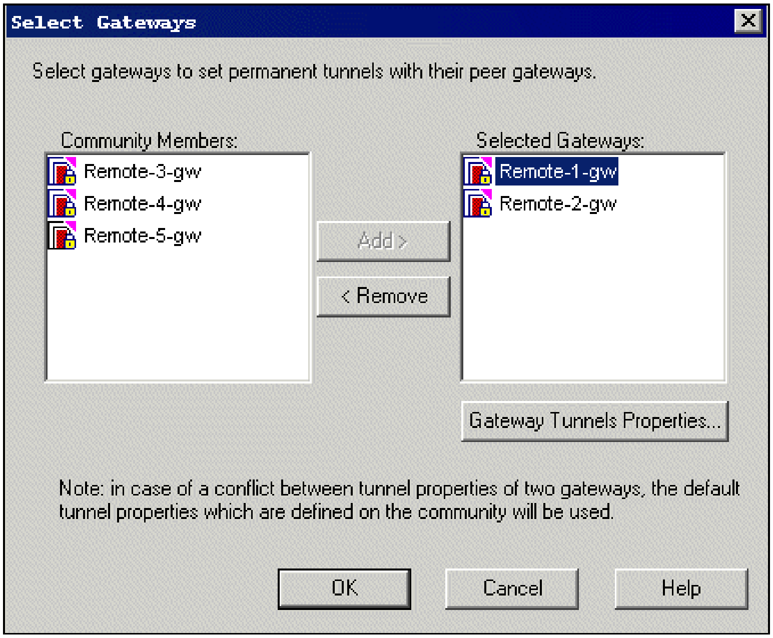
1. Select **On all tunnels of specific gateways** and click the **Select Gateways...** button.

   The **Select Gateways** window is displayed.

   The example given in Figure 7-3 shows **Remote -1- gw** and **Remote -2- gw** as the only two gateways selected. As a result, all VPN tunnels connected with **Remote -1- gw** or **Remote -2- gw** will be permanent.

   To terminate Permanent Tunnels connected to a specific gateway, highlight the gateway and click **Remove**.

Figure 7-3 Selected Gateways window



Select Gateways

Select gateways to set permanent tunnels with their peer gateways.

Community Members:

- Remote-3-gw
- Remote-4-gw
- Remote-5-gw

Selected Gateways:

- Remote-1-gw
- Remote-2-gw

Note: in case of a conflict between tunnel properties of two gateways, the default tunnel properties which are defined on the community will be used.

2. To configure the Tracking options for a specific gateway, highlight a gateway and click on **Gateway Tunnels Properties**.

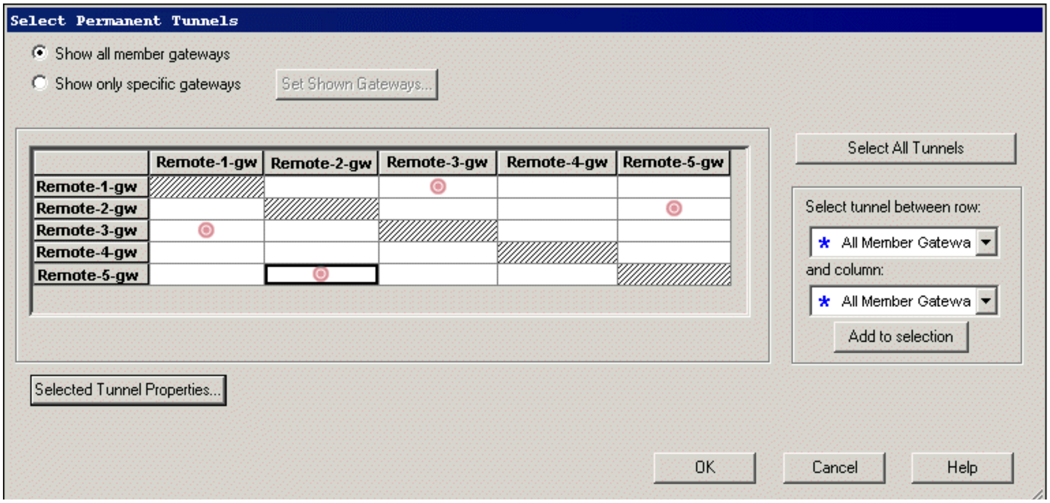
To configure **On specific tunnels in the community:**

1. Select **On specific tunnels in the community** and click the **Select Permanent Tunnels...** button.

   The **Select Permanent Tunnels...** window is displayed.

   In the example given in Figure 7-4, a permanent tunnel was configured between **Remote -1- gw** and **Remote -3- gw** and another permanent tunnel was configured between **Remote -2- gw** and **Remote -5- gw**.

Figure 7-4 Select Permanent Tunnels window



● Show all member gateways

○ Show only specific gateways

	Remote-1-gw	Remote-2-gw	Remote-3-gw	Remote-4-gw	Remote-5-gw
Remote-1-gw	▨		●		
Remote-2-gw		▨			●
Remote-3-gw	●		▨		
Remote-4-gw				▨	
Remote-5-gw		●			▨

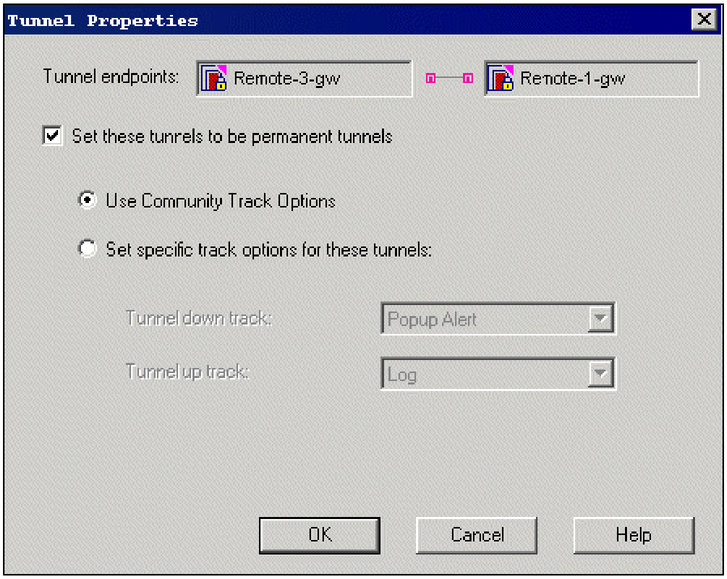
Select tunnel between row: * All Member Gatewa

and column: * All Member Gatewa

2. Click in the cell that intersects the gateways where a permanent tunnel is required.

3. Click **Selected Tunnel Properties** and the **Tunnel Properties** window is displayed.

Figure 7-5 Tunnel Properties window



Tunnel endpoints: Remote-3-gw — Remote-1-gw

☑ Set these tunnels to be permanent tunnels

● Use Community Track Options

○ Set specific track options for these tunnels:

Tunnel down track: Popup Alert

Tunnel up track: Log

4. Select **Set these tunnels to be permanent tunnels**.

To terminate the Permanent Tunnel between these two gateways, deselect **Set these tunnels to be permanent tunnels**.

5. Click **OK**.

Advanced Permanent Tunnel Configuration

In SmartDashboard:

1. Click **Policy > Global Properties**.

   The **Global Properties** window is displayed.

2. Select **SmartDashboard Customization** from the properties list.
3. In the **Advanced Configuration** section, click **Configure**.

   The **Advanced configuration** window is displayed.

4. Click **VPN Advanced Properties > Tunnel Management** to view the five attributes that may be configured to customize the amount of tunnel tests sent and the intervals in which they are sent:

   - **life_sign_timeout** - Designate the amount of time the tunnel test runs without a response before the peer host is declared ‘down.’
   - **life_sign_transmitter_interval** - Set the time between tunnel tests.
   - **life_sign_retransmissions_count** - When a tunnel test does not receive a reply, another test is resent to confirm that the peer is ‘down.’ The Life Sign Retransmission Count is set to how many times the tunnel test is resent without receiving a response.
   - **life_sign_retransmissions_interval** - Set the time between the tunnel tests that are resent after it does not receive a response from the peer.
   - **cluster_status_polling_interval** - (Relevant for HA Clusters only) - Set the time between tunnel tests between a primary gateway and a backup gateway. The tunnel test is sent by the backup gateway. When there is no reply, the backup gateway will become active.

Tracking Options

Several types of alerts can be configured to keep administrators up to date on the status of the VPN tunnels. The Tracking settings can be configured on the **Tunnel Management** page of the **Community Properties** screen for all VPN tunnels or they can be set individually when configuring the permanent tunnels themselves. The different options are **Log**, **Popup Alert**, **Mail Alert**, **SNMP Trap Alert**, and **User Defined Alert**. Choosing one of these alert types will enable immediate identification of the problem and the ability to respond to these issues more effectively.

Terminating Permanent Tunnels

Once a Permanent Tunnel is no longer required, the tunnel can be shut down. Permanent Tunnels are shut down by deselecting the configuration options to make them active and re-installing the policy.

VPN Tunnel Sharing

For a VPN community, the configuration is set on the **Tunnel Management** page of the **Community Properties** window.

For a specific gateway, the configuration is set on the **VPN Advanced** page of the gateway's properties window.

VPN Tunnel Sharing provides greater interoperability and scalability by controlling the number of VPN tunnels created between peer gateways. Configuration of VPN Tunnel Sharing can be set on both the VPN community and gateway object.

- **One VPN Tunnel per each pair of hosts** - A VPN tunnel is created for every session initiated between every pair of hosts.
- **One VPN Tunnel per subnet pair**- Once a VPN tunnel has been opened between two subnets, subsequent sessions between the same subnets will share the same VPN tunnel. This is the default setting and is compliant with the IPSec industry standard.
- **One VPN Tunnel per Gateway pair**- One VPN tunnel is created between peer gateways and shared by all hosts behind each peer gateway.

In case of a conflict between the tunnel properties of a VPN community and a gateway object that is a member of that same community, the "stricter" setting is followed. For example, a gateway that was set to **One VPN Tunnel per each pair of hosts** and a community that was set to **One VPN Tunnel per subnet pair**, would follow **One VPN Tunnel per each pair of hosts**.

Monitoring Tunnels

The status of all VPN tunnels can be viewed in SmartView Monitor. For more information on monitoring see the *Monitoring Tunnels* chapter in the *SmartView Monitor User Guide*.

Chapter 8

Route Injection Mechanism

In This Chapter

Overview

Route Injection Mechanism (RIM) enables a security gateway to use dynamic routing protocols to propagate the encryption domain of a VPN peer gateway to the internal network and then initiate back connections. When a VPN tunnel is created, RIM updates the local routing table of the security gateway to include the encryption domain of the VPN peer.

RIM can only be enabled when permanent tunnels are configured for the community. Permanent tunnels are kept alive by tunnel test packets. When a gateway fails to reply, the tunnel will be considered 'down.' As a result, RIM will delete the route to the failed link from the local routing table, which triggers neighboring dynamic routing enabled devices to update their routing information accordingly. This will result in a redirection of all traffic destined to travel across the VPN tunnel, to a pre-defined alternative path.

There are two possible methods to configure RIM:

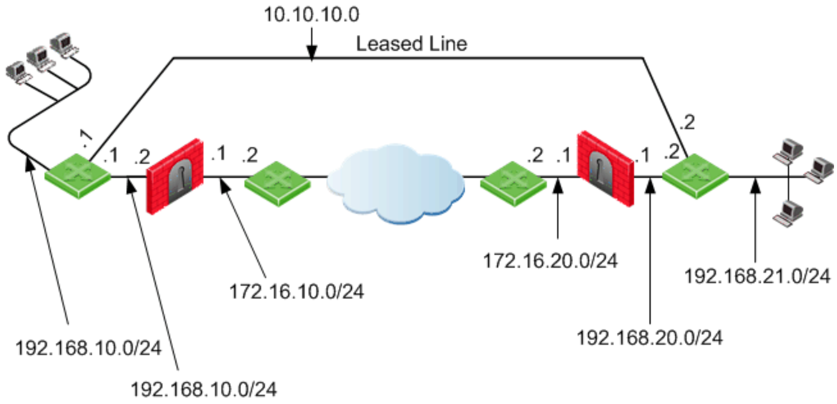
- Automatic RIM - RIM automatically injects the route to the encryption domain of the peer gateways.
- Custom Script - Specify tasks for RIM to perform according to specific needs.

Route injection can be integrated with MEP functionality (which route return packets back through the same MEP gateway). For more information on MEP, see "Multiple Entry Point VPNs" on page 229.

Automatic RIM

Automatic RIM can be enabled using the GUI when the operating system on the gateway is SecurePlatform, IPSO or Linux. Although a custom script can be used on these systems, no custom-written scripts are required.

Figure 8-1 Automatic RIM



In this scenario:

- Gateways 1 and 2 are both RIM and have a dynamic routing protocol enabled.
- R1 and R4 are enabled routers.
- When a VPN tunnel is created, RIM updates the local routing tables of gateway 1 and gateway 2 to include the encryption domain of the other gateway.
- Should the VPN tunnel become unavailable, traffic is redirected to the leased line.

The routing tables for the gateways and routers read as follows. Entries in bold represent routes injected into the gateways local routing tables by RIM:

Gateway 1:

Network Destination	Netmask	Gateway	Metric
0.0.0.0	0.0.0.0	172.16.10.2	1
192.168.21.0	**255.255.255.0**	**172.16.10.2**	**1**
192.168.11.0	255.255.255.0	192.168.10.1	1

Gateway 2:

Network Destination	Netmask	Gateway	Metric
0.0.0.0	0.0.0.0	172.16.20.2	1
192.168.11.0	**255.255.255.0**	**172.16.20.2**	**1**
192.168.21.0	255.255.255.0	192.168.20.1	1

R1 (behind gateway 1):

Network Destination	Netmask	Gateway	Metric
0.0.0.0	0.0.0.0	192.168.10.2	1
192.168.21.0	255.255.255.0	192.168.10.2	1
192.168.21.0	255.255.255.0	10.10.10.2	2

R4 (behind gateway 2):

Network Destination	Netmask	Gateway	Metric
0.0.0.0	0.0.0.0	192.168.20.2	1
192.168.11.0	255.255.255.0	192.168.20.2	1
192.168.11.0	255.255.255.0	10.10.10.1	2

Custom Scripts

Custom scripts can be run on any gateway in the community. These scripts are executed whenever a tunnel changes its state, i.e. goes “up” or “down.” Such an event, for example, can be the trigger that initiates a dial-up connection.

A script template custom_rim (with a .sh or .bat extension depending on the operating system) is provided in the $FWDIR/Scripts directory. The basic script (for SecurePlatform, IPSO, or Linux only) is shown in Figure 8-2:

Figure 8-2 Sample customized script for SecurePlatform, IPSO, or Linux

```
#!/bin/sh

# This script is invoked each time a tunnel is configured with the
RIM option
# and the tunnel changed state.
#
# You may add your custom commands to be invoked here.

# Parameters read from command line.
RIM_PEER_GATEWAY=$1
RIM_NEW_STATE=$2
RIM_HA_STATE=$3
RIM_FIRST_TIME=$4
RIM_PEER_ENC_NET=$5

case "${RIM_NEW_STATE}" in
  up)
    # Place your action for tunnels that came up
    ;;
  down)
    # Place your action for tunnel that went down
    ;;
esac
```

For Windows platforms, the script takes the form of a batch file:

Figure 8-3 Sample customized script for Windows

```
@echo off

rem . This script is invoked each time a tunnel is configured with
the RIM option
rem . and the tunnel changed state.
rem .
rem . You may add your custom commands to be invoked here.

rem . Parameters read from command line.
set RIM_PEER_GATEWAY=%1
set RIM_NEW_STATE=%2
set RIM_HA_STATE=%3
set RIM_FIRST_TIME=%4
set RIM_PEER_ENC_NET=%5

goto RIM_%RIM_NEW_STATE%

:RIM_up
rem . Place your action for tunnels that came up
goto end

:RIM_down
rem . Place your action for tunnel that went down
goto end

:end
```

Where:

- RIM_PEER_GATEWAY: Peer gateway
- RIM_NEW_STATE: Change in the gateways's state, i.e. up or down.
- RIM_HA_STATE: State of a single gateway in a cluster (i.e., standby or active).
- RIM_FIRST_TIME: The script is executed separately for each network within the peers encryption domain. Although the script might be executed multiple times on a peer, this parameter will only be transferred to the script with the value of '1' the first time the script runs on the peer. The value '1' indicates that this is the first time this script is being executed. The next time the script is executed, it is transferred with the value of '0' and the parameter is disregarded. For example, you may send an email alert to the system administrator the moment a tunnel goes down.
- RIM_PEER_ENC_NET: VPN domain of the VPN peer.

tnlmon.conf File

In R54 and R55, RIM was configured using the `tnlmon.conf` file. If your RIM settings are already configured using the `tnlmon.conf` file, there is no need to reconfigure RIM using SmartDashboard. RIM is supported in a mixed community where there are gateways configured using the GUI and other gateways using the `tnlmon.conf` file. RIM is not supported when communicating with 3rd party gateways. However, if RIM is configured in the `tnlmon.conf` file, `these` settings will take precedence over any RIM settings in the GUI.

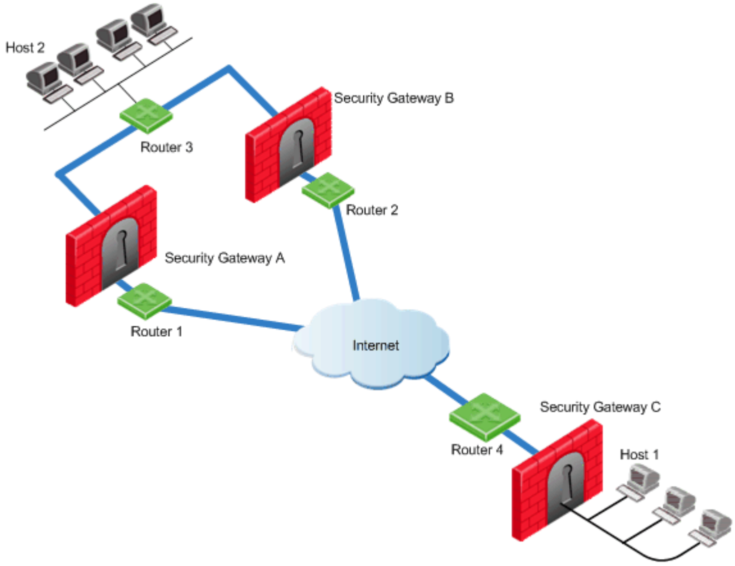
To configure RIM using the `tnlmon.conf` file, refer to the R54 and R55 User Guides.

Injecting Peer Gateway Interfaces

The RIM_inject_peer_interfaces flag is used to inject into the routing tables the peer gateway's IP addresses in addition to the networks behind the gateway.

For example, after a VPN tunnel is created, RIM injects into the local routing tables of both gateways, the encryption domain of the peer gateway. However, when RIM enabled gateways communicate with a gateway that has Hide NAT enabled, the peer's interfaces need to be injected as well.

Figure 8-4 Gateway with Hide NAT



In this scenario:

- Secuirity gateways A and B are both RIM enabled and gateway C has Hide NAT enabled on the external interface ("hiding" all the IP addresses behind it).
- Host 1, behind gateway C, initiates a VPN tunnel with Host 2, through gateway A.

In Figure 8-4, Router 3 contains the routes to all the hosts behind gateway C. Router 3 however, does not have the Hide NAT IP address of gateway C and as a result, cannot properly route packets back to host 1.

This solution for routing the packets back properly is twofold:

1. Select the flag RIM_inject_peer_interfaces in the **Global Properties** page. This flag will inject router 3 with all of the IP addresses of gateway C including the Hide NAT address.

2. Configure the router not to propagate the information injected to other gateways. If the router is not configured properly, using the example in Figure 8-4, could result in gateway B routing traffic to gateway C through gateway A.

Configuring RIM

In This Section

Configuring RIM in a Star Community:

1. Open the **Star Community properties > Tunnel Management** page.

In the **Permanent Tunnels** section, select **Set Permanent Tunnels.** The following Permanent Tunnel modes are then made available:

- **On all tunnels in the community**
- **On all tunnels of specific gateways**
- **On specific tunnels in the community**

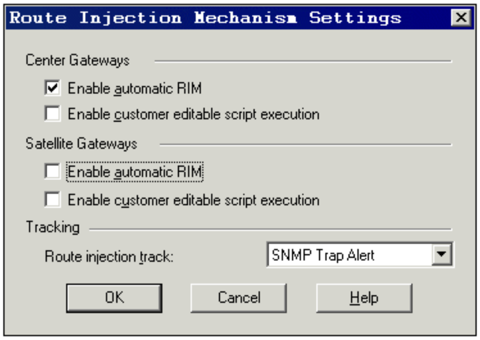
For more information on these options, see "Permanent Tunnels" on page 170.

When choosing tunnels, keep in mind that RIM can only be enabled on tunnels that have been configured to be permanent. **On all tunnels in the community** must be selected if MEP is enabled on the community.

To configure permanent tunnels, see "Configuring Tunnel Features" on page 168.

2. Select **Enable Route Injection Mechanism (RIM).**
3. Click **Settings...**

The **Route Injection Mechanism** Settings window opens



Route Injection Mechanism Settings

Center Gateways

☑ Enable automatic RIM

□ Enable customer editable script execution

Satellite Gateways

□ Enable automatic RIM

□ Enable customer editable script execution

Tracking

Route injection track: SNMP Trap Alert

Decide if:

- RIM should run automatically on the central or satellite gateways (SecurePlatform, IPSO or Linux only).
- A customized script should be run on central or satellite gateways whenever a tunnel changes its states (goes up or down).

For tracking options, see "Tracking Options" on page 188.

4. If a customized script is run, edit `custom_rim` (`.sh` or `.bat`) script in the `$FWDIR/Scripts` directory on each of the gateways.

Configuring RIM in a Meshed Community:

1. Open the **Meshed Community properties > Tunnel Management** page.

In the **Permanent Tunnels** section, select **Set Permanent Tunnels**. The following Permanent Tunnel modes are then made available:

- **On all tunnels in the community**
- **On all tunnels of specific gateways**
- **On specific tunnels in the community**

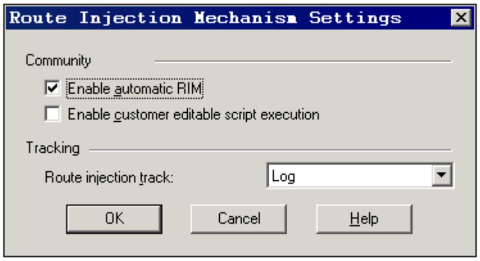
For more information on these options, see "Permanent Tunnels" on page 170.

When choosing tunnels, keep in mind that RIM can only be enabled on tunnels that have been configured to be permanent. To configure permanent tunnels, see "Configuring Tunnel Features" on page 168.

2. Select **Enable Route Injection Mechanism (RIM)**.

3. Click **Settings...**

   The **Route Injection Mechanism** Settings window open



Route Injection Mechanism Settings

Community

☑ Enable automatic RIM

□ Enable customer editable script execution

Tracking

Route injection track: Log

Decide if:

- RIM should run automatically on the gateways (SecurePlatform, IPSO or Linux only).
- A customized script should be run on the gateway whenever a tunnel changes its state (goes up or down).

For tracking options, see "Tracking Options" on page 188

4. If a customized script is run, edit `custom_rim` (`.sh` or `.bat`) script in the `$FWDIR/Scripts` directory on each of the gateways.

Enabling the RIM_inject_peer_interfaces flag

To enable the `RIM_inject_peer_interfaces` flag:

1. In SmartDashboard, click **Policy > Global Properties**.
2. Go to **SmartDashboard Customization > Configure > VPN Advanced Properties > Tunnel Management.**
3. Select `RIM_inject_peer_interfaces`.
4. Click **OK**.

Tracking Options

Several types of alerts can be configured to keep administrators up to date on the status of gateways. The Tracking settings can be configured on the **Route Injection Mechanism Settings** page. The different options are **Log**, **Popup Alert**, **Mail Alert**, **SNMP Trap Alert**, and **User Defined Alert**.

Chapter 9

Wire Mode

In This Chapter

The Need for Wire Mode

The overall increase in VPN usage has made the need for reliable, uninterrupted connections more important than ever. When a connection fails, vital information can potentially be lost. A new connection needs to be immediately established to resend the information. By avoiding stateful inspection, *Wire Mode* enables VPN connections to successfully failover, thus improving performance and reducing downtime.

The Check Point Solution

Wire Mode was designed to improve connectivity by allowing existing connections to fail over successfully by bypassing firewall enforcement. Traffic within a VPN community is, by definition, private and secure. In many cases, the firewall and the rule on the firewall concerning VPN connections is unnecessary. Using *Wire Mode*, the firewall can be bypassed for VPN connections by defining internal interfaces and communities as "trusted".

When a packet reaches a gateway, the gateway asks itself two questions regarding the packet(s):

1. Is this information coming from a "trusted" source?
2. Is this information going to a "trusted" destination?

If the answer to both questions is yes, stateful inspection is not enforced and the traffic between the trusted interfaces bypasses the firewall when the VPN Community to which both gateways belong is designated as "*Wire Mode* enabled". Since no stateful inspection takes place, the possibility of packets being discarded does not exist. The VPN connection is no different from any other connection along a dedicated wire. This is the meaning of "*Wire Mode*." Since stateful inspection no longer takes place, dynamic routing protocols (which do not survive state verification in non-wire mode configuration) can now be deployed. *Wire Mode* thus facilitates Route Based VPN. For information on Route Based VPN, see "Route Based VPN" on page 129.

Wire Mode is supported for NGX (R60) gateways and forward.

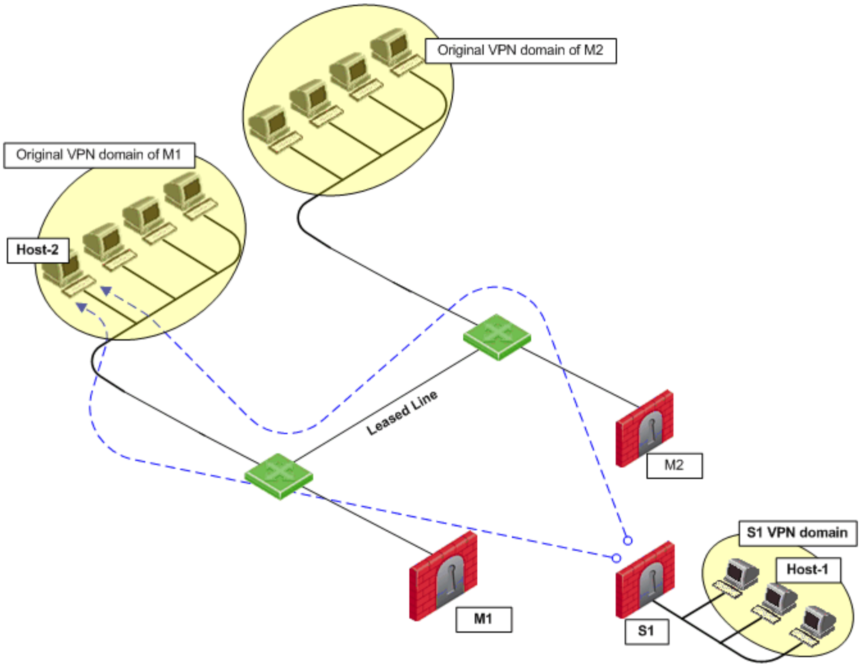
Wire Mode Scenarios

In This Section

Wire mode may be used to improve connectivity and performance in different infrastructures. This section describes scenarios that benefit from the implementation of wire mode.

Wire Mode in a MEP Configuration

Figure 9-1 Wire Mode in MEP scenario



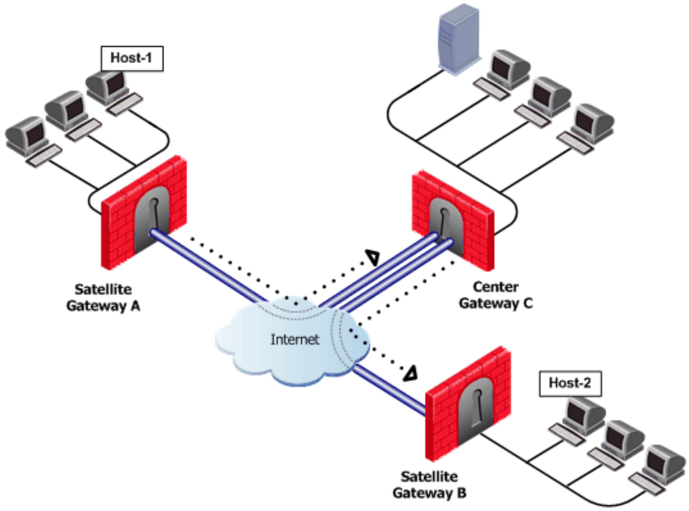
In this scenario:

- Gateway M1 and gateway M2 are both wire mode enabled and have trusted internal interfaces.
- The community where gateway M1 and gateway M2 reside, is wire mode enabled.
- Host 1, residing behind gateway S1 is communicating through a VPN tunnel with Host 2 residing behind gateway M1.
- MEP is configured for gateway M1 and gateway M2 with gateway M1 being the primary gateway and gateway M2 as the backup. For more information on MEP see, "Multiple Entry Point VPNs" on page 229.

In this case, if gateway M1 goes down, the connection fails over to gateway M2. A packet leaving Host 2 will be redirected by the router behind gateway M1 to gateway M2 since gateway M2 is designated as the backup gateway. Without wire mode, stateful inspection would be enforced at gateway M2 and the connection would be dropped because packets that come into a gateway whose session was initiated through a different gateway, are considered "out-of-state" packets. Since gateway M2's internal interface is "trusted," and wire mode in enabled on the community, no stateful inspection is performed and gateway M2 will successfully continue the connection without losing any information.

Wire Mode with Route Based VPN

Figure 9-2 Wire Mode in a Satellite community



In the scenario depicted in Figure 9-2:

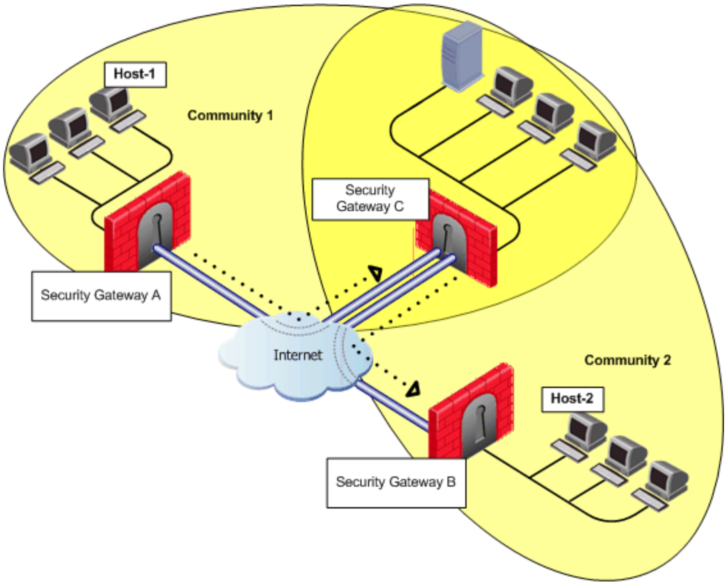
- Wire mode is enabled on Center gateway C (without an internal trusted interface specified).
- The community is wire mode enabled.
- Host 1 residing behind Satellite gateway A wishes to open a connection through a VPN tunnel with Host 2 behind Satellite gateway B.

In a satellite community, Center gateways are used to route traffic between Satellite gateways within the community.

In this case, traffic from the Satellite gateways is only rerouted by gateway C and cannot pass through gateway C's firewall. Therefore, stateful inspection does not need to take place at gateway C. Since wire mode is enabled on the community and on gateway C, making them trusted, stateful inspection is bypassed. Stateful inspection, however, does take place on gateways A and B.

Wire Mode Between Two VPN Communities

Figure 9-3 Wire Mode Between Two VPN Communities



In the scenario portrayed in Figure 9-3:

- Gateway A belongs to Community 1.
- Gateway B belongs to Community 2.
- Gateway C belongs to Communities 1 and 2.

- Wire mode is enabled on Center gateway C (without an internal trusted interface specified).
- Wire mode is enabled on both communities.
- Host 1 residing behind Satellite gateway A wishes to open a connection through a VPN tunnel with Host 2 behind Satellite gateway B.

Wire mode can also be enabled for routing VPN traffic between two gateways which are not members of the same community. Gateway C is a member of both communities and therefore recognizes both communities as trusted. When host 1 behind gateway A initiates a connection to host 2 behind gateway B, gateway C is used to route traffic between the two communities. Since the traffic is not actually entering gateway C, there is no need for stateful inspection to take place at that gateway. Stateful inspection, however, does take place on gateways A and B.

Special Considerations for Wire Mode

Currently, wire mode is only supported on SecurePlatform and Nokia IPSO platforms.

Configuring Wire Mode

Wire mode is configured in two places:

1. Community Properties (meshed or star)
2. Gateway Properties

Enabling Wire Mode on a VPN Community

1. In SmartDashboard, click **Manage** > **VPN Communities**. The **VPN Communities** window appears.
2. Select the community to be configured and click **Edit...**
3. Double-click **Advanced Settings** to view the various options.
4. Click **Wire Mode**.

   The **Wire Mode** window appears.
5. To enable Wire Mode on the community, select **Allow uninspected encrypted traffic between Wire mode interfaces of the Community's members.**
6. To enable Wire Mode Routing, select **Wire Mode Routing - Allow members to route uninspected encrypted traffic in VPN routing configurations**.

Enabling Wire Mode on a Specific Gateway

1. In SmartDashboard, click **Manage** > **Network Objects**. The **Network Objects** window will appear.
2. Select the gateway to be configured and click **Edit...**
3. Double-click **VPN** to expand **VPN** tree. Select the **VPN Advanced** to display the VPN Advanced window.
4. To enable Wire Mode on the gateway, select **Support Wire Mode**.
5. Click **Add** to include the interfaces to be trusted by the selected gateway.
6. Click **Log Wire mode traffic** to log wire mode activity.

Chapter 10

Directional VPN Enforcement

In This Chapter

The Need for Directional VPN

When a VPN community is selected in the VPN column of the Security Policy Rule Base, the source and destination IP addresses can belong to any of the gateways in the community. In other words, the traffic is bidirectional; any of the gateways can be the source of a connection, any of the gateways can be the destination endpoint. But what if the administrator (in line with the companies security policy) wished to enforce traffic in one direction only? Or to allow encrypted traffic to or from gateways *not* included in the VPN community? To enable enforcement within VPN communities, VPN implements Directional VPN.

The Check Point Solution

In This Section:

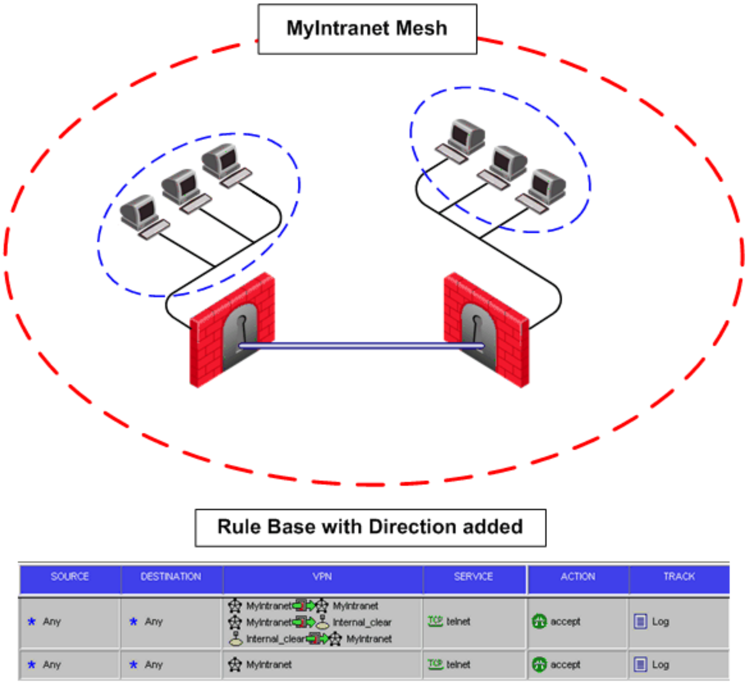
Directional VPN specifies where the source address must be, and where the destination address must be. In this way, enforcement can take place:

- Within a single VPN community
- Between VPN communities

Directional Enforcement within a Community

Figure 10-1 shows a simple meshed VPN community called *MyIntranet*. VPN traffic within the MyIntranet Mesh is bidirectional; that is, either of the gateways (or the hosts behind the gateways in the VPN domains) can be the source or destination address for a connection.

Figure 10-1 Directional Enforcement within a Community



SOURCE	DESTINATION	VPN	SERVICE	ACTION	TRACK
* Any	* Any	MyIntranet → MyIntranet MyIntranet → Internal_clear Internal_clear → MyIntranet	telnet	accept	Log
* Any	* Any	MyIntranet	telnet	accept	Log

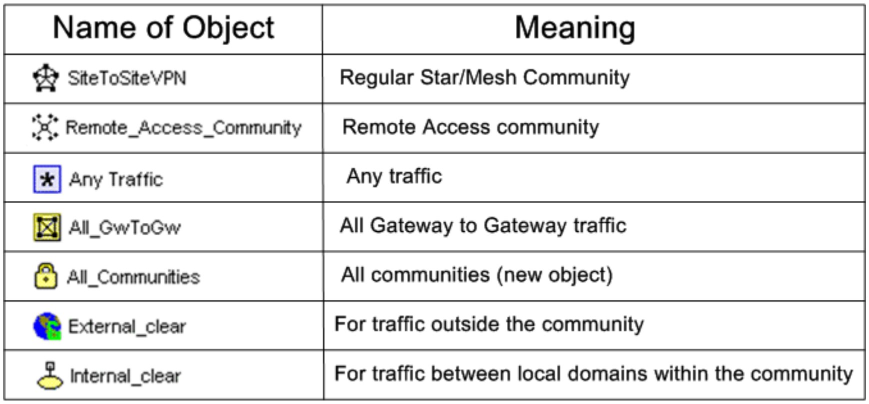
The match conditions are represented by a series of compound objects. The match conditions enforce traffic in the following directions:

- To and from the VPN Community via VPN routing (**MyIntranet => MyIntranet**)
- From the Community to the local VPN domains (**MyIntranet =>internal_clear**)
- From the local VPN domains to the VPN community (**internal_clear => MyIntranet**)

Configurable Objects in a Direction

Figure 10-2 below lists all the objects that can be configured in a direction. This includes three new objects created for Directional VPN:

Figure 10-2 Objects List



Name of Object	Meaning
SiteToSiteVPN	Regular Star/Mesh Community
Remote_Access_Community	Remote Access community
Any Traffic	Any traffic
All_GwToGw	All Gateway to Gateway traffic
All_Communities	All communities (new object)
External_clear	For traffic outside the community
Internal_clear	For traffic between local domains within the community

Note - Clear text connections originating from the following objects are not subject to enforcement:

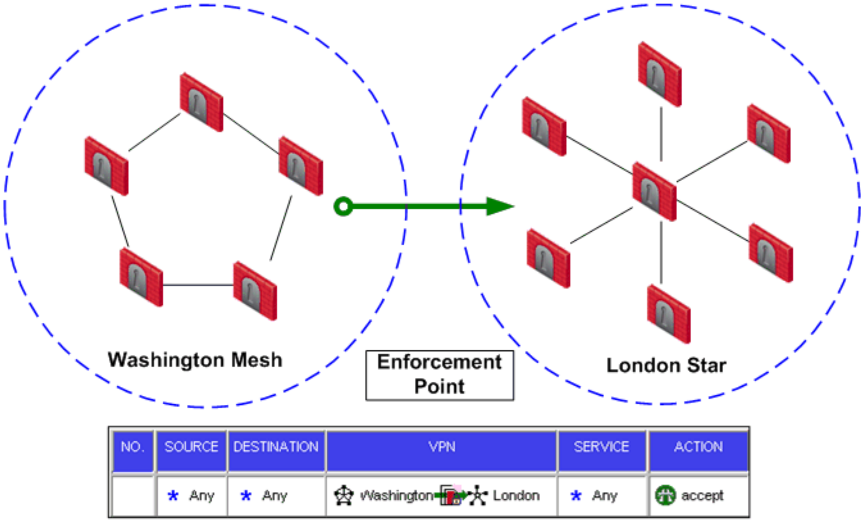
- Any Traffic
- External_clear
- Internal_clear

There is *no limit* to the number of VPN directions that can be configured on a single rule. In general, if you have many directional enforcements, consider replacing them with a standard bidirectional condition.

Directional Enforcement between Communities

VPN Directional enforcement can take place between VPN communities. Figure 10-3 shows two VPN communities, *Washington* and *London*:

Figure 10-3 Directional VPN between a mesh and star communities



NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION
	* Any	* Any	Washington → London	* Any	accept

Washington is a Mesh community, and London is a VPN Star. In the VPN column of the Security Policy Rule Base, a directional VPN rule has been implemented. This means that for a VPN connection to match this rule, the source of the connection must be in the Washington Mesh, and the destination host must be within the London Star.

This does not mean that "return" or "back" connections are not allowed from London to Washington (the three-way handshake at the start of every TCP connection demands return connections), only that the *first packet* must originate within the Washington Mesh. If a host within the London Star tries to open a connection to a host in the Washington Mesh, the connection is dropped.

This directional enforcement does not affect the topology of either Washington or London. The enforcement can be thought of as taking place somewhere between the two communities.

Configuring Directional VPN

In This Section:

Configuring Directional VPN Within a Community

To configure Directional VPN within a community:

1. In **Global Properties > VPN** page **> Advanced >** Select **Enable VPN Directional Match in VPN Column**.
2. In the VPN column of the appropriate rule, right-click on the VPN community. From the pop-up menu, select **Edit Cell....**

   The **VPN Match Conditions window opens**.
3. Select **Match traffic in this direction only**, and click **Add...**

   The **Directional VPN Match Condition** window opens.
4. In the **Match on traffic reaching the Gateway from:** drop-down box, select the object for **internal_clear**. (the source).
5. In the **Match on traffic leaving the Gateway to:** box, select the relevant community object (the destination).
6. Add another directional match in which the relevant community object is both the source and destination.

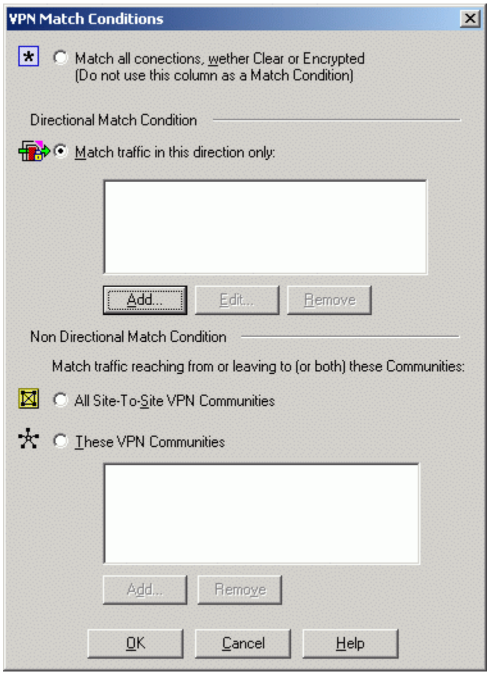
   This allows traffic from the local domain to the community, and within the community.
7. Click **OK.**

Configuring Directional VPN Between Communities

To configure Directional VPN between communities:

1. In **Global Properties > VPN** page **> Advanced >** Select **Enable VPN Directional Match in VPN Column**.
2. Right-click inside the VPN column of the appropriate rule. From the pop-up menu, select **Edit Cell...** or **Add Direction...**

   The **VPN Match Conditions** window opens.



VPN Match Conditions

○ Match all conections, wether Clear or Encrypted
(Do not use this column as a Match Condition)

Directional Match Condition

● Match traffic in this direction only:

Non Directional Match Condition

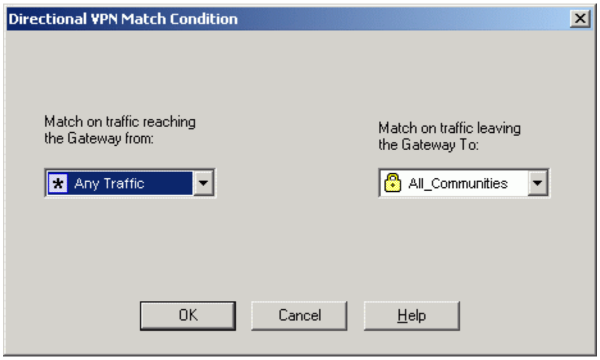
Match traffic reaching from or leaving to (or both) these Communities:

○ All Site-To-Site VPN Communities

○ These VPN Communities

3. Click **Add...**

The **Directional VPN Match Conditions** window opens:



Directional VPN Match Condition

Match on traffic reaching the Gateway from: Any Traffic

Match on traffic leaving the Gateway To: All_Communities

4. From the drop-down box on the left, select the source of the connection.
5. From the drop-down box on the right, select the connection’s destination.
6. Click **OK**.

Chapter 11

Link Selection

In This Chapter

Overview

Link Selection is a method used to determine which interface is used for incoming and outgoing VPN traffic as well as the best possible path. Using the Link Selection mechanisms, the administrator can focus individually on which IP addresses are used for VPN traffic on each gateway.

Configuration settings for remote access clients can be configured together or separately from the Site-to-Site configuration. For more information, see "Link Selection for Remote Access Clients" on page 567.

Using Link Selection

In This Section

Link Selection employs two mechanisms:

- IP Selection by Remote Peer for incoming traffic.
- Outgoing Route Selection for outbound traffic.

IP Selection by Remote Peer

There are several methods that can determine how remote peers resolve the IP address of the local gateway. Remote peers can connect to the local gateway using:

- **Always use this IP address:**
  - **Main address** - The VPN tunnel is created with the gateway main IP, specified in the **IP Address** field on the **General Properties** page of the gateway.
  - **Selected address from topology table** - The VPN tunnel is created with the gateway using a selected IP address chosen from the drop down menu that lists the IP addresses configured in the **Topology** page of the gateway.
  - **Statically NATed IP** - The VPN tunnel is created using a NATed IP address. This address is not required to be listed in the topology tab.
- **Calculate IP based on network topology** - This method calculates the IP address used for the VPN tunnel by network topology based on the location of the remote peer. For more information, see "Link Selection for Remote Access Scenarios" on page 570.
- **DNS Resolving -** This method is required for Dynamically Assigned IP (DAIP) gateways. A VPN tunnel to a DAIP gateway can only be initiated using DNS resolving since the IP address of the DAIP gateway cannot be known in advance. If using this method for a non-DAIP gateway, the IP address must be defined in the **Topology** tab. Without DNS resolving, a DAIP gateway can only initiate the first connection between two peers. The second connection can be initiated by the peer gateway as long as the IP address of the DAIP gateway has not changed.

  - **Full hostname** - Enter the full Fully Qualified Domain Name (FQDN). The DNS host name that is used is "gateway_name.domain_name." For example, if the object name is "john" and the domain name is "smith.com" then the FQDN will be "john.smith.com."
  - **Gateways name and domain name (Specified in global properties)** - The gateway name is derived from the **General Properties** page of the gateway and the domain name is derived from the Global Properties page.
- **Use a probing method:**
  - **Using ongoing probing** - When a session is initiated, all possible destination IP addresses continuously receive RDP packets. The VPN tunnel uses the first IP to respond (or to a primary IP if a primary IP is configured and active), and stays with this IP until the IP stops responding. The RDP probing is activated when a connection is opened and continues as a background process.
  - **Using one time probing** - When a session is initiated, all possible destination IP addresses receive an RDP session to test the route. The first IP to respond is chosen, and stays chosen until the next time a policy is installed.

RDP Probing

When more than one IP address is available on a gateway for VPN, *Link Selection* may employ a probing method to determine which link is to be used.

The probing methods of choosing a destination IP are implemented using a proprietary protocol that uses UDP port 259. This protocol:

- Is proprietary to Check Point
- Does not comply with RDP as specified in RFC 908/1151
- Works only between Check Point entities

IP addresses you do not wish to be probed (i.e., internal IP addresses) may be removed from the list of IP’s to be probed - see “Resolving Addresses via Probing” on page 224.

For both the probing options (one-time and on-going) a *Primary Address* can be assigned.

Note - UDP RDP packets are not encrypted. The RDP mechanism tests connectivity only.

Primary Address

When implementing one of the probing methods on a gateway that has a number of IP addresses for VPN, one of the IP addresses can be designated as a Primary Address. As a result, peers assign the primary address IP a higher priority even if other interfaces have a lower metric.

Enabling a primary address has no influence on the IP selected for outgoing VPN traffic. If the remote gateway connects to a peer gateway that has a primary address defined, then the remote gateway will connect to the primary address (if active) regardless of network speed (latency) or route metrics.

If the primary address fails and the connection fails over to a backup, the VPN tunnel will stay with the backup until the primary becomes available again.

Last Known Available Peer

The IP address used by a gateway during a successful IKE negotiation with a peer gateway, is used by the peer gateway as the destination IP address for the next IPSec traffic and next IKE negotiations initiated by the peer gateway. This is only the case when the Link Selection configuration is static (without probing).

Outgoing Route Selection

For outbound traffic, there are two methods used to determine which path to use when connecting with a remote peer:

- **Operating system routing table** (default setting) - Using this method, the routing table is consulted for the available route with the lowest metric and best match for the VPN tunnel negotiation.
- **Route based probing** - This method also consults the routing table for an available route with the lowest metric and best match. However, before a route is chosen, it will be tested for availability using RDP probing. The gateway then selects the best match (highest prefix length) active route with the lowest *metric*. This method is recommended when there is more than one external interface.

  Route based probing enables use of *On Demand Links (ODL)* which are triggered upon failure of all primary links. A script is run to activate an On Demand Link when all other links with higher priorities become unavailable. The ODL's metric must be set to be larger than a configured minimum in order for it to be considered an ODL. For more information, see "On Demand Links (ODL)" on page 218.

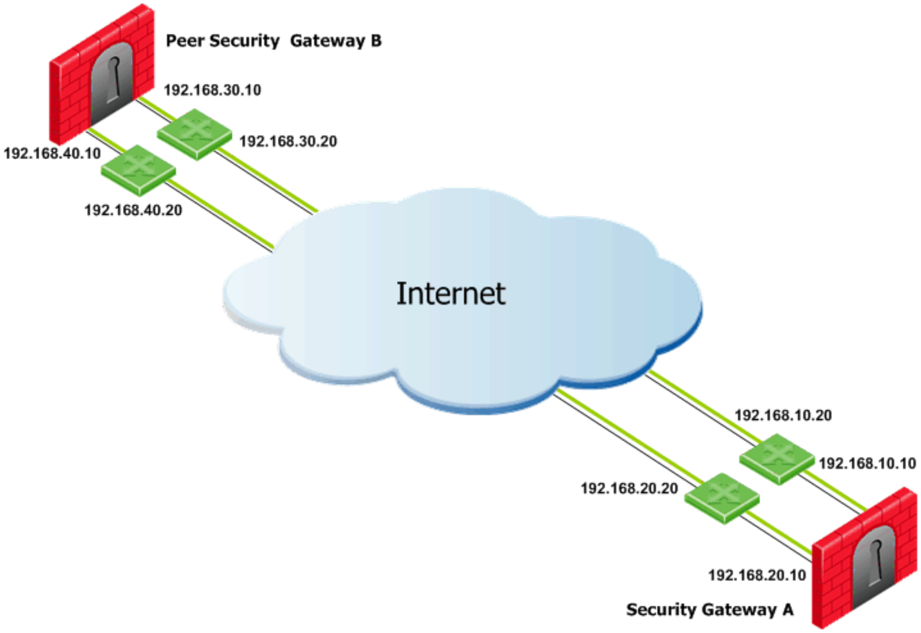
For IKE and RDP sessions, Route Based probing uses the same IP address and interface for responding traffic.

Route Based probing is only supported on the SecurePlatform, Linux, and Nokia IPSO platforms.

Using Route Based Probing

The local gateway, using RDP probing, considers all possible routes between itself and the remote peer gateway. The gateway then decides on the most effective route between the two gateways, as shown in Figure 11-1.

Figure 11-1 Route Based Probing



In this scenario, gateway A has two external interfaces, 192.168.10.10 and 192.168.20.10. Peer gateway B also has two external interfaces: 192.168.30.10 and 192.168.40.10.

For gateway A, the routing table reads:

Table 11-1

Destination	Netmask	Next hop	Metric
192.168.40.10	255.255.255.0	192.168.10.20	1
192.168.40.10	255.255.255.0	192.168.20.20	2

For gateway B, the routing table reads:

Table 11-2

Destination	Netmask	Next hop	Metric
192.168.20.10	255.255.255.0	192.168.40.20	1
192.168.20.10	255.255.255.0	192.168.30.20	2

If both routes for outgoing traffic from gateway A are available, the route from 192.168.10.10 to 192.168.40.10 has the lowest metric (highest priority) and is therefore the preferred route.

Responding Traffic

When responding to a remotely initiated tunnel, there are two options for selecting the interface and next hop that is used (these settings are relevant for IKE and RDP sessions only):

- **Use outgoing traffic configuration** - Select this option to choose an interface using the same method selected in the **Outgoing Route Selection** section.
- **Reply from the same interface** - This option will send the returning traffic through the same interface and next hop it came in.

Note - When Route Based Probing is enabled, **Reply from the same interface** is the selected method.

Source IP Address Settings

The source IP address used for outgoing packets can be configured for sessions initiated by the gateway.

When initiating a VPN tunnel, set the source IP address using one of the following:

- **Automatic (derived from the method of IP selection by remote peer)** - The source IP address of outgoing traffic is derived from the method selected in the **IP Selection by Remote Peer** section.

  If **Main address or Selected address from topology table** are chosen in the **IP Selection by Remote Peer** section, then the source IP when initiating a VPN tunnel is the IP specified for that method.

If **Calculate IP based on network topology**, **Statically NATed IP, Use DNS resolving** or **Use a probing method** is chosen in the **IP Selection by Remote Peer** section, then the source IP when initiating a VPN tunnel is the IP address of the chosen outgoing interface.

- **Manual**:
  - **Main IP address -** The source IP is derived from the **General Properties** page of the gateway.
  - **Selected address from topology table -** The chosen IP from the drop down menu becomes the source IP.
  - **IP address of chosen interface -** The source IP is the same IP of the interface where the traffic is being routed through.

These settings are relevant for RDP and IKE sessions. When responding to an IKE session, use the `reply_from_same_IP (default: true)` attribute to follow the settings in the **Source IP address settings** window or to respond from the same IP. For more information, see "Configuring Source IP Address Settings" on page 226.

Note - When Route Baed Probing is enabled, `reply_from_same_IP` will be seen as **true**.

Link Selection Scenarios

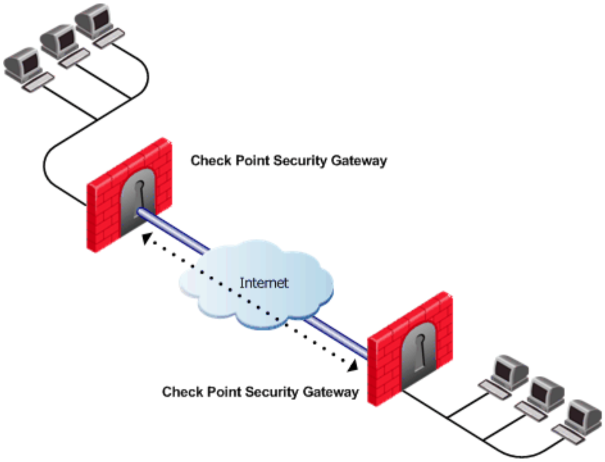
Link Selection may be used in different infrastructures. This section describes scenarios that benefit from the implementation of *Link Selection*.

In This Section

Gateway with a Single External Interface

This is the simplest case. In Figure 11-2, the local gateway has a single external interface for VPN:

Figure 11-2 Single IP for VPN



Now consider configuration for the local gateway in terms of:

- How peer gateways select an IP on the local gateway for VPN traffic

Since there is only one interface available for VPN:

- For determining how remote peers discover the local gateways IP for VPN, select **Main address** or choose an IP address from the **Selected address from topology table** drop down menu.
- If the IP address is located behind a static NAT device, select **Statically NATed IP**.

To configure, see: "Resolving Addresses via Main and Single IPs" on page 223

Gateway with a Dynamic IP Address (DAIP)

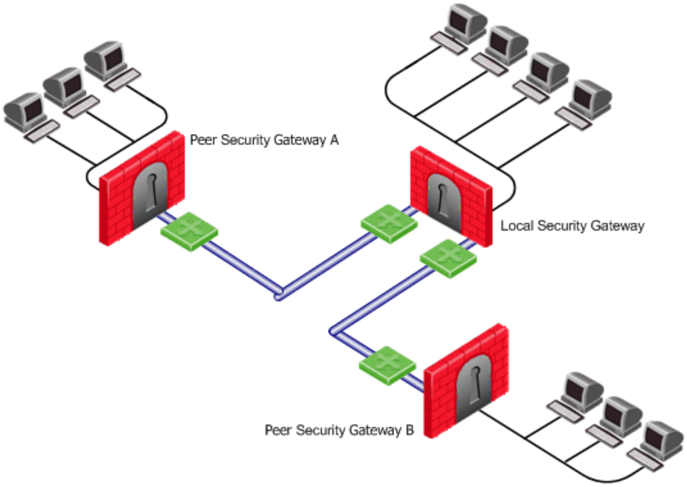
A VPN tunnel negotiation with a DAIP gateway can only be initiated using DNS resolving since the IP address of the DAIP gateway cannot be known in advance. The peer gateway can then resolve the DAIP gateways IP address by using its hostname. The hostname can be entered on the Link Selection page or can be derived from the global properties page. Without DNS resolving, a DAIP gateway can only initiate a connection. The second connection can be initiated by the peer gateway as long as the IP address of the DAIP gateway has not changed.

To configure, see "Resolving Addresses using DNS lookup" on page 224.

Gateway with Several IP Addresses Used by Different Parties

In this scenario, the local gateway has a point-to-point connection from two different interfaces. Each interface is used by a different remote party:

Figure 11-3 Several IP Addresses used by different parties



In Figure 11-3, the local gateway has two IP addresses used for VPN. One interface is used for VPN with a peer gateway A and one interface for peer gateway B.

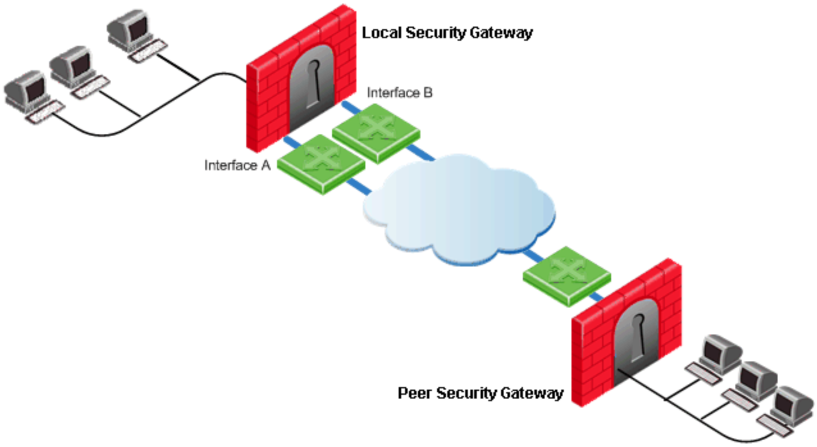
For determining how peer gateways discover the local gateway's IP, enable one-time probing. Since only one IP is available for each peer gateway, probing only has to take place one time.

See: “Resolving Addresses via Probing” on page 224.

Gateway With One External Interface and Interface Behind a Static NAT Device

In this scenario, the local gateway has two external interfaces available for VPN. The address of interface A is being translated using a NAT device.

Figure 11-4 Local Gateway hidden behind a NATing device

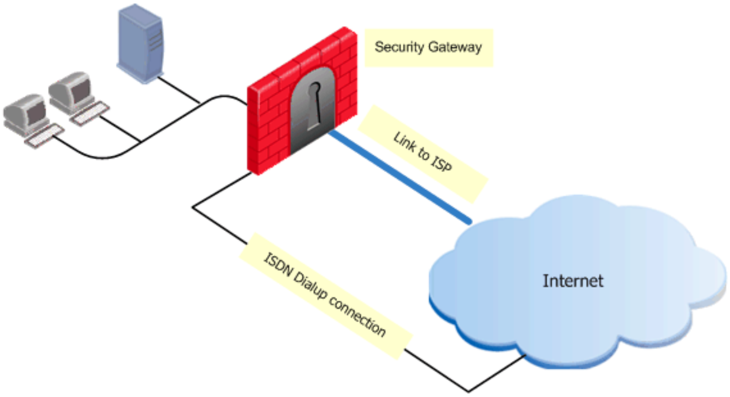


For determining how peer gateways discover the local gateway’s IP, use *ongoing probing*. In order for the Static NAT IP address to be probed, it must be added to the **Probe the following addresses** list in the **Probing Settings** window. (See: “Resolving Addresses via Probing” on page 224).

On Demand Links (ODL)

Route based probing enables use of an On Demand Link which is triggered upon failure of all primary links. When a failure is detected, a custom script is used to activate the ODL and change the appropriate routing information. The ODL's metric must be set to be larger than a configured minimum in order for it to be considered an ODL.

Figure 11-5 On Demand Links



In Figure 11-5 the gateway has two external links: one to an ISP, the other an ISDN dialup, both of which provide connectivity to the Internet.

On the gateway shown in Figure 11-5, the Route Based Probing mechanism probes all the non On Demand Links and selects the active link with the lowest metric. A script is run to activate an On Demand Link when all other links with higher priorities become unavailable. When the link becomes available again, a shut down script is run and the connection continues through the link with the ISP.

See: "Configuring On Demand links" on page 227.

Note - On Demand Links are probed only once using a single RDP session. For this reason, fail over between On Demand Links is not supported.

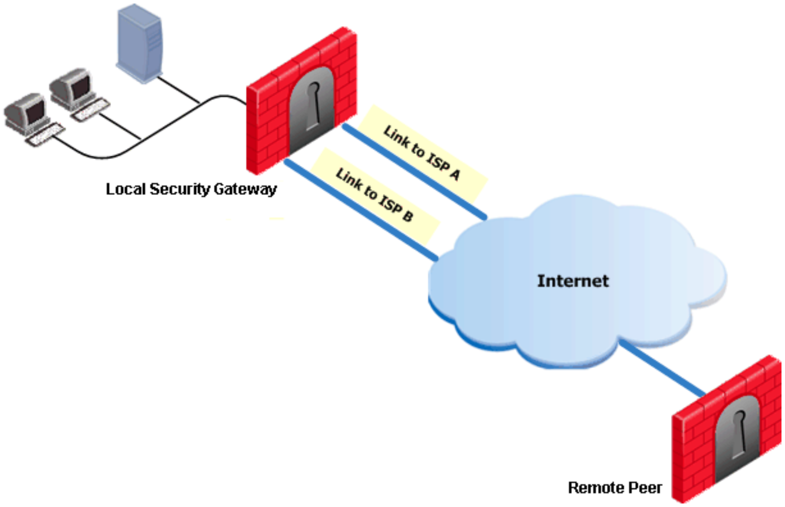
Link Selection and ISP Redundancy

ISP Redundancy enables reliable Internet connectivity by allowing a single or clustered security gateway(s) to connect to the Internet via redundant ISP connections. As part of standard VPN installation, it offers two modes of operation:

- **Load Sharing** mode connects to both ISPs while sharing the load of outgoing connections between the ISPs. New connections are randomly assigned to a link. If a link fails, all new outgoing connections are directed to the active link. This configuration effectively increases the WAN bandwidth while providing connectivity protection. This is for firewall traffic only. For VPN traffic, this method provides redundancy.
- **Primary/Backup** mode connects to an ISP through the primary link, and switches to a backup ISP if the primary ISP link fails. When the primary link is restored, new outgoing connections are assigned to it, while existing connections are maintained over the backup link until they are complete.

The settings configured in the *ISP Redundancy* window are by default, applied to the *Link Selection* page and will overwrite any pre-existing configuration. If the *Primary/Backup* setting is configured, this will be carried over to the *Link Selection* configuration. For more information on ISP Redundancy, see the *ISP Redundancy* chapter in the *Firewall Administration Guide*.

Figure 11-6 Local Gateway links to more than one ISP

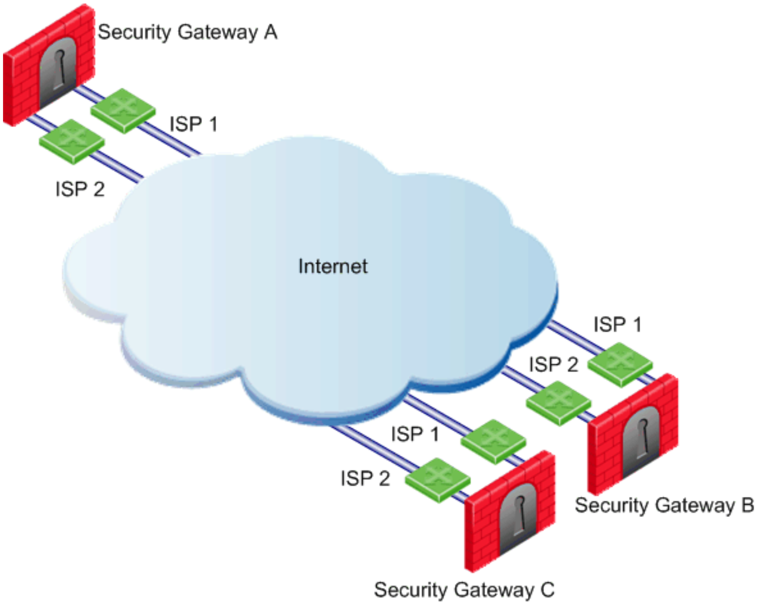


In Figure 11-6, the local gateway maintains links to ISPs A and B, both of which provide connectivity to the Internet.

On the **VPN > Topology > ISP Redundancy** window, configure the appropriate settings. When *ISP Redundancy* is configured, the default setting in the *Link Selection* page is **Use ongoing probing**. However, *Link Selection* will only probe the ISP's configured in the *ISP Redundancy* window. This enables connection failover of the VPN tunnel if connectivity to one of the gateway interfaces fails.

There are instances when having a different configuration for Link Selection is required.

Figure 11-7 Two Gateways with two ISP's



In this scenario:

- Security gateways A, B, and C have two ISP's.
- *ISP Redundancy* is configured on security gateway A.
- Gateway A should use ISP 1 in order to connect to gateway B and ISP 2 in order to connect to gateway C. If one of the ISPs becomes unavailable, the other ISP should be used.

In Figure 11-7, the administrator of gateway A needs to do three things:

- Uncheck the **Apply settings to VPN traffic** box in the *ISP Redundancy* window.
- Reconfigure the *Outgoing Route Selection* to *Route Based Probing* in the *Link Selection* window.
- Configure the routing table so that ISP 1 is the highest priority for peer gateway B and ISP 2 has the highest priority for peer gateway C.

In this scenario, load distribution is achieved since different ISPs are used for different remote peer gateways. When a large number of remote peer gateways are involved, it is possible to configure a different preferred ISP for each group of peer gateways.

Early Versions Compatibility Resolving Mechanism

When connecting with gateways that are pre-NGX, the *Early Versions Compatibility - Resolving Mechanism* effects the *Link Selection* method.

If an R70 security gateway is configured with a *Link Selection* method not available in previous versions, the pre NGX gateway will not be able to resolve which IP address to use to connect. By configuring the *Early Versions Compatibility - Resolving Mechanism*, the R70 gateway "assigns" a method to the pre-NGX gateway that it is familiar with so it can resolve the IP address of the R70 security gateway. See: "Configuring the Early Version Compatibility Resolving Mechanism" on page 228.

Configuring Link Selection

In This Section:

Link Selection is configured on each gateway in the **VPN > Link Selection** window.

Resolving Addresses via Main and Single IPs

If remote VPN peers should connect to the main IP address of the gateway or a single IP address reserved for VPN traffic, then on the **VPN > Link Selection** page of the gateway object, select one of the following:

- **Main address** (of the gateway)
- **Selected address from topology table** (select an IP address from the drop down menu
- **Statically NATed IP** (enter the required IP address)

Resolving Addresses using DNS lookup

If remote VPN peers should resolve the local gateway's IP address through DNS lookup, then:

1. On the **VPN > Link Selection** page of the gateway object, select **Use DNS resolving:**

   The fully qualified domain name (FQDN) queried can be set on this page or derived from the **Global Properties**.

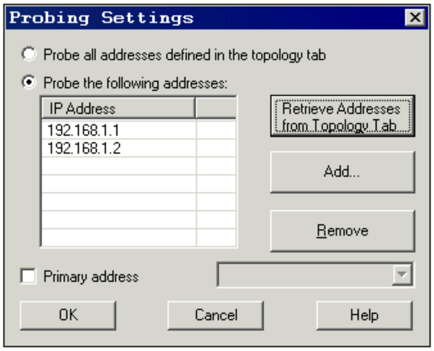
   - **Full hostname -** Enter the FQDN of the gateway, e.g. `www.checkpoint.com` (not which DNS server should be queried). `www` is the host, `checkpoint` is the second-level domain, and `.com` is the top-level domain.
   - **Derived from global properties -** By selecting this method, the host name is derived from the **General Properties** page in the gateway and the domain name is derived from the **Global Properties > VPN** page.

Resolving Addresses via Probing

If remote peers should resolve the local gateway's IP address via RDP probing, then:

1. On the **VPN > Link Selection** page of the gateway object, select **Use a probing method.**
2. Select **Using ongoing probing** or **Using one time probing**. Click **Configure...**

   The **Probing Settings** window opens.



Probing Settings

○ Probe all addresses defined in the topology tab

● Probe the following addresses:

IP Address	
192.168.1.1	
192.168.1.2	

□ Primary address

Select one of the following:

- **Probe all addresses defined in the topology tab**
- **Probe the following addresses**

The remote peer can probe all of the available interfaces, or only those IP addresses manually defined in the list. Statically NATed IP addresses (that are not assigned to any interface configured in the topology tab) can be added to the manually defined IP list.

Configuring an interface that has priority over the other interfaces is only relevant for a gateway whose IP is resolved via one time or ongoing probing. Select **Primary Address** and select the interface from the drop-down box.

Configuring Outgoing Route Selection

Outgoing Route Selection is configured on each gateway in the **VPN > Link Selection** window.

To select the method to choose an interface for outbound traffic select either:

- **Operating system routing table** to use the link with the highest priority.
- **Route based probing** to probe all links and send traffic with active link with the highest priority.

Configuring For Responding Traffic

On the **Link Selection** page:

1. Click **Setup**.
2. Choose either:
   - **Use outgoing traffic configuration** (default) to use the settings configured in the **Outgoing Route Selection** section.
   - **Reply from the same interface** to route traffic back to the interface and next hop that it came through.

Configuring Source IP Address Settings

On the **Link Selection** page:

1. Click **Source IP address settings...**
2. Choose either:
   - **Automatic (derived from the method of IP selection by remote peer)** (default) - The source IP address of outgoing traffic is derived from the method selected in the **IP Selection by Remote Peer** section.
   - **Manual**:
     - **Main IP address -** The source IP is derived from the **General Properties** page of the gateway.
     - **Selected address from topology table -** The chosen IP from the drop down menu becomes the source IP.
     - **IP address of chosen interface -** The source IP is the same IP of the interface where the traffic is being routed through.

When responding to an IKE session, use Dbedit to configure the `reply_from_same_IP (default:` **true**) attribute to follow the settings in the **Source IP address settings** window or to respond from the same IP. When set to true, the same IP address will be used. When set to false, the IP address will be derived from the **Source IP address settings** window.

Configuring On Demand links

On Demand Links can only be enabled when *Route Based Probing* is configured. The properties to edit are:

Table 11-3 Configuring ODL

Property	**Description**
`use_on_demand_links`	Enables on-demand links (default: FALSE)
`on_demand_metric_min`	Defines the minimum metric level for an on-demand link. A link is considered on-demand only if its metric is equal or higher than the configured minimum metric.
`on_demand_initial_script`	This property contains the name of the on-demand script. This script is run when all not-on-demand routes stop responding. Place the script in the `$FWDIR/conf` directory.
on_demand_shutdown_script	This script is run when the failed links become available. Place the script in the `$FWDIR/conf` directory.

The On Demand Links commands are configured by changing a property using the database tool **DBedit**.

The commands `use_on_demand_links` and `on_demand_metric_min`, may also be configured as follows:

1. In SmartDashboard, click **Policy > Global Properties > SmartDashboard Customization > Configure**.
2. Expand the **VPN Advanced Properties** tree and click on the **Link Selection** page.
3. Click the **use_on_demand_links** checkbox to enable On Demand Links.
4. Set the minimum metric level for an On Demand Link next to the **on_demand_metric_min** command.

Configuring the Early Version Compatibility Resolving Mechanism

In SmartDashboard:

1. Click **Policy > Global Properties > VPN > Early Versions Compatibility.**
2. Select **Static calculation based on network topology** if pre NGX gateways should use topology calculation to resolve IP addresses.
3. Select **Dynamic interface resolving mechanism** to convert to one of the methods in Table 11-4.

The R70 gateway will "convert" the method used by the Pre NGX gateway as follows:

Table 11-4 Backward Compatibility

R70 Gateway Method	Method used by Pre NGX Gateway
Selected address from topology table	Ongoing Probing
Statically NATed IP	Ongoing Probing
Use DNS resolving	Ongoing Probing
Calculate IP based on network topology	Main IP
Main IP	Main IP
Ongoing Probing	Ongoing Probing
One Time Probing	One Time Probing

Note - If the manual IP address list for probing is configured, Pre NGX (R60) gateways will probe all of the IP addresses and not just those listed in the IP address list.

Outgoing Link Tracking

When link tracking is activated on the local gateway, the gateway sends a log for every new resolving decision performed to one of its remote VPN peers (VPN tunnels). When dynamic ongoing resolving is configured on the remote peer or when Route Based Probing is activated on the local gateway, log entries are issued for all resolving changes.

Chapter 12

Multiple Entry Point VPNs

In This Chapter

Overview

Multiple Entry Point (MEP) is a feature that provides a high availability and load sharing solution for VPN connections. A gateway on which the VPN module is installed provides a single point of entry to the internal network. It is the gateway that makes the internal network “available” to remote machines. If a gateway should become unavailable, the internal network too, is no longer available. A MEPed environment has two or more security gateways both protecting and enabling access to the same VPN domain, providing peer gateways with uninterrupted access.

VPN High Availability Using MEP or Clustering

Both MEP and Clustering are ways of achieving High Availability and load sharing. However:

- Unlike the members of a ClusterXL gateway Cluster, there is no physical restriction on the location of MEPed gateways. MEPed gateways can be geographically separated machines. In a cluster, the clustered gateways need to be in the same location, directly connected via a *sync* interface.
- MEPed gateways can be managed by different Security Management servers; cluster members must be managed by the same Security Management server.
- In a MEP configuration there is no “state synchronization” between the MEPed gateways. In a cluster, all of the gateways hold the “state” of all the connections to the internal network. If one of the gateways fails, the connection passes seamlessly over (performs *failover*) to another gateway, and the connection continues. In a MEPed configuration, if a gateway fails, the current connection is lost and one of the backup gateways picks up the *next* connection.
- In a MEPed environment, the decision which gateway to use is taken on the remote side; in a cluster, the decision is taken on the gateway side.

How It Works

MEP is implemented via a proprietary *Probing Protocol* (PP) that sends special UDP RDP packets to port 259 to discover whether an IP is reachable. This protocol is proprietary to Check Point and does not conform to RDP as specified in RFC 908/1151.

Note - These UDP RDP packets are not encrypted, and only test the availability of a peer.

The peer continuously probes or polls all MEPed gateways in order to discover which of the gateways are "up", and chooses a gateway according to the configured selection mechanism. Since RDP packets are constantly being sent, the status of all gateways is known and updated when changes occur. As a result, all gateways that are "up" are known.

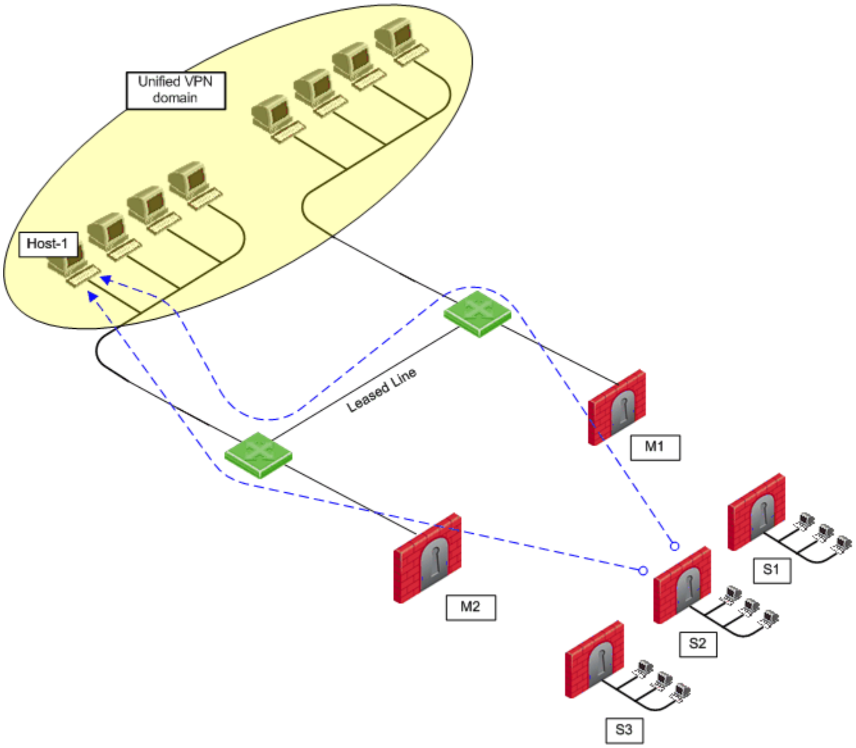
There are two available methods to implement MEP:

- Explicit MEP - Only Star communities with more than one central gateway can enable explicit MEP, providing multiple entry points to the network behind the gateways. When available, Explicit MEP is the recommended method.
- Implicit MEP - Implicit MEP is supported in all scenarios where fully or partially overlapping encryption domains exist or where *Primary-Backup Gateways* are configured. When upgrading from a version prior to NGX (R60) where Implicit MEP was already configured, the settings previously configured will remain.

Explicit MEP

In a site to site Star VPN community, explicit MEP is configured via the community object. When MEP is enabled, the satellites consider the "unified" VPN domain of all the gateways as the VPN domain for each gateway. This unified VPN domain is considered the VPN domain of each gateway, as shown in Figure 12-1

Figure 12-1 Unified encryption domain



In Figure 12-1, a Star VPN community has two central gateways, M1 and M2 (for which MEP has been enabled) and three satellite gateways — S1, S2, and S3. When S2 opens a connection with host-1 (which is behind M1 and M2), the session will be initiated through either M1 or M2. Priority amongst the MEP gateways is determined by the MEP entry point selection mechanism.

If M2 is the selected entry point and becomes unavailable, the connection to host-1 fails over to M1. Returning packets will be rerouted using RIM or IP Pool NAT. For more information about returning packets, see "Routing Return Packets" on page 244.

There are four methods used to choose which of the gateways will be used as the entry point for any given connection:

- Select the closest gateway to source (First to respond)
- Select the closest gateway to destination (By VPN domain)
- Random Selection (for Load distribution)
- Manually set priority list (MEP rules)

If either “By VPN domain” or “Manually set priority list” is selected, then **Advanced** options provide additional granularity.

MEP Selection Methods

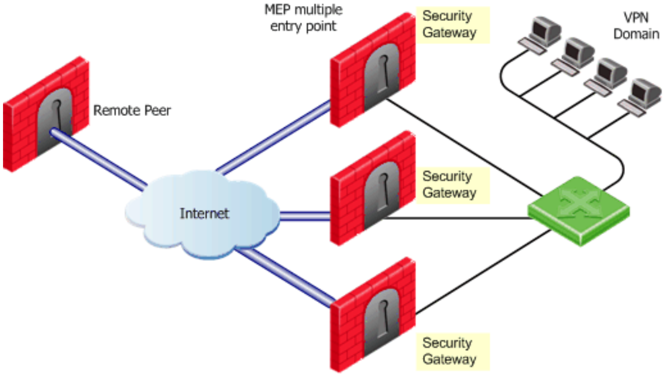
- **First to Respond**, in which the first gateway to reply to the peer gateway is chosen. An organization would choose this option if, for example, the organization has two gateways in a MEPed configuration - one in London, the other in New York. It makes sense for peers located in England to try the London gateway first and the NY gateway second. Being geographically closer to the peers in England, the London gateway will be the first to respond, and becomes the entry point to the internal network. See: “First to Respond” on page 234.
- **VPN Domain,** is when the destination IP belongs to a particular VPN domain, the gateway of that domain becomes the chosen entry point. This gateway becomes the primary gateway while other gateways in the MEP configuration become its backup gateways. See: “By VPN Domain” on page 235.
- **Random Selection**, in which the remote peer randomly selects a gateway with which to open a VPN connection. For each IP source/destination address pair, a new gateway is randomly selected. An organization might have a number of machines with equal performance abilities. In this case, it makes sense to enable load distribution. The machines are used in a random and equal way. See: “Random Selection” on page 236.
- **Manually set priority list,** gateway priorities can be set manually for the entire community or for individual satellite gateways. See: “Manually Set Priority List” on page 237.

First to Respond

When there is no primary gateway, all gateways share "equal priority". When all gateway's share "equal priority," as in Figure 12-2:

- Remote peers send RDP packets to all the gateways in the MEP configuration.
- The first gateway to respond to the probing RDP packets gets chosen as the entry point to network. The idea behind *first to respond* is proximity. The gateway which is "closer" to the remote peer responds first.
- A VPN tunnel is opened with the first to respond. All subsequent connections pass through the chosen gateway.
- If the gateway ceases to respond, a new gateway is chosen.

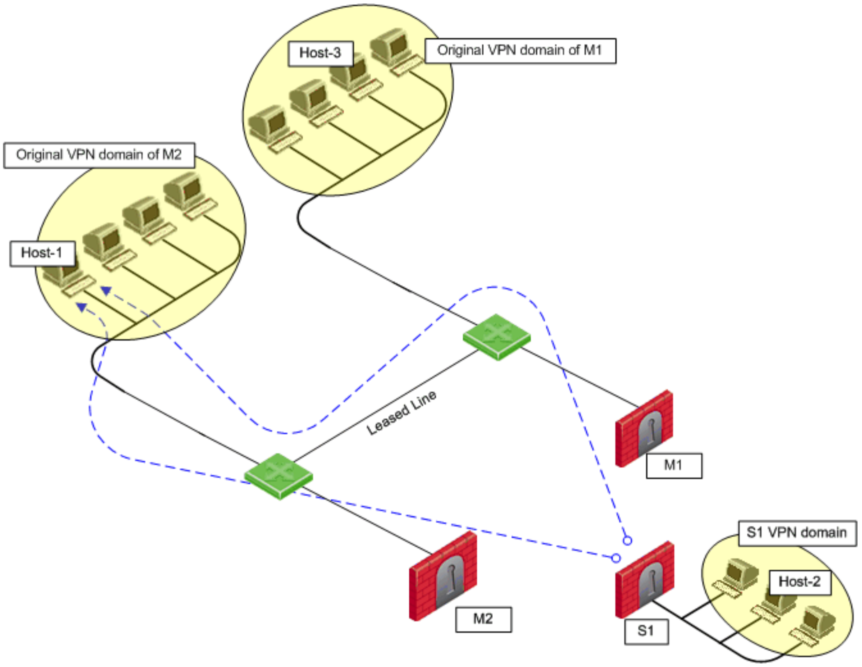
Figure 12-2 No primary, equal priority



By VPN Domain

Prior to enabling MEP, each IP address belonged to a specific VPN domain. Using *By VPN Domain*, the gateway of that domain becomes the chosen entry point. In Figure 12-3, the VPN Star community has two central MEPed gateways (M1 and M2, each of which *have their own VPN domains*), and remote satellite S1.

Figure 12-3 By VPN domain



Host-2 (in the VPN domain of satellite S1 initiates a connection with host-1. The connection can be directed through either M1 or M2. However, host-1 is within M2's original VPN domain. For this reason, M2 is considered the gateway "closest" to the destination IP Address. M2 is therefore considered the primary gateway and M1 the backup gateway for Host-1. If there were additional gateways in the center, these gateways would also be considered as backup gateways for M2.

If the VPN domains have fully or partially overlapping encryption domains, then more than one gateway will be chosen as the "closest" entry point to the network. As a result, more than one gateway will be considered as "primary." When there are more than one primary or backup gateways available, the gateway is selected using an additional selection mechanism. This advanced selection mechanism can be either (See "Advanced Settings" on page 239):

- First to Respond
- Random Selection (for load distribution)

For return packets you can use RIM on the center gateways. If RIM is also enabled, set a metric with a lower priority value for the leased line than the VPN tunnel. The satellite S1 might simultaneously have more than one VPN tunnel open with the MEPed gateways, for example M2 as the chosen entry point for host-1 and M1 as the chosen entry point for host-3. While both M1 and M2 will publish routes to host-1 and host-3, the lower priority metric will ensure the leased line is used only when one of the gateways goes down.

Random Selection

Using this method, a different gateway is randomly selected as an entry point for incoming traffic. Evenly distributing the incoming traffic through all the available gateways can help prevent one gateway from becoming overwhelmed with too much incoming traffic.

The gateways are probed with RDP packets, as in all other MEP configurations, to create a list of responding gateways. A gateway is randomly chosen from the list of responding gateways. If a gateway stops responding, another gateway is (randomly) chosen.

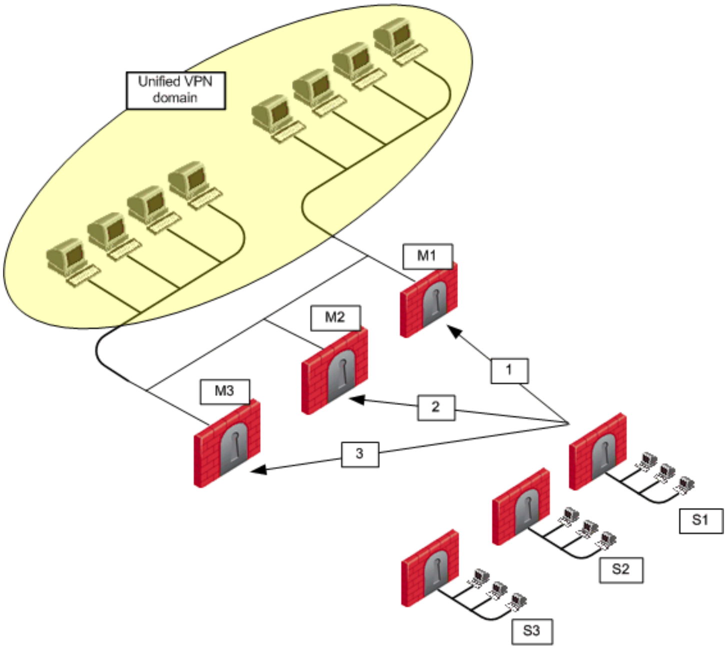
A new gateway is randomly selected for every source/destination IP pair. While the source and destination IP’s remain the same, the connection continues through the chosen gateway.

In such a configuration, RIM is not supported. IP Pool NAT must be enabled to ensure return packets are correctly routed through the chosen gateway.

Manually Set Priority List

The gateway that will be chosen (from the central gateways in the star community) as the entry point to the core network can be controlled by manually setting a priority per source gateway. Each priority constitutes a MEP Rule, as illustrated in Figure 12-4:

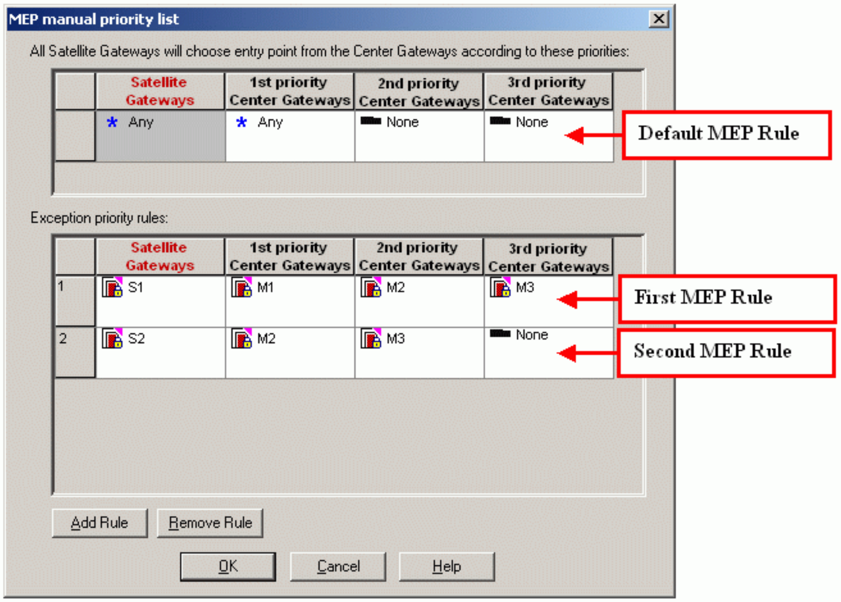
Figure 12-4 MEP Rules



In Figure 12-4, three MEP members (M1, M2, M3) provide entry points to the network for three satellite gateways (S1, S2, S3). Satellite S1 can be configured to try the gateways in the following order: M1, M2, M3, giving the highest priority to M1, and the lowest priority to M3. Satellite S2 can be configured to try the gateways in the following order: M2, M3 (but not to try M1).

Each of these priorities constitutes a MEP rule in the **MEP manual priority list** window, as shown in Figure 12-5:

Figure 12-5 MEP Rules



MEP manual priority list

All Satellite Gateways will choose entry point from the Center Gateways according to these priorities:

	Satellite Gateways	1st priority Center Gateways	2nd priority Center Gateways	3rd priority Center Gateways
	* Any	* Any	None	None

Default MEP Rule

Exception priority rules:

	Satellite Gateways	1st priority Center Gateways	2nd priority Center Gateways	3rd priority Center Gateways
1	S1	M1	M2	M3
2	S2	M2	M3	None

First MEP Rule

Second MEP Rule

The **MEP manual priority list** window is divided into the default rule, and rules which provide exceptions to the default rule. The default MEP rule takes effect when:

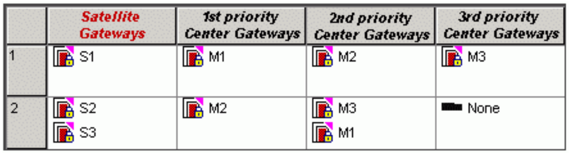
- No MEP rules are defined
- When the source of the connection cannot be found in the **Exception priority rules**

The **Exception priority rules** section contains three priority levels: primary, secondary, and tertiary. While there are only three priority levels,

- The same priority can be assigned to several central gateways
- The same rule can be assigned to several satellite gateways
- A priority level can be left blank

In Figure 12-6, in the second MEP rule, central gateways M3 and M1 have equal priority. The same rule is being applied to satellites S2 and S3.

Figure 12-6 Sample MEP rules



	Satellite Gateways	*1st priority Center Gateways*	*2nd priority Center Gateways*	*3rd priority Center Gateways*
1	S1	M1	M2	M3
2	S2 S3	M2	M3 M1	None

When more than one gateway is assigned the same priority level, which gateway will be chosen is resolved according to the **Advanced** settings. See "Advanced Settings" on page 239.

Advanced Settings

In some instances, more than one gateway is available in the center with no obvious priority between them. For example — as shown in Figure 12-6 — more than one gateway is assigned "second" priority. In this scenario, **Advanced** options are used to decide which gateway is chosen: *First to Respond*, or *Random Selection*. (Choose Random selection to enable load balancing between the gateways.)

When "manually set priority list" is the MEP selection mechanism, *RIM is supported.* RIM can be configured with "manually set priority list" because the "random selection" mechanism available on the **Advanced** button is different from the random selection mechanism used for MEP.

For the "random selection" mechanism employed for MEP, a different gateway is selected for each IP source/destination pair. For the random selection mechanism available from the **Advanced** button, a single MEP entry point is randomly selected and then used for all connections, and does not change according to source/destination pair. Load distribution is therefore achieved since every satellite gateway is randomly assigned a gateway as its entry point. This makes it possible to enable RIM at the same time.

Tracking

If the tracking option is enabled for MEP, the following information is logged by each satellite gateway:

- The resolved peer gateway (a gateway in the MEP)
- The priority of the resolved gateway (primary, secondary, tertiary)
- Whether the resolved gateway is responding

For example, in the scenario shown in Figure 12-4, satellite S1 opens a connection to the VPN domain that includes gateways M1, M2, and M3. M1 is the resolved peer. If tracking is enabled, the log reads:

```
Resolved peer for tunnel from S1 to the MEP that contains M1, M2,
and M3, is: M1 (Primary gateway, responding).
```

Implicit MEP

There are three methods to implement implicit MEP:

- *First to Respond*, in which the first gateway to reply to the peer gateway is chosen. An organization would choose this option if, for example, the organization has two gateways in a MEPed configuration - one in London, the other in New York. It makes sense for VPN-1 peers located in England to try the London gateway first and the NY gateway second. Being geographically closer to VPN peers in England, the London gateway is the first to respond, and becomes the entry point to the internal network. See: "First to Respond" on page 234.
- *Primary-Backup*, in which one or multiple backup gateways provide "high availability" for a primary gateway. The remote peer is configured to work with the primary gateway, but switches to the backup gateway if the primary goes down. An organization might decide to use this configuration if it has two machines in a MEP environment, one of which is stronger than the other. It makes sense to configure the stronger machine as the primary. Or perhaps both machines are the same in terms of strength of performance, but one has a cheaper or faster connection to the Internet. In this case, the machine with the better Internet connection should be configured as the primary. See: "Primary-Backup Gateways" on page 242.
- *Load Distribution*, in which the remote VPN peer randomly selects a gateway with which to open a connection. For each IP source/destination address pair, a new gateway is randomly selected. An organization might have a number of machines with equal performance abilities. In this case, it makes sense to enable load distribution. The machines are used in a random and equal way. See: "Random Selection" on page 236.

Implicit MEP is supported if the gateways with overlapping encryption domains are in the same community. If they are located in different communities, only one of the gateways will be used for this encryption domain.

Note - When upgrading from a version prior to NGX R60 where Implicit MEP was already configured, the settings previously configured will remain.

First to Respond

When there is no primary gateway, all gateways share “equal priority.” When all gateway’s share “equal priority”:

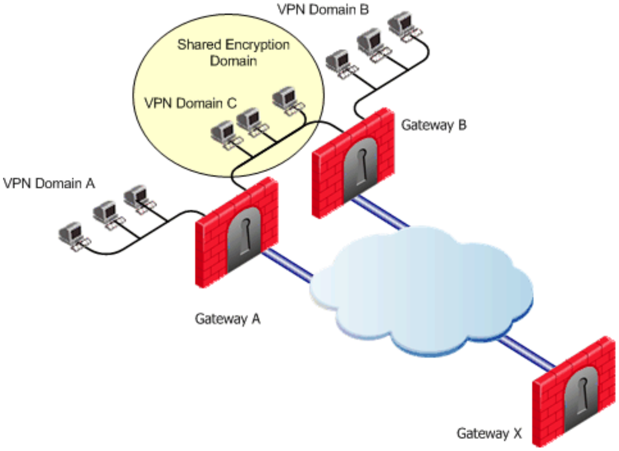
- Remote VPN peers send RDP packets to all the gateways in the MEP configuration.
- The first gateway to respond to the probing RDP packets gets chosen as the entry point to network. The idea behind *first to respond* is “proximity”. The gateway which is “closer” to the remote VPN peer responds first.
- A VPN tunnel is opened with the first to respond. All subsequent connections pass through the chosen gateway.
- If the gateway ceases to respond, a new gateway is chosen.

In a star community, RDP packets are sent to the gateways and the first to respond is used for routing only when:

1. There is more than one center gateway, **and**
2. One of the following VPN routing options was selected:
   - **To center and to other satellites through center**
   - **To center, or through the center to other satellites, to internet and other VPN targets**

   This setting is found on the **Community Properties > VPN Advanced > VPN Routing** page.

Figure 12-7 Implicit MEP



In this scenario:

- MEP is **not** enabled on the community

- First to respond method is used
- Gateway X accesses VPN domain A through gateway A
- Gateway X accesses VPN domain B through gateway B
- Gateway X accesses VPN domain C through gateway A or B

In a star community, RDP packets are sent to the gateways and the first to respond is used for routing when:

1. There is more than one center gateway, and
2. One of the following VPN routing options was selected:
   - **To center and to other satellites through center**
   - **To center, or through the center to other satellites, to internet and other VPN targets**

   This setting is found on the **Community Properties > VPN Advanced > VPN Routing** page.

Primary-Backup Gateways

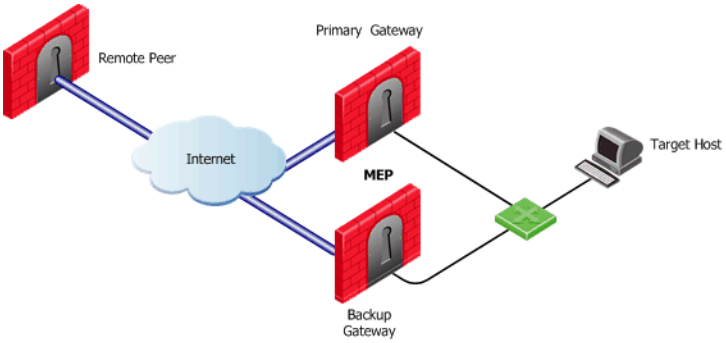
Backup gateways provide redundancy for primary gateways. If the primary gateway fails, connections go through the backup.

In Figure 12-8, the first gateway is configured as the "primary," and the second gateway as the "backup." If the primary gateway fails, for whatever reason, the remote VPN peer detects that the link has gone down and works through the backup gateway. The backup gateway inherits the complete VPN domain of the primary. Failover within an existing connection is not supported; the current connection is lost.

When the primary gateway is restored, new connections go through the primary gateway while connections that already exist will continue to work through the backup gateway.

Note - When using the Primary-Backup gateways method, the encryption domains should not overlap

Figure 12-8 MEP configuration with Primary Gateway defined



Load Distribution

To prevent any one gateway from being flooded with connections, the connections can be evenly shared amongst all the gateways to distribute the load. When all gateways share equal priority (no primary) and are MEPed to the *same* VPN domain, it is possible to enable load distribution between the gateways. The gateways are probed with RDP packets, as in all other MEP configurations, to create a list of responding gateways. A gateway is randomly chosen from the list of responding gateways. If a gateways stops responding, a new gateway is (randomly) chosen.

A new gateway is randomly selected for every source/destination IP pair. While the source and destination IP's remain the same, the connection continues through the chosen gateway.

Routing Return Packets

To make sure return packets are routed correctly, the MEPed gateway can make use of either:

- IP pool NAT (which means static NAT) or
- Route Injection Mechanism

IP Pool Network Address Translation (NAT)

IP pool NAT is a type of NAT in which source IP addresses from remote VPN domains are mapped to an IP address drawing from a pool of registered IP addresses. In order to maintain symmetric sessions using MEPed gateways, the MEPed gateway performs NAT using a range of IP addresses dedicated to that specific gateway and should be routed within the internal network to the originating gateway. When the returning packets reach the gateway, the gateway restores the original source IP address and forwards the packets to the source.

RIM

Route Injection Mechanism (RIM) enables a security gateway to use a dynamic routing protocol to propagate the encryption domain of a VPN peer gateway to the internal network. When a VPN tunnel is created, RIM updates the local routing table of the security gateway to include the encryption domain of the VPN peer.

When a tunnel to a MEPed gateway goes down, the gateway removes the appropriate “return route” from its own local routing table. This change is then distributed backwards to the routers behind the gateway.

RIM is based both on the ability of the gateway to update its local routing table, and the presence of the a dynamic routing protocol to distribute the change to the network behind the gateway. There is little sense in enabling RIM on the gateway if a dynamic routing protocol is not available to distribute changes.

When MEP is enabled, RIM can be enabled only if permanent tunnels are enabled for the whole community. In a MEP configuration RIM is available when using the *First to Respond*, *Manual set priority list,* and *VPN Domain* mechanisms. In the first two options, satellite gateways "see" the center gateways as unified as if one tunnel is connecting them. As a result, only the chosen MEP gateway will inject the routes. In *VPN Domain* MEP, it could be that all MEP gateways will inject the routes, which requires configuring the routers behind the MEP gateways to return packets to the correct gateway.

RIM is not available when *Random Selection* is the selected entry point mechanism.

For more information on RIM, see "Route Injection Mechanism" on page 177.

Special Considerations

1. If one of the central gateways is an externally managed gateway:
    - The VPN domain of the central gateways will not be automatically inherited by an externally managed gateway
    - The RIM configuration will not be automatically downloaded
2. UTM-1 Edge gateways cannot be configured as a MEP gateway but can connect to MEPed gateways.
3. DAIP gateways require DNS resolving in order to be configured as a MEP gateway.

Configuring MEP

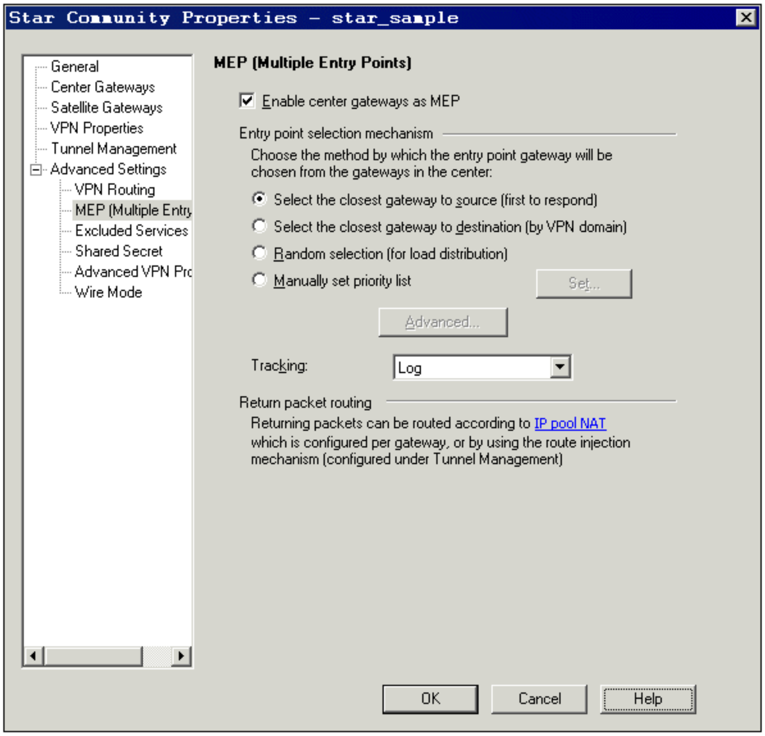
To configure MEP, decide on:

1. The MEP method
   - Explicit MEP - See “Explicit MEP” on page 232.
   - Implicit MEP - See “Implicit MEP” on page 240.
2. If required, method for returning reply packets:
   - IP pool NAT
   - RIM - To configure RIM, see “Configuring RIM” on page 186.

Configuring Explicit MEP

Explicit MEP is only available in Site-to-Site Star VPN communities where multiple central gateways are defined. To configure MEP:

1. Open the **Star Community properties page > Advanced Settings > MEP (Multiple Entry Point)**: Select **Enable center gateways as MEP**.



MEP (Multiple Entry Points)

☑ Enable center gateways as MEP

Entry point selection mechanism

Choose the method by which the entry point gateway will be chosen from the gateways in the center:

● Select the closest gateway to source (first to respond)

○ Select the closest gateway to destination (by VPN domain)

○ Random selection (for load distribution)

○ Manually set priority list

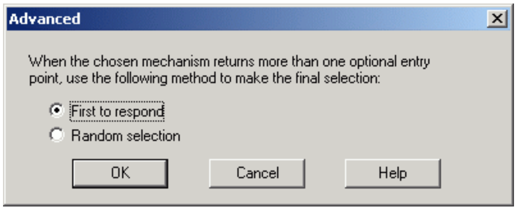
Tracking: Log

Return packet routing

Returning packets can be routed according to IP pool NAT which is configured per gateway, or by using the route injection mechanism (configured under Tunnel Management)

2. Select an entry point mechanism:
   - First to respond
   - By VPN domain
   - Random selection
   - Manual priority list

If “By VPN domain” or “Manually set priority list” is selected, click **Advanced** to resolve how more than one gateway with equal priority should be selected.



When the chosen mechanism returns more than one optional entry point, use the following method to make the final selection:

● First to respond

○ Random selection

If “Manually set priority list” is selected, click **Set** to create a series of MEP rules.

3. Select a tracking option, if required.

Configuring Implicit MEP

First to Respond in an Overlapping Encryption Domain

When more than one gateway leads to the same (overlapping) VPN domain, they are considered MEPed by the remote VPN peer, and the first gateway to respond to the probing protocol is chosen. To configure *first to respond*, define that part of the network that is shared by all the gateways into a single group and assign that group as the VPN domain.

To display all overlapping encryption domains, use the `vpn overlap_encdom` command. For more information, see “VPN Command Line Interface” on page 711.

On the **Properties** window of each gateway network object, **Topology** page > **VPN Domain** section, select **Manually defined**, and define a VPN domain for all gateways (some of which will be overlapping).

Primary-Backup

1. In the **Global Properties** window, **VPN > Advanced** page, select **Enable Backup Gateway**.
2. In the network objects tree, **Groups** section, create a group consisting of gateways that act as backup gateways.
3. On the **Properties** window of the network object selected as the Primary gateway, **VPN** page, select **Use Backup Gateways**, and select the group of backup gateways from the drop-down box. This gateway now functions as the primary gateway for a specific VPN domain.
4. Define the VPN for the backup gateway(s). Backup gateways do not always have a VPN domain of their own. They simply back-up the primary. If the backup gateway does not have a VPN domain of its own, the VPN domain should include only the backup gateway itself:

    a. On the **Properties** window of the backup network object, **Topology** page **> VPN Domain** section, select **Manually defined**.

    b. Select a group or network that contains only the backup gateway.

    If the backup *does* have a VPN domain:

    a. Verify that the IP address of the backup gateway is *not* included in the VPN domain of the primary.

    b. For each backup gateway, define a VPN domain that does *not* overlap with the VPN domain of any other backup gateway.

Note - There must be no overlap between the VPN domain of the primary gateway and the VPN domain of the backup gateway(s); that is, no IP address can belong to both.

5. If required, configure IP pool NAT or RIM to handle return packets. To configure IP pool NAT, see “Configuring IP Pool NAT” on page 250. To configure RIM, see “Configuring RIM” on page 186.

Load Distribution

1. In the **Global Properties** window, **VPN > Advanced** page, select **Enable load distribution for Multiple Entry Point configurations (Site to Site connections)**. Checking this options also means that in MEP configurations, the remote VPN peer randomly selects a MEPed gateway to work with.
2. Define the same VPN domain for all the gateways.

Configuring IP Pool NAT

To configure IP pool NAT:

1. In **Global Properties > NAT** page, select **Enable IP Pool NAT**.
2. Set tracking options for address exhaustion and for address allocation and release. Then:
3. For each gateway, create a network object that represents the IP pool NAT addresses for that gateway. The IP pool can be a network, group, or address range. For example:
   - On the network objects tree, right-click **Network Objects** branch **> New > Address Range...** The Address Range Properties window opens.
   - On the **General** tab, enter the first IP and last IP of the address range.
   - Click **OK**. In the network objects tree, **Address Ranges** branch, the new address range appears.
4. On the gateway object where IP pool NAT translation is performed, **Gateway Properties** window, **NAT > IP Pool NAT** page, select either
   - **Allocate IP Addresses** from, and select the address range you created, OR
   - **Define IP Pool addresses on gateway interfaces**. If you choose this option, you need to define the IP Pool on each required interface, in the **Interface Properties** window, **IP Pool NAT** tab.
5. In the **IP Pool NAT** page, select either (or all):
   - **Use IP Pool NAT for VPN clients connections**
   - **Use IP Pool NAT for gateway to gateway connections**
   - **Prefer IP Pool NAT over Hide NAT**
6. Click **Advanced...**
   - Decide after how many minutes unused addressees are returned to the IP pool.
   - Click **OK** twice.
7. Edit the routing table of each internal router, so that packets with an a IP address assigned from the NAT pool are routed to the appropriate gateway.

Chapter 13
Traditional Mode VPNs

In This Chapter

Introduction to Traditional Mode VPNs

Simplified Mode makes it possible to maintain and create simpler, and therefore less error prone and more secure VPNs. It also makes it easier to understand the VPN topology of an organization, and to understand who is allowed to communicate with who. In addition, new VPN features such as VPN routing are supported only with a Simplified Mode Security Policy.

However, organizations that have large VPN deployments with complex networks may prefer to maintain existing VPN definitions and continue to work within Traditional Mode until they are able to migrate their policies to Simplified Mode.

For guidelines on how to convert Traditional Mode VPNs to Simplified Mode, see "Converting a Traditional Policy to a Community Based Policy" on page 717.

VPN Domains and Encryption Rules

Figure 13-1 shows a VPN between gateways, and the VPN Domain of each gateway. Net_A and Net_B are the VPN Domain of gateway 1, Net_D is the VPN Domain of gateway 2, and Net_E is the VPN Domain of gateway 3.

Figure 13-1 A VPN between Gateways, and the Encryption (VPN) Domain of each Gateway

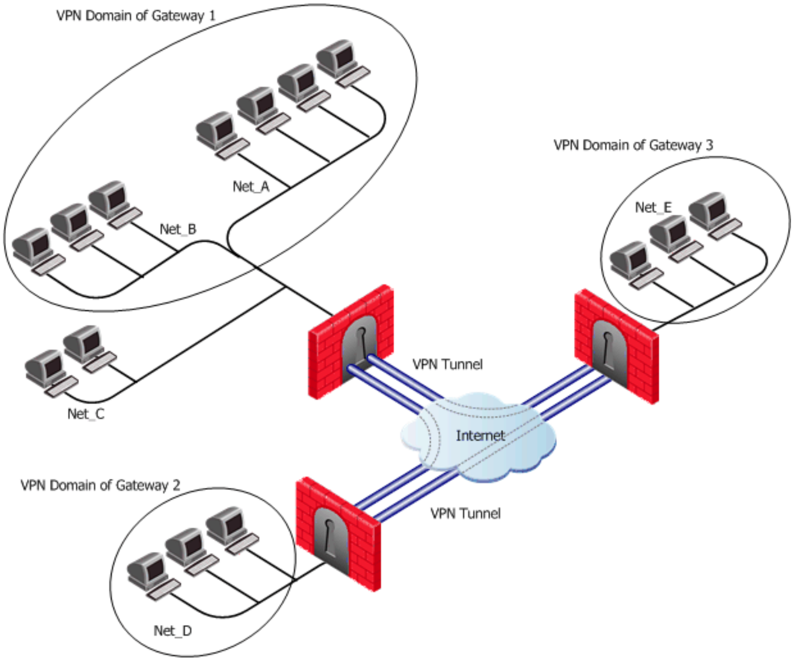


Table 13-1 shows how the VPN is implemented in a rule. In Traditional VPN Mode, a single rule with the Encrypt rule action, deals with both access control and encryption.

Table 13-1 An example Encrypt rule in a Traditional Rule Base

Source	Destination	Service	Action	Track	Install On
Net_A Net_E	Net_A Net_E	My_Services	Encrypt	Log	Gateway 1 Gateway 3

A connection that matches an Encrypt rule is *encrypted* (or *decrypted*) and forwarded by the gateways enforcing the policy.

Sometimes, a connection may match the encrypt rule, but will not be encrypted. Consider the following rule:

Table 13-2 An Encrypt rule where encryption does not take place

Source	Destination	Service	Action	Track	Install On
X	Y	My_Services	Encrypt	Log	Policy Targets

1. If the source or the destination are behind the security gateway, but are not in the VPN Domain of the gateway, the connection is *dropped*.

   For example, referring to Figure 13-1 and Table 13-2, if Source X is in Net_C and Destination Y is in Net_D, gateway 1 drops the connection. This is because the Action says Encrypt but the connection cannot be encrypted because the source is not in the VPN Domain of gateway 1.

2. If the source and destination are inside the VPN Domain of the same gateway. In this case, the connection is *accepted in the clear*.

   For example, referring to Figure 13-1 and Table 13-2, if Source X is in Net_A and Destination Y is in Net_B, the connection originates at X and reaches the gateway, which forwards the response back to Y. The connection is not encrypted because there is no peer gateway for Y that could decrypt the connection. A SmartView Tracker log is issued "`Both endpoints are in the Encryption Domain`".

Defining VPN Properties

It is possible to use different encryption methods between the same gateways. Different connections between two gateways can be encrypted using different methods. This is because different IKE phase 2 properties can be defined per Encrypt rule.

IKE Phase 1 properties are defined per gateway.

Internally and Externally Managed Gateways

The gateways at each end of a VPN tunnel can be managed by the same Security Management server or by different Security Management servers. A security gateway that is managed by the Security Management server is called an internal gateway. If it is managed by a different Security Management server it is called an external gateway.

If the peer security gateway is external, you must obtain certain details about that gateway from the peer administrator, and configure them in SmartDashboard.

Considerations for VPN Creation

There are many ways of setting up a VPN. Before starting, a number of issues need to be considered, such as:

Choosing the Authentication Method

Before security gateways can create a VPN tunnel, they need to authenticate to each other. This authentication is performed either by means of certificates or with a pre-shared secret. Certificates are considered to be a stronger form of authentication.

Choosing the Certificate Authority

If the gateways use certificates, the certificates can be issued either by the Internal Certificate Authority (ICA) on the Security Management server, or by a third party OPSEC certified CA.

The Internal CA makes it very easy to use PKI for Check Point applications such as site-to-site and remote access VPNs. However, an administrator may prefer to continue using a CA that is already used within the organization, for generalized applications such as secure email, and disk encryption.

If the gateways are both internally managed and use certificates for authentication, the easiest strategy is for both gateways to present a certificate signed by the Internal CA.

Configuring Traditional Mode VPNs

In This Section

Editing a Traditional Mode Policy

An existing Traditional Mode policy will open in Traditional Mode. To start a new Traditional Mode policy, proceed as follows.

1. In the **Global Properties** window, **VPN** page, select either **Traditional mode to all new Security Policies** or **Traditional or Simplified per new Security Policy**, and save the policy.

Assuming you selected **Traditional or Simplified per new Security Policy**:

2. From the **File** menu, select **New**. The **New Policy Package** window opens.
3. Give the new policy package a name.
4. Select **Security and Address Translation**.
5. In the VPN configuration method area, select **Traditional mode** and click **OK.**

In the Security Policy Rule Base, notice that one of the available Actions is **Encrypt.**

Configuring VPN Between Internal Gateways using ICA Certificates

Defining the Gateways

1. For each gateway that is to be part of the VPN define a Check Point gateway object. In the Network Objects tree, right click and select **New > Check Point > Gateway...**.
2. In the **General Properties** page of the Check Point gateway object, select **VPN**.
3. In the **Communication** window, establish Secure Internal Communication.
4. In the **Topology** page, define the IP address, network mask, and anti-spoofing for every gateway interface
5. Still on the **Topology** page, define the **VPN Domain**. select either:
   - **All IP Addresses behind gateway based on Topology information** or
   - **Manually defined**. Either select an existing network or group from the drop-down list or create a new group of machines or networks by clicking **New...**
6. In the **VPN** page, **Certificate List** area, **Add** a certificate issued by the ICA.
7. Still on the **VPN** page, click **Traditional mode configuration**. The **Traditional mode IKE properties** window opens.
   - In the **Support authentication methods** area, select **Public Key Signatures**. To specify that the gateway will only use certificates issued by the ICA, click **Specify** and select the ICA.
   - Select IKE Phase 1 encryption and data integrity methods or accept the checked defaults.

Defining the Encrypt Rule

8. In the Security Rule Base, define the Encrypt rule(s).
9. If you wish to change the IKE Phase 2 properties for this rule, double click the **Encrypt** action and make the required changes.

VPN Between Internal Gateways Using Third Party CA Certificates

1. Obtain the CA certificate, and define the Certificate Authority (CA) object. For details, see “Enrolling with a Certificate Authority” on page 107.

Defining the Gateways

2. Define the Check Point gateway object. In the Network Objects tree, right click and select **New > Check Point > Gateway...**.
3. In the **General Properties** page, select either **VPN**.
4. In the **Communication** window, establish Secure Internal Communication.
5. In the **Topology** page, define the IP address, network mask, and anti-spoofing for every gateway interface.
6. Still on the **Topology** page, define the **VPN Domain**. select either:
   - **All IP Addresses behind gateway based on Topology information** or
   - **Manually defined**. Either select an existing network or group from the drop-down list, or create a new network or group by clicking **New...**.
7. In the **VPN** page, **Certificate List** area, **Add** a certificate issued by the certificate authority defined in step 1. For details, see “Enrolling with a Certificate Authority” on page 107.
8. Still on the **VPN** page, click **Traditional mode configuration**. The **Traditional mode IKE properties** window opens.
   - In the **Support authentication methods** area, select **Public Key Signatures**. To specify that the gateway will only use certificates issued by the CA specified in step 1, click **Specify** and select the CA.
   - Select IKE Phase 1 encryption and data integrity methods or accept the checked defaults.
9. Repeat step 2 to step 8 for each gateway taking part in the VPN.

Defining the Encrypt Rule

10. In the Security Rule Base, define the Encrypt rule(s).
11. If you wish to change the IKE Phase 2 properties for this rule, double click the **Encrypt** action and make the required changes.

Configuring VPN with Externally Managed Gateways Using Certificates

Obtain Information from the Peer Administrator

Obtain the gateway topology and VPN Domain information about the externally managed gateways from the peer administrator.

You must also agree on authentication, encryption and data integrity methods for the VPN.

You must also obtain the CA certificate of the peer, either from the peer administrator or directly from the peer CA.

Defining the CAs

1. Obtain the CA certificate and create the Certificate Authority (CA) object for the internally managed gateways. For details, see "Enrolling with a Certificate Authority" on page 107.
2. Define the CA object for the externally managed gateways, and configure it using the peer CA certificate.

Defining the Internally Managed Gateways

3. Create the Check Point gateway object. In the Network Objects tree, right click and select **New > Check Point > Gateway...**.
4. In the **General Properties** page, select either **VPN**.
5. In the **Communication** window, establish Secure Internal Communication.
6. In the **Topology** page, define the IP address, network mask, and anti-spoofing for every gateway interface.
7. Still on the **Topology** page, define the **VPN Domain**. select either:
   - **All IP Addresses behind gateway based on Topology information** or
   - **Manually defined**. Either select an existing network or group from the drop-down list, or create a new network or group by clicking **New...**.
8. In the **VPN** page, **Certificate List** area, **Add** a certificate issued by the certificate authority defined in step 1. For details, see "Enrolling with a Certificate Authority" on page 107.
9. Still on the **VPN** page, click **Traditional mode configuration**. The **Traditional mode IKE properties** window opens.

- In the **Support authentication methods** area, select **Public Key Signatures**. To specify that the gateway will only use certificates issued by the CA specified in step 1, click **Specify** and select the CA.
- Select IKE Phase 1 encryption and data integrity methods or accept the checked defaults.

10. Repeat step 3 to step 9 for each internally managed gateway.

Defining the Externally Managed Gateways

11. Create the externally managed gateway object:
    - If it is a Check Point gateway, in the **Network Objects** tree, right click and select **New > Check Point > Externally Managed Gateway...**.
    - If it is not a Check Point gateway, select **Manage > Network Objects.. .> New...> Interoperable Device...**.
12. For an external Check Point gateway only: In the **General Properties** page, select **VPN**.
13. Using the topology information supplied by the peer administrator, in the **Topology** page, manually define the IP address and network mask for every gateway interface.
14. Using the VPN Domain information supplied by the peer administrator, define the VPN domain in the **VPN Domain** section of the Topology page. Either select **All IP Addresses behind gateway based on Topology information** or manually define a group of machines or a network and set them as the VPN domain.
15. On the **VPN** page, click **Traditional mode configuration**. The **Traditional mode IKE properties window** opens.
    - Select IKE Phase 1 encryption and integrity methods (in coordination with the peer gateway's administrator) or accept the defaults.
    - In the **Support authentication methods** area, select **Public Key signatures**.
16. On the VPN page, click **Matching Criteria...**. The **Certificate Matching Criteria** window opens. The configurations settings in this window force the externally managed gateway to present a certificate from a defined CA, and require that the details on the certificate match those specified here. This is enforced by the internally managed gateways during IKE negotiation.

Defining the Encrypt Rule

17. In the Security Rule Base, define the Encrypt rule(s).
18. If you wish to change the IKE Phase 2 properties for this rule, double click the **Encrypt** action and make the required changes.

Configuring a VPN using a Pre-Shared Secret

When using a pre-shared secret to authenticate gateways, you need to enable each gateway in the VPN for pre-shared secrets. Then, on each gateway, define a pre-shared secret for each of the other gateways. However, for each pair of gateways, you only need to define the pre-shared secrets for the pair on one of the gateways.

Note - A pre-shared secret defined for externally managed VPN modules in the community is not supported if one of the internally managed members is of version earlier than NG FP3.

For example, in a VPN with four gateways, A,B, C and D, there will be six secrets: A-B, A-C, A-D, B-C, B-D and C-D.

- On A define the secrets for B, C and D.
- On B define the secrets for C and D.
- On C define the secret for D.

The following procedure applies to both internal and external security gateways. When working with externally managed gateways, the administrator of the peer external gateways must configure his or her gateways appropriately.

Obtain Information from the Peer Administrator

If working with externally managed gateways, obtain from the peer administrator the external gateway topology and VPN Domain information.

You must also agree on the pre-shared secrets, and on authentication, encryption and data integrity methods for the VPN.

Defining the Gateways

1. Define the gateway object.
   - If the gateway is an internal gateway, define a Check Point gateway object. In the Network Objects tree, right click and select **New > Check Point > Gateway...**.
   - If the gateway is externally managed:
     - If it is a Check Point gateway, In the **Network Objects** tree, right click and select **New > Check Point > Externally Managed Gateway...**.
     - If it is not a Check Point gateway, select **Manage > Network Objects... > New... > Interoperable Device...**.
2. For an internally managed gateway or for a Check Point externally managed gateway, in the **General Properties** page of the gateway object, select **VPN**.
3. For an internally managed gateway only, in the **Communication** window, establish Secure Internal Communication.
4. In the **Topology** page, define the IP address, network mask, and anti-spoofing for every gateway interface
5. Still on the **Topology** page, define the **VPN Domain**. select either:
   - **All IP Addresses behind gateway based on Topology information** or
   - **Manually defined**. Either select an existing network or group from the drop-down list or create a new group of machines or networks by clicking **New...**
6. In the **VPN** page, click **Traditional mode configuration**. The **Traditional mode IKE properties** window opens.
   - In the **Support authentication methods** area, select **Pre-shared Secret**, click **Edit Secrets...**. Only peer gateways which support pre-shared secrets appear in the list.
   - Type a secret for each peer gateway.
   - Select IKE phase 1 encryption and data integrity methods or accept the checked defaults.
7. Repeat step 1 to step 6 for each gateway taking part in the VPN.

Defining the Encrypt Rule

8. In the Security Rule Base, define the Encrypt rule(s).
9. If you wish to change the IKE Phase 2 properties for this rule, double click the **Encrypt** action and make the required changes.

Remote Access VPN

Chapter 14

Introduction to Remote Access VPN

In This Chapter

Need for Remote Access VPN

Whenever users access the organization from remote locations, it is essential that the usual requirements of secure connectivity be met but also the special demands of remote clients, for example:

- The IP of a remote access client might be unknown.
- The remote access client might be connected to a corporate LAN during the working day and connected to a hotel LAN during the evening, perhaps hidden behind some kind of NATing device.
- The remote client might need to connect to the corporate LAN via a wireless access point.
- Typically, when a remote client user is out of the office, they are not protected by the current security policy; the remote access client is both exposed to Internet threats, and can provide a way into the corporate network if an attack goes through the client.

To resolve these issues, a security framework is needed that ensures remote access to the network is properly secured.

The Check Point Solution for Remote Access

In This Section

VPN-1 SecuRemote — Check Point's Remote Access VPN solution — enables you to create a VPN tunnel between a remote user and your organization's internal network. The VPN tunnel guarantees:

- Authenticity, by using standard authentication methods
- Privacy, by encrypting data
- Integrity, by using industry-standard integrity assurance methods

SecuRemote/SecureClient extends VPN functionality to remote users, enabling users to securely communicate sensitive information to networks and servers over the VPN tunnel, using LAN, wireless LAN and various dial-up (including broadband) connections. Users are managed either in the internal database of the security gateway or via an external LDAP server.

After a SecuRemote user is authenticated, a transparent secured connection is established.

SecuRemote works with:

- Check Point Security Gateways.
- VPN-1 and UTM-1 Edge gateways

Enhancing SecuRemote with SecureClient Extensions

SecureClient is a remote access client that includes and extends SecuRemote by adding a number of features:

- Security features
- Connectivity features
- Management features

Security Features

- A Desktop Security Policy. See: "Desktop Security" on page 489.
- Logging and Alerts
- Secure Configuration Verification (SCV); (see: "Secure Configuration Verification" on page 519)

Connectivity Features

- Office mode addresses (see: "Office Mode" on page 299).
- Visitor mode (see: "Resolving Connectivity Issues" on page 675.)
- Hub mode. (see: "Hub Mode (VPN Routing for Remote Clients)" on page 560.)

Management Features

- Automatic software distribution. (see: "Packaging SecureClient" on page 479.)
- Advanced packaging and distribution options (see: "Packaging SecureClient" on page 479.)
- Diagnostic tools

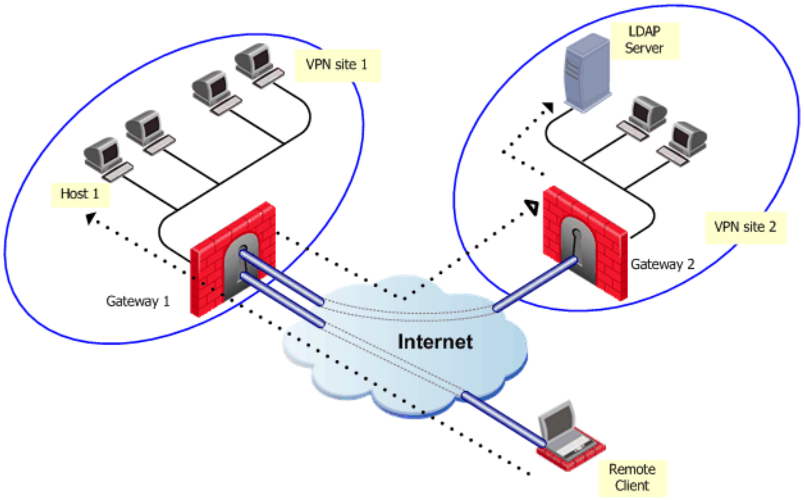
Establishing a Connection Between a Remote User and a Gateway

To allow the user to access a network resource protected by a security gateway, a VPN tunnel establishment process is initiated. An IKE (Internet Key Exchange) negotiation takes place between the peers.

During IKE negotiation, the peers' identities are authenticated. The gateway verifies the user's identity and the client verifies that of the gateway. The authentication can be performed using several methods, including digital certificates issued by the Internal Certificate Authority (ICA). It is also possible to authenticate using third-party PKI solutions, pre-shared secrets or third party authentication methods (for example, SecurID, RADIUS *etc*.).

After the IKE negotiation ends successfully, a secure connection (a VPN tunnel) is established between the client and the gateway. All connections between the client and the gateway's VPN domain (the LAN behind the gateway) are encrypted inside this VPN tunnel, using the IPSec standard. Except for when the user is asked to authenticate in some manner, the VPN establishment process is transparent.

Figure 14-1 Remote to Gateway



In Figure 14-1, the remote user initiates a connection to gateway 1. User management is not performed via the VPN database, but an LDAP server belonging to VPN Site 2. Authentication takes place during the IKE negotiation. Gateway 1 verifies that the user exists by querying the LDAP server behind gateway 2. Once the user's existence is verified, the gateway then authenticates the user, for example by validating the user's certificate. Once IKE is successfully completed, a tunnel is created; the remote client connects to Host 1.

If the client is behind the gateway (for example, if the user is accessing the corporate LAN from a company office), connections from the client to destinations that are also behind the LAN gateway are not encrypted.

Remote Access Community

A Check Point Remote Access community enables you to quickly configure a VPN between a group of remote users and one or more security gateways. A Remote Access community is a virtual entity that defines secure communications between security gateways and remote users. All communications between the remote users and the gateways' VPN domains are secured (authenticated and encrypted) according to the parameters defined for Remote Access communications in SmartDashboard Global Properties.

Identifying Elements of the Network to the Remote Client

SecuRemote/SecureClient needs to know the elements of the organization's internal network before it can handle encrypted connections to and from network resources. These elements, known as a *topology*, are downloaded from any security gateway managed by the Security Management server.

A site's topology information includes IP addresses on the network and host addresses in the VPN domains of other gateways controlled by the same Security Management server. If a destination IP is inside the site's topology, the connection is passed in a VPN tunnel.

When the user creates a site, the client automatically contacts the site and downloads topology information and the various configuration properties defined by the administrator for the client. This connection is secured and authenticated using IKE over SSL. The site's topology has a validity timeout after which the client would download an updated topology. The network administrator can also configure an *automatic* topology update for remote clients. This requires no intervention by the user.

Connection Mode

The remote access clients connect with gateways using Connect mode.

During connect mode, the remote user deliberately initiates a VPN link to a specific gateway. Subsequent connections to any host behind other gateways will transparently initiate additional VPN links as required.

Connect mode offers:

- **Office mode**, to resolve routing issues between the client and the gateway. See, "Office Mode" on page 299.
- **Visitor mode**, for when the client needs to tunnel all client to gateway traffic through a regular TCP connection on port 443.
- **Routing all traffic through Gateway (Hub mode)**, to achieve higher levels of security and connectivity.
- **Auto connect**, when an application tries to open a connection to a host behind a gateway, the user is prompted to initiate a VPN link to that gateway. For example, when the e-mail client tries to access the IMAP server behind gateway X, SecureClient prompts the user to initiate a tunnel to that gateway.
- **User profiles (Location Profiles).** See: "User Profiles" on page 275.

User Profiles

Mobile users are faced with a variety of connectivity issues. During the morning they find themselves connected to the LAN of a partner company; during the evening, behind some kind of NATing device employed by the hotel where they are staying.

Different user profiles are used to overcome changing connectivity conditions. Users create their own profiles, or the network administrator creates a number of profiles for them. If the administrator creates a profile, the profile is downloaded to the client when the user updates the site topology. The user selects which profile to work with from a list. For example, a profile that enables UDP encapsulation in order to cope with some NATing device, or a profile that enables *Visitor mode* when the remote client must tunnel the VPN connection over port 443. The policy server used to download the Desktop Security Policy is also contained in the profile.

Access Control for Remote Access Community

Typically the administrator needs to define a set of rules that determines access control to and from the network. This is also true for remote access clients belonging to a remote access community. Policy rules must be created in order to control the way remote clients access the internal network via the gateway. (Membership of a community does not give automatic access to the network.)

The gateway's Security Policy Rule Base defines access control; in other words, whether a connection is allowed. Whether a connection is encrypted is determined by the community. If both the source and the destination belong to the community, the connection is encrypted; otherwise, it is not encrypted. For example, consider a rule that allows FTP connections. If a connection matching the rule is between community members, the connection is encrypted. If the connection is not between community members, the connection is not encrypted.

The gateway's Security Policy controls access to resources behind the gateway, protects the security gateway and the networks behind it. Since the remote client is not behind the gateway, it is not protected by the gateway's Security Policy. Remote access using SecureClient can be protected by a Desktop Security Policy. See "Desktop Security" on page 489.

Client-Gateway Authentication Schemes

Authentication is a key factor in establishing a secure communication channel among gateways and remote clients. Various authentication methods are available, for example:

- Digital certificates
- Pre-shared secrets
- Other authentication methods (made available via Hybrid mode)

Digital Certificates

Digital Certificates are the most recommended and managable method for authentication. Both parties present certificates as a means of proving their identity. Both parties verify that the peer's certificate is valid (i.e. that it was signed by a known and trusted CA, and that the certificate has not expired or been revoked).

Digital certificates are issued either by Check Point's Internal Certificate Authority or third-party PKI solutions. Check Point's ICA is tightly integrated with VPN and is the easiest way to configure a Remote Access VPN. The ICA can issue certificates both to security gateways (automatically) and to remote users (generated or initiated).

Using the ICA, generate a certificate and transfer it to the user "out-of-band." Alternatively, initiate the certificate generation process on Security Management server. The process is completed independently by the user. The administrator can also initiate a certificate generation on the ICA management tool (the only option available if users are defined on an LDAP server).

It is also possible to use third-party Certificate Authorities to create certificates for authentication between security gateways and remote users. The supported certificate formats are PKCS#12, CAPI, and *Entrust*.

Users can also be provided with a hardware token for storing certificates. This option offers the advantage of higher level of security, since the private key resides only on the hardware token.

As part of the certificate validation process during the IKE negotiation, both the client and the gateway check the peer's certificate against the *Certificate Revocation List* (CRL) published by the CA which issued the certificate. If the client is unable to retrieve a CRL, the gateway retrieves the CRL on the client's behalf and transfers the CRL to the client during the IKE negotiation (the CRL is digitally signed by the CA for security).

Pre-Shared Secret

This authentication method has the advantage of simplicity, but it is less secure than certificates. Both parties agree upon a password before establishing the VPN. The password is exchanged “out-of-band”, and reused multiple times. During the authentication process, both the client and gateway verify that the other party knows the agreed-upon password.

Note - Passwords configured in the pre-shared secret tab are used in hybrid mode IKE and not in pre-shared secret mode. Pre-shared secret IKE mode is used for working with 4.1 Clients.

Other Authentication Methods Available via Hybrid Mode

Different organizations employing various means of user authentication may wish to utilize these means for remote access. Hybrid mode is an IKE mode that supports an asymmetrical way of authentication to address this requirement. Using Hybrid mode, the user employs one of the methods listed below to authenticate to the gateway. In return, the gateway authenticates itself to the client using strong, certificate-based authentication. Authentication methods which can be used in Hybrid mode are all those supported for normal user authentication in VPN, namely:

- **One Time Password** — The user is challenged to enter the number displayed on the Security Dynamics SecurID card. There are no scheme-specific parameters for the SecurID authentication scheme. The VPN module acts as an ACE/Agent 5.0. For agent configuration.

  SoftID (a software version of RSA's SecurID) and various other One Time Password cards and USB tokens are also supported.
- **Security Gateway - Password** — The user is challenged to enter his or her password held on the gateway.
- **OS Password** — The user is challenged to enter his or her Operating System password.
- **RADIUS** — The user is challenged for the correct response, as defined by the RADIUS server.
- **TACACS** — The user is challenged for the correct response, as defined by the TACACS or TACACS+ server.
- **SAA**. SAA is an OPSEC API extension to SecuRemote/SecureClient that enables third party authentication methods, such as biometrics, to be used with SecuRemote/SecureClient.

For additional information regarding authentication methods that are not based on certificates or pre-shared secrets see: The *Authentication* chapter in the *FireWall Administration guide*.

Advanced Features

Remote Access VPN supports other advanced features such as:

- Resolving connectivity and routing issues. See: "Office Mode" on page 299, and "Resolving Connectivity Issues" on page 675.
- IP-per-user/group.
- L2TP clients.

Alternatives to SecuRemote/SecureClient

To avoid the overhead of installing and maintaining client software, Check Point also provides the SSL Network Extender, a simple-to-implement thin client installed on the user's machine via a web browser. The browser connects to an SSL enabled web server and downloads the thin client as an ActiveX component. Installation is automatic.

VPN for Remote Access Considerations

In This Section

When designing Remote Access VPN, consider the following issues:

Policy Definition for Remote Access

There must be a rule in the Security Policy Rule Base that grants remote users access to the LAN. Consider which services are allowed. Restrict those services that need to be restricted with an explicit rule in the Security Policy Rule Base.

User Certificate Creation Methods when Using the ICA

Check Point's Internal Certificate Authority (ICA) offers two ways to create and transfer certificates to remote users:

1. The administrator **generates** a certificate in Security Management server for the remote user, saves it to removable media and transfers it to the client "out-of-band."
2. The administrator **initiates** the certificate process on the Security Management server (or ICA management tool), and is given a registration key. The administrator transfers the registration key to the user "out-of-band." The client establishes an SSL connection to the ICA (using the CMC protocol) and completes the certificate generation process using the registraion key. In this way:
   - Private keys are generated on the client.
   - The created certificate can be stored as a file on the machines hard-drive, on a CAPI storage device, or on a hardware token.

     This method is especially suitable for geographically spaced-remote users.

Internal User Database vs. External User Database

Remote Access functionality includes a flexible user management scheme. Users are managed in a number of ways:

- INTERNAL - a security gateway can store a static password in its local user database for each user configured in Security Management server. No additional software is needed.
- LDAP - LDAP is an open industry standard that is used by multiple vendors. Check Point products are compliant with LDAP technology. This compliancy enables:
    - Users to be managed externally by an LDP server.
    - The Enforcement modules to retrieve CRLs.
    - User information from other applications gathered in the LDAP users database, to be shared by many different applications. The security gateway consults the user information for authentication purposes.
- RADIUS - Remote Authentication Dial-In User Service (RADIUS) is an external authentication scheme that provides security and scalability by separating the authentication function from the access server.

  When employing RADIUS as an authentication scheme, the security gateway forwards authentication requests by remote users to the RADIUS server. The RADIUS server, which stores user account information, authenticates the users. The RADIUS protocol uses UDP for communications with the gateway. RADIUS Servers and RADIUS Server Group objects are defined in SmartDashboard.
- SecurID Token Management ACE/Server - Developed by RSA Security, SecurID requires users to both possess a token authenticator and to supply a PIN or password. Token authenticators generate one-time passwords that are synchronized to an RSA ACE/Server, and may come in the form of hardware or software. Hardware tokens are key-ring or credit card-sized devices, while software tokens reside on the PC or device from which the user wants to authenticate. All tokens generate a random, one-time-use access code that changes every minute or so. When a user attempts to authenticate to a protected resource, that one-time-use code must be validated by the ACE/Server.

  When employing SecurID as an authentication scheme, the security gateway forwards authentication requests by remote users to the ACE/Server. ACE manages the database of RSA users and their assigned hard or soft tokens. The

VPN module acts as an ACE/Agent 5.0, which means that it directs all access requests to the RSA ACE/Server for authentication. For agent configuration see ACE/Server documentation.

There are two main difference between user management on the internal database, and user management on a SmartDirectory (LDAP) server. Firstly, user management in the SmartDirectory (LDAP) server is done externally and not locally. Secondly, on a SmartDirectory (LDAP) server templates can be modified and applied to users dynamically. This means that user definitions are easy to change and to manage; and changes are instantaneous or “live”. Changes that are applied to a SmartDirectory (LDAP) template are reflected immediately for all users who are using that template.

NT Group/RADIUS Class Authentication Feature

Authentication can take place according to NT groups or RADIUS classes. In this way, remote access users are authenticated according to the remote access community group they belong to.

Note - Only NT groups are supported, not Active Directory.

VPN for Remote Access Configuration

In This Section:

The following configuration assumes you are working in the *Simplified mode*. If not, go to **Policy > Global Properties >VPN**, select **Simplified mode to all new Security Policies** and create a new Security Policy.

Establishing Remote Access VPN requires configuration on both the gateway side (via Security Management server) and remote user side.

For the Gateway side, the administrator needs to:

1. Define the gateway
2. Decide how to manage users
3. Configure the VPN community and its participants
4. Set appropriate access control rules in the Security Policy Rule Base

5. Install the policy on the gateway

On the remote client side, the user needs to:

1. Define a site
2. Register to the internal CA to receive a certificate (if required)
3. Connect to the site.

For more information see the *SecuRemote/SecureClient User Guide*.

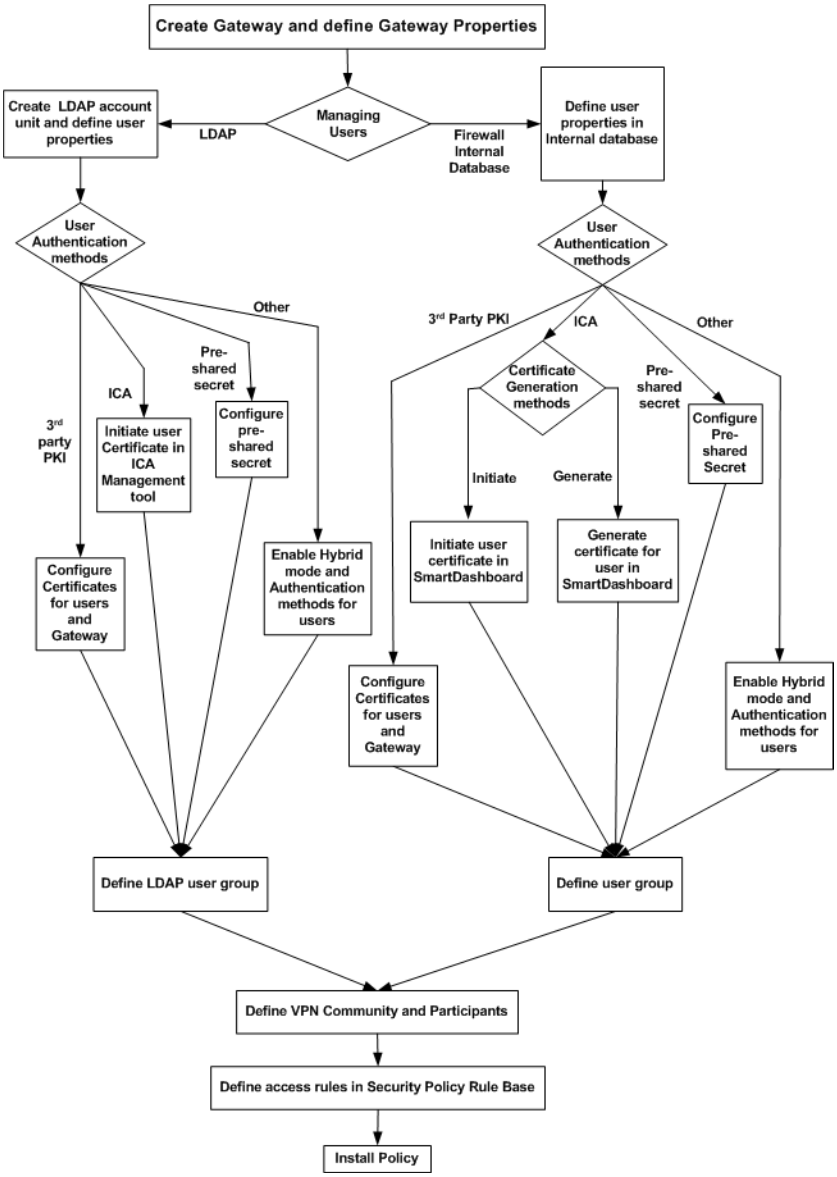
Establishing Remote Access VPN

The general workflow for establishing remote access VPN is shown in Figure 14-2. Start at the top, with *Create Gateway and define Gateway properties*, and trace a route down to *Install policy*.

Sections following the chart detail step-by-step procedures for each phase.

Figure 14-2 displays a general workflow for establishing remote access VPN:

Figure 14-2 Work Flow for establishing remote access VPN


Create Gateway and define Gateway Properties
Managing Users
LDAP
Create LDAP account unit and define user properties
Firewall Internal Database
Define user properties in Internal database
User Authentication methods
User Authentication methods
Other
3rd party PKI
ICA
Pre-shared secret
Initiate user Certificate in ICA Management tool
Configure pre-shared secret
Enable Hybrid mode and Authentication methods for users
Configure Certificates for users and Gateway
3rd Party PKI
ICA
Other
Certificate Generation methods
Pre-shared secret
Configure Pre-shared Secret
Initiate
Generate
Initiate user certificate in SmartDashboard
Generate certificate for user in SmartDashboard
Configure Certificates for users and Gateway
Enable Hybrid mode and Authentication methods for users
Define LDAP user group
Define user group
Define VPN Community and Participants
Define access rules in Security Policy Rule Base
Install Policy

Creating the Gateway and Defining Gateway Properties

1. In SmartDashboard, create the gateway network object.
2. On the **General Properties** page of the network object, select **VPN**.
3. Initialize a secure communication channel between the VPN module and the Security Management server by clicking **Communication...**
4. On the **Topology** page of gateway, define the gateway's interfaces and the VPN domain.

A certificate is automatically issued by the Internal CA for the gateway.

Defining User and Authentication Methods in LDAP

1. Obtain and install a license that enables the VPN module to retrieve information from an LDAP server.
2. Create an LDAP account unit.
3. Define users as LDAP users. A new network object for LDAP users is created on the Users tree. (The LDAP users also appear in the objects list window to the right.)

For more information see: LDAP and User Management in the *Security Management Server Administration Guide*.

Defining User Properties and Authentication Methods

Refer to *Overview* section in the *Security Management Server Administration Guide*.

Initiating User Certificates in the ICA Management Tool

1. Double click a user to open that user's property window. On the **Encryption** tab click **Edit...** The **IKE phase 2 properties** window opens. On the **Authentication** tab of this window, select **Public Key**.
2. Initiate the user certificate in the ICA management tool. For more information see the *Security Management Server Administration Guide*.

Generating Certificates for Users in SmartDashboard

1. On the **User properties** window, **Encryption** tab, click **Edit...** The **IKE phase 2 properties** window opens. On the **Authentication** tab, select **Public key**.
2. In the **Certificates** tab of the **User Properties** window, click **Generate and Save**.
3. Enter and confirm a PKCS #12 password.

   PKCS #12 is a portable format for storing or transporting a user’s private keys, certificates, *etc*. The PKCS #12 file and the password should be securely transferred to the user "out-of-band", preferably via diskette.
4. In **Global Properties**, **Authentication** window, add or disable suffix matching.

   For users with certificates, it is possible to specify that only certificates with a specified suffix in their DN are accepted. This feature is enabled by default, and is required only if the user names are not the full DN. All certificates DN’s are checked against this suffix.

Initiating Certificates for Users in SmartDashboard

An alternative to generating certificates for remote users is to only *initiate* the certificate generation process. The process is then completed by the user.

To initiate the certificate creation process:

1. On the **User properties** window, **Encryption** tab, click **Edit...** The **IKE phase 2 properties** window opens. On the **Authentication** tab, select **Public key**.
2. In the **Certificates** tab of the **User Properties** window, click **Initialize** and select **Copy to clipboard**. The registration key is copied to the clipboard.
3. Open a text editor (for example, Notepad) and paste in the registration key.
4. Transfer the registration key to the user “out-of-band.”
5. In **Global Properties**, **Authentication** window, add or disable suffix matching.

   For users with certificates, it is possible to specify that only certificates with a specified suffix in their DN are accepted. This feature is enabled by default, and is required only if the user names are not the full DN. All certificates DN’s are checked against this suffix.

Configure Certificates Using Third Party PKI

Using third party PKI involves creating:

- A certificate for the user and
- A certificate for the gateway

You can use a third-party OPSEC PKI certificate authority that supports the PKCS#12, CAPI or Entrust standards to issue certificates for security gateways and users. The gateway must trust the CA and have a certificate issued by the CA.

For users managed on an LDAP server, the full distinguished name (DN) which appears on the certificate is the same as the user's name. But if the user is managed on the internal database, the user name and DN on the certificate will not match. For this reason, the user name in the internal database must be either the full DN which appears on the certificate or just the name which appears in the CN portion of the certificate. For example, if the DN which appears on the certificate is:

CN=John, OU=Finance, O=Widget Enterprises, C=US

The name of the user on the internal database must be either:

- **John**, or:
- **CN=John, OU=Finance, O=Widget Enterprises, C=US**

Note - The DN on the certificate must include the user's LDAP branch. Some PKI solutions do not include (by default) the whole branch information in the subject DN, for example the DN only includes the common name. This can be rectified in the CA configuration.

To use a third-party PKI solution:

1. On the **User properties** window, **Encryption** tab, click **Edit...** The **IKE phase 2 properties** window opens. On the **Authentication** tab, select **Public key**.
2. Define the third party Certificate Authority as an object in SmartDashboard. See "Enrolling with a Certificate Authority".
3. Generate a certificate for your security gateway from the third party CA. For more information, see: "Enrolling with a Certificate Authority".
4. Generate a certificate for the remote user from the third party CA. (Refer to relevant third party documentation for details.) Transfer the certificate to the user.
5. In **Global Properties**, **Authentication** window, add or disable suffix matching.

For users with certificates, it is possible to specify that only certificates with a specified suffix in their DN are accepted. This feature is enabled by default, and is required only if:

- Users are defined in the internal database, *and*
- The user names are not the full DN.

All certificates DN’s are checked against this suffix.

Note - If an hierarchy of Certificate Authorities is used, the chain certificate of the user must reach the same root CA that the gateway trusts.

Enabling Hybrid Mode and Methods of Authentication

Hybrid mode allows the gateway and remote access client to use different methods of authentication. To enable Hybrid Mode:

From **Policy > Global Properties > Remote Access >VPN - Basic** select **Hybrid Mode**.

Defining User Authentication Methods in Hybrid Mode

1. On the **User Properties** window, **Authentication** tab, select an appropriate authentication scheme.
2. Enter authentication credentials for the user.
3. Supply the user ("out-of-band") with these credentials.

For information regarding authentication methods for users, authentication servers, and enabling authentication methods on the gateway, see the chapter on “Authentication” in the *FireWall Administration Guide*.

Configuring Authentication for NT groups and RADIUS Classes

To enable this group authentication feature:

1. Set the `add_radius_groups` property in `objects.C` to “true”,
2. Define a generic* profile, with RADIUS as the authentication method.
3. Create a rule in the Policy rule base whose “source” is this group of remote users that authenticate using NT Server or RADIUS.

Office Mode IP assignment file

This method also works for Office Mode. The group listed in the `ipassignment.conf` file points to the group that authenticates using NT group authentication or RADIUS classes. See: “Office Mode via ipassignment.conf File” on page 318.

Using a Pre-Shared Secret

When using pre-shared secrets, the remote user and security gateway authenticate each other by verifying that the other party knows the shared secret: the user’s password. To enable the use of pre-shared secrets:

1. In **Policy > Global Properties > Remote Access > VPN — Basic**, select **Pre-Shared Secret (For SecuRemote/SecureClient users)**
2. Deselect **Hybrid Mode**.
3. For each user, go to the **Encryption** tab of the **User Properties** window, select **IKE** and click **Edit...** to display the **IKE Phase 2 Properties** window.
4. In the **Authentication** tab, enable **Password (Pre-Shared Secret)** and enter the pre-shared secret into the **Password (Pre-shared secret)** and **Confirm Password** fields.
5. Inform the user of the password “out-of-band”.

Defining an LDAP User Group

See: *LDAP and User Management* in the *Security Management Server Administration Guide*.

Defining a User Group

In SmartDashboard, create a group for remote access users. Add the appropriate users to this group.

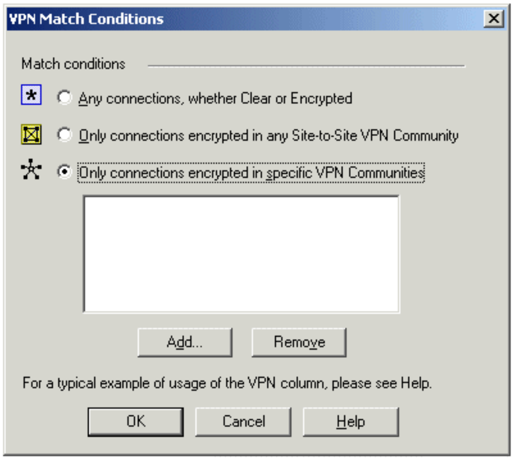
Defining a VPN Community and its Participants

1. On the VPN Communities tree, double-click **Remote_Access_Community**. The **Remote Access Community Properties** window opens.
2. On the **Participating Gateways** page, **Add...** gateways participating in the Remote Access Community.
3. On the **Participating User Groups** page, **Add...** the group that contains the remote access users.

Defining Access Control Rules

Access control is a layer of security not connected with VPN. The existence of a remote access community does not mean that members of that community have free automatic access to the network. Appropriate rules need to be created in the Security Policy Rule Base blocking or allowing specific services.

1. Create a rule in the Security Policy Rule Base that deals with remote access connections.
2. Double-click the entry in the VPN column. The **VPN Match Conditions** window opens:



VPN Match Conditions

Match conditions

○ Any connections, whether Clear or Encrypted

○ Only connections encrypted in any Site-to-Site VPN Community

● Only connections encrypted in specific VPN Communities

For a typical example of usage of the VPN column, please see Help.

3. Select **Only connections encrypted in specific VPN Communities**.

4. Click **Add...** to include a specific community in this Security Policy Rule.
5. Define services and actions. For example, to allow remote access users to access the organization's SMTP server, called SMTP_SRV, create the following rule:

Table 14-1

Source	Destination	VPN	Service	Action	Track
Any	SMTP_SRV	Remote_Access_Community	SMTP	Accept	Log

Installing the Policy

Install the policy and instruct the users to create or update the site topology.

User Certificate Management

Managing user certificates involves:

- Tracing the status of the user's certificate
- Automatically renewing a certificate
- Revoking certificates

Tracing the Status of User's Certificate

The status of a user's certificate can be traced at any time in the **Certificates** tab of the user's Properties window. The status is shown in the **Certificate state** field. If the certificate has not been generated by the user by the date specified in the **Pending until** field, the registration key is deleted.

If the user is defined in LDAP, then tracing is performed by the ICA management tool.

Automatically Renewing a Users' Certificate

ICA certificates for users can be automatically renewed a number of days before they expire. The client initiates a certificate renewal operation with the CA before the expiration date is reached. If successful, the client receives an updated certificates.

To configure automatic certificate renewal:

1. Select **Policy > Global Properties > Remote Access > Certificates**.

2. Select **Renew users internal CA certificates** and specify a time period. The time period is the number of days before the user’s certificate is about to expire in which the client will attempt to renew the certificate.
3. Install the Security Policy.
4. Instruct the user to update the site’s topology.

Revoking Certificates

The way in which certificates are revoked depends on whether they are managed internally or externally, via LDAP.

For internally managed Users

When a user is deleted, their certificate is automatically revoked. Certificates can be disabled or revoked at any time.

If you initiated a certificate generation that was not completed by the user, you can disable the pending certificate by clicking **Disable** in the **Certificates** tab of the **User Properties** window.

If the certificate is already active, you can revoke it by clicking **Revoke** in the **Certificates** tab of the **User Properties** window.

For Users Managed in LDAP

If users are managed in LDAP, certificates are revoked using the ICA management tool.

Modifying Encryption Properties for Remote Access VPN

The encryption properties of the users participating in a Remote Access community are set by default. If you must modify the encryption algorithm, the data integrity method and/or the Diffie-Hellman group, you can either do this globally for all users or configure the properties per user.

To modify the user encryption properties globally:

1. Select **Policy > Global Properties > Remote Access > VPN - (IKE Phase 1)**.

Configure the appropriate settings:

- **Support encryption algorithms** - Select the encryption algorithms that will be supported with remote hosts.
- **Use encryption algorithms** - Choose the encryption algorithm that will have the highest priority of the selected algorithms. If given a choice of more that one encryption algorithm to use, the algorithm selected in this field will be used.
- **Support Data Integrity** - Select the hash algorithms that will be supported with remote hosts to ensure data integrity.
- **Use Data Integrity** - The hash algorithm chosen here will be given the highest priority if more than one choice is offered.
- **Support Diffie-Hellman groups** - Select the Diffie-Hellman groups that will be supported with remote hosts.
- **Use Diffie-Hellman group** - SecureClient users utilize the Diffie-Hellman group selected in this field.

To enforce the global encryption properties for some users while being able to modify them for specific users go to **Policy > Global Properties > Remote Access > VPN - (IPSEC Phase 2)**:

1. Set the required properties in the window and disable **Enforce Encryption Algorithm and Data Integrity on all users**.
2. In the **Encryption** tab of the **User Properties** window select **IKE** and click **Edit**.

   The **IKE Phase 2 Properties** window is displayed.
3. Select the **Encryption** tab.

4. If you want the encryption and data integrity algorithms of the user to be taken from the **Global Properties** definitions, select **Defined in the Remote Access VPN** page of the **Global Properties** window. If you want to customize the algorithms for this user, select **Defined below** and select the appropriate encryption and data integrity algorithms.

Working with RSA'S Hard and Soft Tokens

If you use SecurID for authentication, you must manage the users on RSA's ACE management server. ACE manages the database of RSA users and their assigned hard or soft tokens. SecureClient contacts the site's gateway. The gateway contacts the ACE Server for user authentication information. This means:

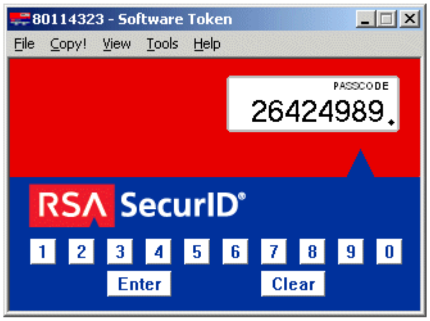
- The remote users must be defined as RSA users on the ACE Server.
- On the security gateway, the SecurID users must be placed into a group with an external user profile account that specifies SecurID as the authentication method.

SecurID Authentication Devices

Several versions of SecurID devices are available. The older format is a small device that displays a numeric code, called a *tokencode*, and time bars. The token code changes every sixty seconds, and provides the basis for authentication. To authenticate, the user must add to the beginning of the tokencode a special password called a PIN number. The time bar indicates how much time is left before the next tokencode is generated. The remote user is requested to enter both the PIN number and tokencode into SecureClient's connection window.

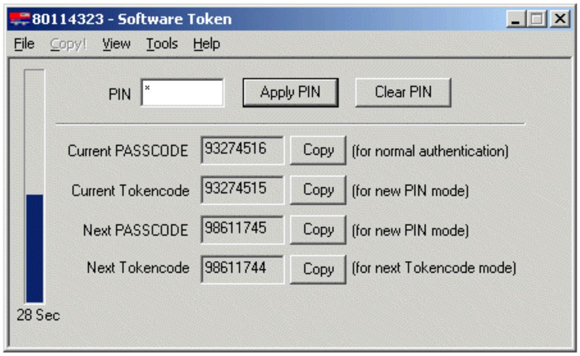
The newer format resembles a credit card, and displays the tokencode, time bars and a numeric pad for typing in the PIN number. These type of device mixes the tokencode with the entered PIN number to create a *Passcode*. SecureClient requests only the passcode.

SoftID operates the same as the passcode device but consists only of software that sits on the desktop.



PASSCODE 26424989

The Advanced view displays the tokencode and passcode with COPY buttons, allowing the user to cut and paste between softID and SecureClient.



PIN *

Current PASSCODE	93274516	(for normal authentication)
Current Tokencode	93274515	(for new PIN mode)
Next PASSCODE	98611745	(for new PIN mode)
Next Tokencode	98611744	(for next Tokencode mode)

28 Sec

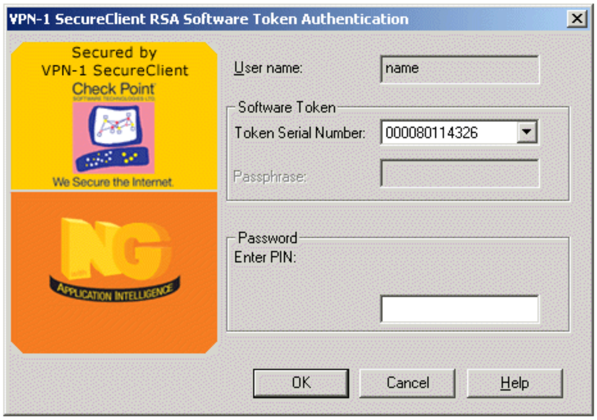
SoftID and SecureClient

For remote users to successfully use RSA’s softID:

1. The administrator creates the remote users on the Ace Server
2. "Out-of-band", the administrator distributes the SDTID token file (or several tokens) to the remote users.
3. The remote user imports the tokens.
4. The following `userc.c` property on SecureClient must be set in the OPTIONS section:

```
support_rsa_soft_tokens (true)
```

The remote user sees three windows:



VPN-1 SecureClient RSA Software Token Authentication

User name: name

Software Token

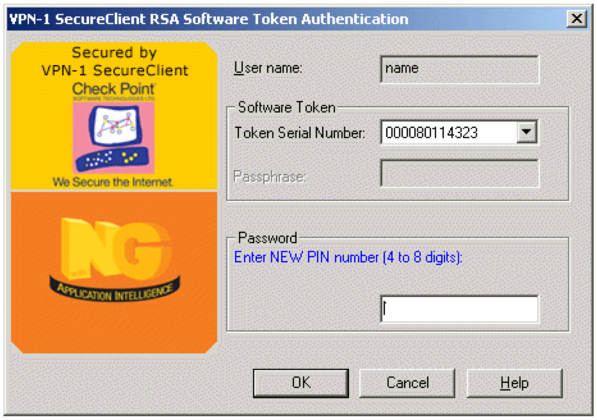
Token Serial Number: 000080114326

Passphrase:

Password

Enter PIN:

In this window, the remote user needs to enter the Token Serial Number and PIN. If the remote user does not enter a PIN number, the following window appears:



VPN-1 SecureClient RSA Software Token Authentication

User name: name

Software Token

Token Serial Number: 000080114323

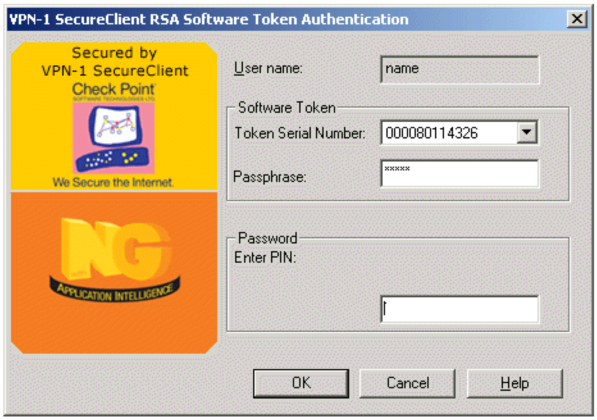
Passphrase:

Password

Enter NEW PIN number (4 to 8 digits):

The PIN must be entered.

If the token requires a passphrase, the remote user sees this window:



VPN-1 SecureClient RSA Software Token Authentication

User name: name

Software Token

Token Serial Number: 000080114326

Passphrase: *****

Password

Enter PIN:

Chapter 15

Office Mode

In This Chapter

The Need for Remote Clients to be Part of the LAN

As remote access to internal networks of organizations becomes widespread, it is essential that remote users are able to access as many of the internal resources of the organization as possible.

Typically, when remote access is implemented, the client connects using an IP address locally assigned by, for example, an ISP. The client may even receive a non-routable IP which is then hidden behind a NATing device. Because of this, several problems may arise:

- Some networking protocols or resources may require the client's IP address to be an internal one. Router ACLs (access lists), for example, might be configured to allow only specific or internal IP addresses to access network resources. This is difficult to adjust without knowing the a remote client's IP address in advance.
- When assigned with a non-routable IP address a conflict may occur, either with similar non-routable addresses used on the corporate LAN, or with other clients which may receive the same IP address while positioned behind some other hiding NAT device.

  For example, if a SecuRemote/SecureClient user receives an IP of 10.0.0.1 which is entered into the headers of the IPSec packet. The packet is NATed. The packet's new source IP is 192.168.17.5. The gateway decapsulates the NATed IP and decrypts the packet. The IP address is reverted to its original source IP of 10.0.0.1. If there is an internal host with the same IP, the packet will probably be dropped (if anti-spoofing is turned on). If there is no duplicate IP, and the packet is forwarded to some internal server, the server will then attempt to reply to an non-existent address.
- Two remote users are assigned the same IP address by an ISP (for example, two users are accessing the organization from hotels which provide internal addresses and NAT them on the outbound). Both users try to access the internal network with the same IP address. The resources on the internal network of the organization may have difficulty distinguishing between the users.

Office Mode Solution

In This Section

Introducing Office Mode

Office Mode enables a security gateway to assign a remote client an IP address. The assignment takes place once the user connects and authenticates. The assignment lease is renewed as long as the user is connected. The address may be taken either from a general IP address pool, or from an IP address pool specified per user group. The address can be specified per user, or via a DHCP server, enabling the use of a name resolution service. With DNS name resolution, it is easier to access the client from within the corporate network.

It is possible to allow all your users to use Office Mode, or to enable the feature for a specific group of users. This can be used, for example, to allow privileged access to a certain group of users (e.g., administrators accessing the LAN from remote stations). It is also useful in early integration stages of Office Mode, allowing you time to “pilot” this feature on a specific group of users, while the rest of the users continue to work in the traditional way.

Office Mode is supported with the following:

- SecureClient
- SSL Network Extender
- Crypto
- L2TP

How Office Mode Works

When you connect to the organization, an IKE negotiation is initiated automatically to the security gateway. When using Office Mode, a special IKE mode called *config mode* is inserted between phase 1 and phase 2 of IKE. During config mode, the client requests an IP from the gateway. Several other parameters are also configurable this way, such as a DNS server IP address, and a WINS server IP address.

After the gateway allocates the IP address, the client assigns the IP to a Virtual Adapter on the Operating system. The routing of packets to the corporate LAN is modified to go through this adapter. Packets routed in this way bear the IP address assigned by the gateway as their source IP address. Before exiting through the real adapter, the packets will be IPSec encapsulated using the external IP address (assigned to the real adapter) as the source address. In this way, non-routable IP addresses can be used with Office Mode; the Office Mode non-routable address is concealed within the IPSec packet.

For Office Mode to work, the IP address assigned by the security gateway needs to be routable to that gateway from within the corporate LAN. This will allow packets on the LAN being sent to the client to be routed back through the gateway. (See also: "Office Mode and Static Routes in a Non-flat Network" on page 304).

Note - A remote user with SecuRemote only is not supported in Office Mode.

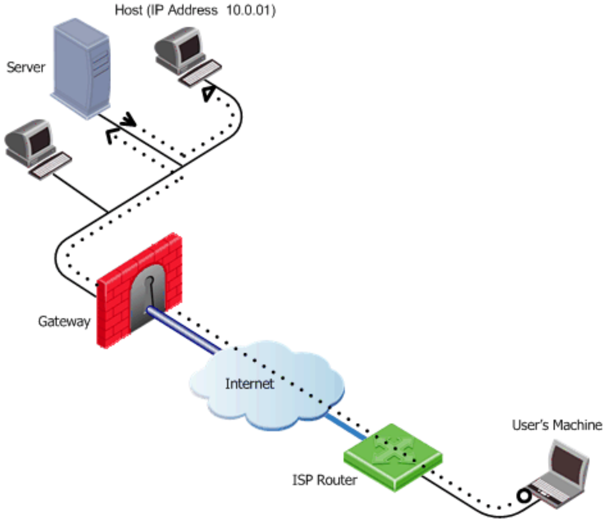
A Closer Look

The following steps illustrate the process taking place when a remote user connected through Office Mode wishes to exchange some information with resources inside the organization:

- The user is trying to connect to some resource on the LAN, thus a packet destined for the internal network is to be sent. This packet is routed through the virtual interface that Office Mode had set up, and bears the source IP address allocated for the remote user.
- The packet is encrypted and builds a new encapsulating IP header for it. The source IP of the encapsulating packet is the remote client's original IP address, and its destination is the IP address of the security gateway. The encapsulated packet is then sent to the organization through the Internet.

- The security gateway of the organization receives the packet, decapsulates and decrypts it, revealing the original packet, which bears the source IP allocated for the remote user. The gateway then forwards the decapsulated packet to its destination.
- The internal resource gets a packet seemingly coming from an internal address. It processes the packet and sends response packets back to the remote user. These packets are routed back to the (internal) IP address assigned to the remote user.
- The gateway gets the packet, encrypts and encapsulates it with the remote users' original (routable) IP address and returns the packet back to the remote user:

Figure 15-1 Packets routed correctly to the remote client.



In Figure 15-1:

- The remote host uses the Office mode address in the encapsulated packet and 10.0.0.1 in the encapsulating header.
- The packet is NATed to the new source address: 192.168.17.5
- The gateway decapsulates the NATed IP address and decrypts the packet. The source IP address is the Office Mode address.
- The packet is forwarded to the internal server, which replies correctly.

Office Mode and Static Routes in a Non-flat Network

A flat network is one in which all stations can reach each other without going through a bridge or a router. One segment of a network is a "flat network". A static route is a route that is manually assigned by the system administrator (to a router) and needs to be manually updated to reflect changes in the network.

If the LAN is non-flat (stations reach each other via routers and bridges) then the OM address of the remote client must be statically assigned to the routers so that packets on the LAN, destined for the remote client, are correctly routed to the gateway.

Assigning IP Addresses

The internal IP addresses assigned by the gateway to the remote user can be allocated using one of the following methods:

- IP Pool
- DHCP Server

IP Pool

The System Administrator designates a range of IP addresses to be utilized for remote client machines. Each client requesting to connect in Office Mode is provided with a unique IP address from the pool.

IP Assignment Based on Source IP Address

IP addresses from the IP pool may be reserved and assigned to remote users based on their source IP address. When a remote host connects to the gateway, its IP address is compared to a predefined range of source IP addresses. If the IP address is found to be in that range, then it is assigned an Office Mode IP address from a range dedicated for that purpose.

The IP addresses from this reserved pool can be configured to offer a separate set of access permissions given to these remote users.

DHCP Server

A Dynamic Host Configuration Protocol (DHCP) server can be used to allocate IP addresses for Office Mode clients. When a remote user connects to the gateway using Office Mode, the gateway requests the DHCP server to assign the user an IP address from a range of IP addresses designated for Office Mode users.

Security gateway DHCP requests can contain various client attributes that allow DHCP clients to differentiate themselves. The attributes are pre configured on the client side operating system, and can be used by different DHCP servers in the process of distributing IP addresses. Security gateways DHCP request can contain the following attributes:

- Host Name
- Fully Qualified Domain Name (FQDN)
- Vendor Class
- User Class

RADIUS Server

A RADIUS server can be used for authenticating remote users. When a remote user connects to a gateway, the username and password are passed on to the RADIUS server, which checks that the information is correct, and authenticates the user. The RADIUS server can also be configured to allocate IP addresses.

Note - Authentication and IP assignment must be performed by the same RADIUS server.

IP Address Lease duration

When a remote user’s machine is assigned an IP address, that machine can use it for a certain amount of time. This time period is referred to as the “IP address lease duration.” The remote client automatically asks for a lease renewal after half of the IP lease duration period has elapsed. Hence, if the IP lease duration time is set to 60 minutes, a renewal request will be sent after 30 minutes. If a renewal is granted, the client will request a renewal again after 30 minutes and so on. If the renewal fails, the client attempts again after half of the remaining time, e.g. 15 minutes, then 7.5 minutes, etc. If no renewal is granted and the 60 minutes of the lease duration times out, the tunnel link terminates. To renew the connection the remote user must reconnect to the gateway. Upon reconnection, an IKE renegotiation is initiated and a new tunnel created.

When the IP address is allocated from a predefined IP pool on the gateway, the gateway determines the IP lease duration period, default being 15 minutes.

When using a DHCP server to assign IP addresses to users, the DHCP server’s configuration determines the IP lease duration. When a user disconnects and reconnects to the gateway within a short period of time, it is likely that the user will get the same IP address as before.

Using Name Resolution - WINS and DNS

To facilitate access of a remote user to resources on the internal network, the administrator can specify WINS and DNS servers for the remote user. This information is sent to the remote user during IKE config mode along with the IP address allocation information, and is used by the remote user’s operating system for name-to-IP resolution when the user is trying to access the organization’s internal resources.

Anti Spoofing

With Anti Spoofing, a network administrator configures which IP addresses are expected on each interface of the security gateway. Anti-spoofing ensures IP addresses are only received or transmitted in the context of their respective gateway interfaces. Office Mode poses a problem to the anti-spoofing feature, since a client machine can connect and authenticate through several interfaces, e.g. the external interface to the Internet, or the wireless LAN interface; thus an Office Mode IP address may be encountered on more than one interface. Office Mode enhances Anti Spoofing by making sure an encountered Office Mode IP address is indeed assigned to the user, authenticated on the source IP address on the IPSec encapsulating packet, i.e. the external IP.

Using Office Mode with Multiple External Interfaces

Typically, routing is performed before encryption in VPN. In some complex scenarios of Office Mode, where the gateway may have several external interfaces, this might cause a problem. In these scenarios, packets destined at a remote user's virtual IP address will be marked as packets that are supposed to be routed through one external interface of the gateway. Only after the initial routing decision is made do the packets undergo IPSEC encapsulation. After the encapsulation, the destination IP address of these packets is changed to the original IP address of the client. The routing path that should have been selected for the encapsulated packet might be through a different external interface than that of the original packet (since the destination IP address changed), in which case a routing error occurs. Office Mode has the ability to make sure that all Office Mode packets undergo routing *after* they are encapsulated.

Office Mode Per Site

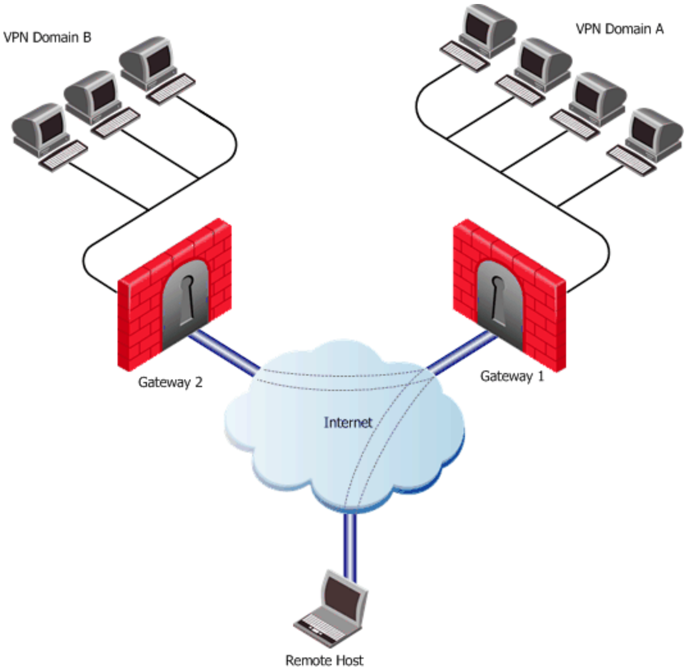
After a remote user connects and receives an Office Mode IP address from a gateway, every connection to that gateways encryption domain will go out with the Office Mode IP as the internal source IP. The Office Mode IP is what hosts in the encryption domain will recognize as the remote user's IP address.

The Office Mode IP address assigned by a specific gateway can be used in its own encryption domain and in neighboring encryption domains as well. The neighboring encryption domains should reside behind gateways that are members of the same VPN community as the assigning gateway. Since the remote hosts connections are dependant on the Office Mode IP address it received, should the gateway that issued the IP become unavailable, all the connections to the site will terminate.

In order for all gateways on the site to recognize the remote users Office Mode IP addresses, the Office Mode IP range must be known by all of the gateways and the IP ranges must be routable in all the networks. However, when the Office Mode per Site feature is in use, the IP-per-user feature cannot be implemented.

Note - When Office Mode per Site is activated, Office Mode Anti-Spoofing is not enforced.

Figure 15-2 Office Mode per Site



In this scenario:

- The remote user makes a connection to gateway 1.
- Gateway 1 assigns an Office Mode IP address to the remote user.
- While still connected to gateway 1, the remote user can make a connection to hosts behind gateway 2 using the Office Mode IP address issued by gateway 1.

Enabling IP Address per User

The Problem

In some configurations, a router or other device restricts access to portions of the network to specified IP addresses. A remote user connecting in Office Mode must be able to ensure that he or she is allocated an IP address which will allow the connection to pass through the router.

Note - If this feature is implemented, it is imperative to enable anti-spoofing for Office Mode. See "Anti Spoofing" on page 307 for more information.

The Solution

There are two ways to implement this feature, depending on whether IP addresses are allocated by a DHCP server or IP Pool.

DHCP Server

If Office Mode addresses are allocated by a DHCP server, proceed as follows:

1. Open the Check Point object from the Objects Tree.
2. In the **Object Properties > Remote Access > Office Mode** page:
    - Enable Office Mode (either for all users or for the relevant group)
    - Select a DHCP server and under **MAC address for DHCP allocation**, select **calculated per user name**
3. Install the Policy on the Module.
4. On the Module, run the following command to obtain the MAC address assigned to the user.

```
vpn macutil <username>
```

5. On the DHCP Server make a new reservation, specifying the IP address and MAC address, assigning the IP address for the exclusive use of the given user.

ipassignment.conf File

The `$FWDIR/conf/ipassignment.conf` file on the Module, is used to implement the IP-per-user feature. It allows the administrator to assign specific addresses to specific users or specific ranges to specific groups when they connect using Office Mode or L2TP clients.

For an explanation of the file's syntax, see the comments (the lines beginning with the # character) in the sample file below.

Note - This file must be *manually* added to all the modules.

Sample ipassignment.conf File

```
# This file is used to implement the IP-per-user feature. It allows the
# administrator to assign specific addresses to specific users or specific
# ranges to specific groups when they connect using Office Mode or L2TP.
#
# The format of this file is simple: Each line specifies the target
# gateway, the IP address (or addresses) we wish to assign and the user
# (or group) name as in the following examples:
#
# Gateway        Type   IP Address                          User Name
# ============= ===== =========================== ========================
# Paris-GW,             10.5.5.8,                           Jean
# Brasilia,      addr   10.6.5.8,                         Joao  # comments
are allowed
# Miami,         addr   10.7.5.8, CN=John,OU=users,O=cpmgmt.acme.com.gibeuu
# Miami          range  100.107.105.110-100.107.105.119/24  Finance
# Miami          net    10.7.5.32/28                        Accounting
#
# Note that real records do not begin with a pound-sign (#), and the commas
# are optional. Invalid lines are treated as comments. Also, the
# user name may be followed by a pound-sign and a comment.
#
# The first item is the gateway name. This could be a name, an IP
# address or an asterisk (*) to signify all gateways. A gateway will
# only honor lines that refer to it.
#
# The second item is a descriptor. It can be 'addr', 'range' or 'net'.
# 'addr' specifies one IP for one user. This prefix is optional.
# 'range' and 'net' specify a range of addresses. These prefixes are
# required.
#
# The third item is the IP address or addresses. In the case of a single
# address, it is specified in standard dotted decimal format.
# ranges can be specified either by the first and last IP address, or using
# a net specification. In either case you need to also specify the subnet
# mask length ('/24' means 255.255.255.0). With a range, this is the subnet
# mask. With a net it is both the subnet mask and it also determines the
# addresses in the range.
#
# The last item is the user name. This can be a common name if the
# user authenticates with some username/password method (like hybrid
# or MD5-Challenge) or a DN if the user authenticates with a
# certificate.
```

Office Mode Considerations

In This Section

IP pool Versus DHCP

The question of whether IP addresses should be assigned by the Firewall (using IP pools) or by a DHCP server is a network administration and financial issue. Some network administrators may prefer to manage all of their dynamic IP addresses from the same location. For them, a central DHCP server might be preferable. Moreover, DHCP allows a cluster to assign all the addresses from a single pool, rather than have a different pool per cluster member as you have to with Firewall IP pools. On the other hand, purchasing a DHCP server can be viewed by some as an unnecessary financial burden, in which case the IP pool option might be preferred.

Routing Table Modifications

IP addresses, assigned by Office Mode need to be routed by the internal LAN routers to the gateway (or gateway cluster) that assigned the address. This is to make sure packets, destined to remote access Office Mode users, reach the gateway in order to be encapsulated and returned to the client machine. This may require changes to the organization's routing tables.

Using the Multiple External Interfaces Feature

Enabling this feature instructs Office Mode to perform routing decisions *after* the packets are encapsulated using IPSEC, to prevent routing problems discussed in "Using Office Mode with Multiple External Interfaces" on page 307. This feature adds new checks and changes to the routing of packets through the gateway, and has an impact on performance. As a result, it is recommended to use this feature only when:

- The gateway has multiple external interfaces, *and*
- Office Mode packets are routed to the wrong external interface.

Configuring Office Mode

In This Section

Before configuring Office Mode the assumption is that standard VPN Remote Access has already been configured. For more details on how to configure VPN Remote Access, see "Introduction to Remote Access VPN" on page 269.

Before starting the Office Mode configuration, you must select an internal address space designated for remote users using Office Mode. This can be any IP address space, as long as the addresses in this space do not conflict with addresses used within the enterprise domain. It is possible to choose address spaces which are not routable on the Internet, such as 10.x.x.x.

The basic configuration of Office Mode is using IP pools. The configuration of Office Mode using DHCP for address allocation can be found in "Office Mode — DHCP Configuration" on page 320.

Office Mode — IP Pool Configuration

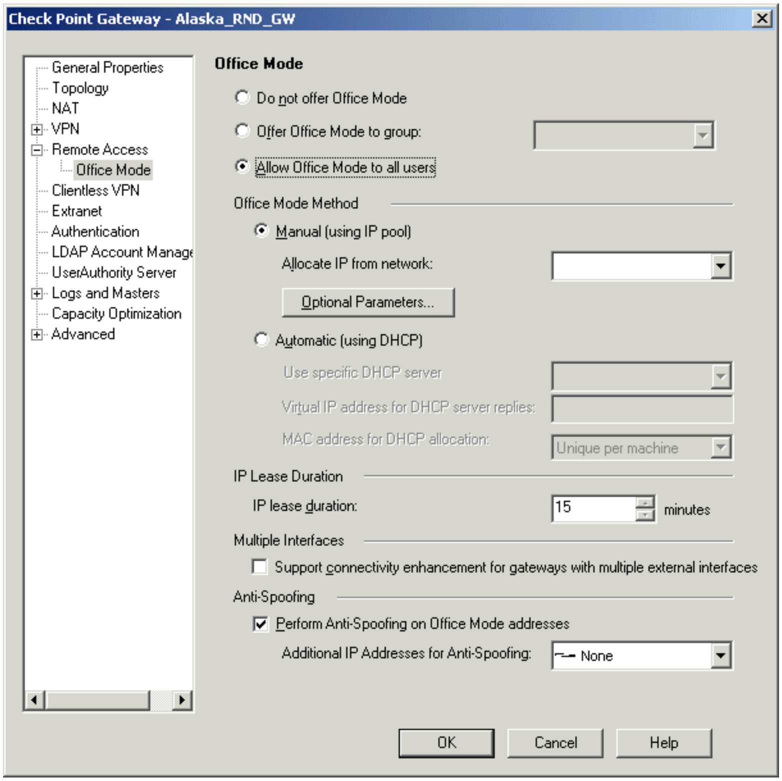
To deploy the basic Office Mode (using IP pools):

1. Create a network object to represent the IP Pool, by selecting **Manage > Network Objects > New > Network**.

   In the **Network Properties — General** tab, set the IP pool range of addresses as follows:

   - In **Network Address** specify the first address to be used (e.g. 10.130.56.0).
   - In **Net Mask** enter the subnet mask according to the amount of addresses you wish to use (entering 255.255.255.0, for example, this will designate all 254 IP addresses from 10.130.56.1 till 10.130.56.254 for Office Mode addresses.)
   - Changes to the **Broadcast Address section** and the **Network Properties — NAT** tab are not necessary.
   - Close the network object properties window.

2. Open the gateway object through which the remote users will connect to the internal network and select the **Remote Access** > **Office Mode** page. Enable **Office Mode** for either all users or for a certain group.



Check Point Gateway - Alaska_RND_GW

Office Mode

○ Do not offer Office Mode

○ Offer Office Mode to group:

● Allow Office Mode to all users

Office Mode Method

● Manual (using IP pool)

Allocate IP from network:

○ Automatic (using DHCP)

Use specific DHCP server

Virtual IP address for DHCP server replies:

MAC address for DHCP allocation: Unique per machine

IP Lease Duration

IP lease duration: 15 minutes

Multiple Interfaces

□ Support connectivity enhancement for gateways with multiple external interfaces

Anti-Spoofing

☑ Perform Anti-Spoofing on Office Mode addresses

Additional IP Addresses for Anti-Spoofing: None

Figure 15-3 Office Mode page

- In the **Allocate IP from network** select the IP Pool network object you have previously created.
- **IP lease duration** — specify the duration in which the IP is used by the remote host.
- Under **Multiple Interfaces**, specify whether you want routing to be done after the encapsulation of Office Mode packets, allowing traffic to be routed correctly when your gateway has multiple external interfaces.
- Select **Anti-Spoofing** if you wish the firewall to check that Office Mode packets are not spoofed.

It is possible to specify which WINS and DNS servers Office Mode users should use. To specify WINS and/or DNS servers, continue to step 3. Otherwise skip to step 6.

Note - WINS and DNS servers should be set on the Security Management server machine only when IP pool is the selected method.

3. Create a DNS server object, by selecting **Manage > Network Objects > New > Node > Host** and specify the DNS machine's name, IP address and subnet mask. Repeat this step if you have additional DNS servers.
4. Create a WINS server object, by selecting **Manage > Network objects > New > Node > Host** and specify the WINS machine's name, IP address and subnet mask. Repeat this step if you have additional WINS servers.
5. In the **Check Point Gateway — Remote Access > Office Mode** page, in the **IP Pool** section click the "**optional parameters**" button.
   - In the **IP Pool Optional Parameters** window, select the appropriate objects for the primary and backup DNS and WINS servers.
   - In the **Domain name** field, specify the suffix of the domain where the internal names are defined. This instructs the Client as per what suffix to add when it addresses the DNS server (e.g. example.com).
6. Install the Policy.
7. Make sure that all the internal routers are configured to route all the traffic destined to the internal address space you had reserved to Office Mode users through the security gateway. For instance, in the example above it is required to add routes to the class C sub network of 10.130.56.0 through the gateway's IP address.

In addition to the steps mentioned for the gateway side configuration, a few configuration steps have to be performed on the client side in order to connect to the gateway in Office Mode.

See: "Office Mode Configuration on SecureClient" on page 323.

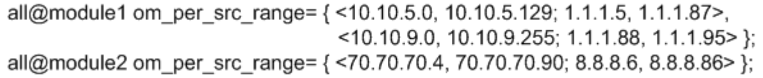
Configuring IP Assignment Based on Source IP Address

The settings for the IP Assignment Based on Source IP Address feature are configured by editing a plain text file called `user.def`. This file is located in the `\FWDIR\conf` directory of the Security Management server which manages the enforcement modules used for remote access.

A range of source IP addresses must be defined along with a corresponding range of Office Mode addresses. The `\FWDIR\conf\user.def` file can contain multiple definitions for multiple modules.

The first range defined per line is the source IP address range. The second range defined per line is the Office Mode IP address range.

Figure 15-4 IP Assignment Based on Source IP Address Example



```
all@module1 om_per_src_range= { <10.10.5.0, 10.10.5.129; 1.1.1.5, 1.1.1.87>,
                                <10.10.9.0, 10.10.9.255; 1.1.1.88, 1.1.1.95> };
all@module2 om_per_src_range= { <70.70.70.4, 70.70.70.90; 8.8.8.6, 8.8.8.86> };
```

In this scenario:

- (10.10.5.0, 10.10.5.129), (10.10.9.0, 10.10.9.255), and (70.70.70.4, 70.70.70.90) are the VPN remote clients source IP address ranges
- (1.1.1.5, 1.1.1.87), (1.1.1.88, 1.1.1.95), and (8.8.8.6, 8.8.8.68) are the Office Mode IP addresses that will be assigned to the remote users whose source IP falls in the range defined on the same line.
- For example: A user with a source IP address between 10.10.10.5.0 and 10.10.5.129, will receive an Office Mode address between 1.1.1.5 and 1.1.1.87.

IP Assignment Based on Source IP Address is enabled using a flag in the `\FWDIR\conf\objects_5_0.C` file. Add the following flag:

```
om_use_ip_per_src_range (followed by value)
```

One of the following values should be applied to the flag:

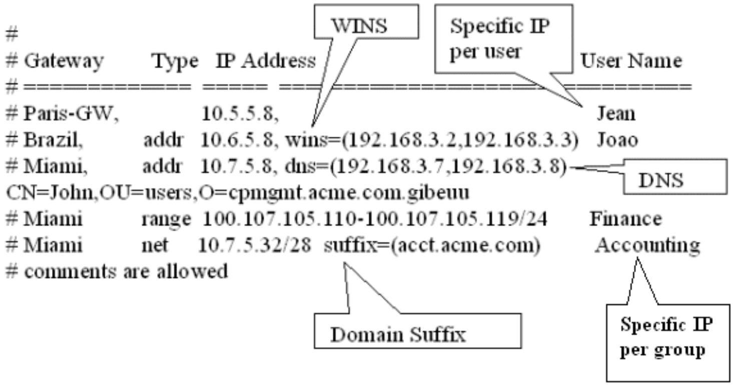
- **[Exclusively]** - If the remote hosts IP is not found in the source range, remote user does not get an Office Mode IP address.
- **[True]** - If the remote hosts IP is not found in the source IP range, the user will get an Office Mode IP address using another method.
- **[False]** (default)- The flag is not used.

Office Mode via ipassignment.conf File

It is possible to over-ride the Office Mode settings created on Security Management server by editing a plain text file called `ipassignment.conf` in the `\FWDIR\conf` directory of the VPN module. The module uses these Office Mode settings and not those defined for the object in Security Management server.

`Ipassignment.conf` can specify:

- An **IP per user/group**, so that a particular user or user group always receives the same Office Mode address. This allows the administrator to assign specific addresses to users, or particular IP ranges/networks to groups when they connect using Office Mode.
- A different **WINS server** for a particular user or group
- A different **DNS server**
- Different **DNS domain suffixes** for each entry in the file.



```
#
# Gateway       Type   IP Address                                  User Name
# ============  =====  ==================================================
# Paris-GW,            10.5.5.8,                                    Jean
# Brazil,       addr   10.6.5.8, wins=(192.168.3.2,192.168.3.3)     Joao
# Miami,        addr   10.7.5.8, dns=(192.168.3.7,192.168.3.8)
CN=John,OU=users,O=cpmgmt.acme.com.gibeuu
# Miami         range  100.107.105.110-100.107.105.119/24           Finance
# Miami         net    10.7.5.32/28  suffix=(acct.acme.com)         Accounting
# comments are allowed
```

Subnet masks and Office Mode Addresses

You cannot use the `ipassignment.conf` file to assign a subnet mask to a single user. If using IP pools, the mask is taken from the network object, or defaults to 255.255.255.0 if using DHCP.

Checking the Syntax

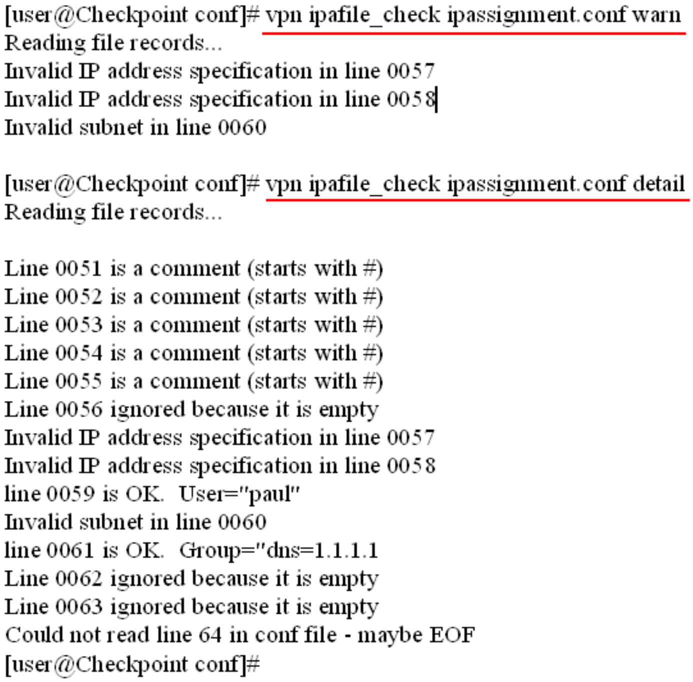
The syntax of the ipassignment file can be checked using the command `ipafile_check`.

From a shell prompt use issue: `vpn ipafile_check ipassignment.conf`

The two parameters are:

- **warn**. Display errors
- **detail**. Show all details

For example:



```
[user@Checkpoint conf]# vpn ipafile_check ipassignment.conf warn
Reading file records...
Invalid IP address specification in line 0057
Invalid IP address specification in line 0058
Invalid subnet in line 0060

[user@Checkpoint conf]# vpn ipafile_check ipassignment.conf detail
Reading file records...

Line 0051 is a comment (starts with #)
Line 0052 is a comment (starts with #)
Line 0053 is a comment (starts with #)
Line 0054 is a comment (starts with #)
Line 0055 is a comment (starts with #)
Line 0056 ignored because it is empty
Invalid IP address specification in line 0057
Invalid IP address specification in line 0058
line 0059 is OK.  User="paul"
Invalid subnet in line 0060
line 0061 is OK.  Group="dns=1.1.1.1
Line 0062 ignored because it is empty
Line 0063 ignored because it is empty
Could not read line 64 in conf file - maybe EOF
[user@Checkpoint conf]#
```

Office Mode — DHCP Configuration

1. When DHCP is the selected mode, DNS and WINS parameters are downloaded from the DHCP server. If using Office Mode in DHCP mode and you wish to supply the user with DNS and/or WINS information, make sure that the DNS and/or WINS information on your DHCP server is set to the correct IP addresses.
2. On your DHCP server's configuration, make sure that you have designated an IP address space for Office Mode users (e.g., 10.130.56.0).
3. Create a new node object by selecting **Manage > Network objects > New > Node > Host,** representing the DHCP server and specify the machine's name, IP address and subnet mask.
4. Open the gateway object through which the remote users will connect to the internal network and select the **Remote Access > Office Mode** page. Enable Office Mode to either all users or to a certain group.
   - Check the **Automatic (use DHCP)** option.
   - Select the DHCP object you have previously created.
   - In the **Virtual IP address for DHCP server replies,** specify an IP address from the sub network of the IP addresses which are designated for Office Mode usage (e.g. 10.130.56.254). Since Office Mode supports DHCP Relay method for IP assignment, you can direct the DHCP server as to where to send its replies. The routing on the DHCP server and that of internal routers must be adjusted so that packets from the DHCP server to this address are routed through the gateway.

     If you wish to use the Anti-Spoofing feature, continue to step 5, otherwise skip to step 7.
5. Create a network object to represent the address space you've allocated for Office Mode on your DHCP server, by selecting **Manage > Network Objects > New > Network**.

   In the **Network Properties — General** tab, set the DHCP address range as follows:

   - In **Network Address** specify the first address that is used (e.g. 10.130.56.0).
   - In **Net Mask** enter the subnet mask according to the amount of addresses that is used (entering 255.255.255.0, for example, designates that all 254 IP addresses from 10.130.56.1 until 10.130.56.254 are set aside for remote host Office Mode addresses on the DHCP server).

- Changes to the **Broadcast Address section** and the **Network Properties — NAT** tab are not necessary.
- Close the network object properties window.

6. Return to the gateway object, open the **Remote Access > Office Mode** page. In the **Additional IP addresses for Anti-Spoofing**, select the network object you have created with the IP address range you have set aside for Office Mode on the DHCP server.
7. Install the policy.
8. Make sure that all the internal routers are configured to route all the traffic destined to the internal address space you had reserved to Office Mode users through the security gateway. For instance, in the example above it is required to add routes to the class C sub network of 10.130.56.0 through the gateway's IP address.

In addition to the steps mentioned for the gateway side configuration, a few configuration steps have to be performed on the client side in order to connect to the gateway in Office mode. See "Office Mode Configuration on SecureClient" on page 323.

Note - Office Mode is supported only in Connect Mode.

Office Mode - Using a RADIUS Server

To configure the RADIUS server to allocate IP addresses, proceed as follows.

In SmartDashboard:

1. Click **Manage > Servers and OPSEC Applications.**
2. Select RADIUS server and click **Edit.**

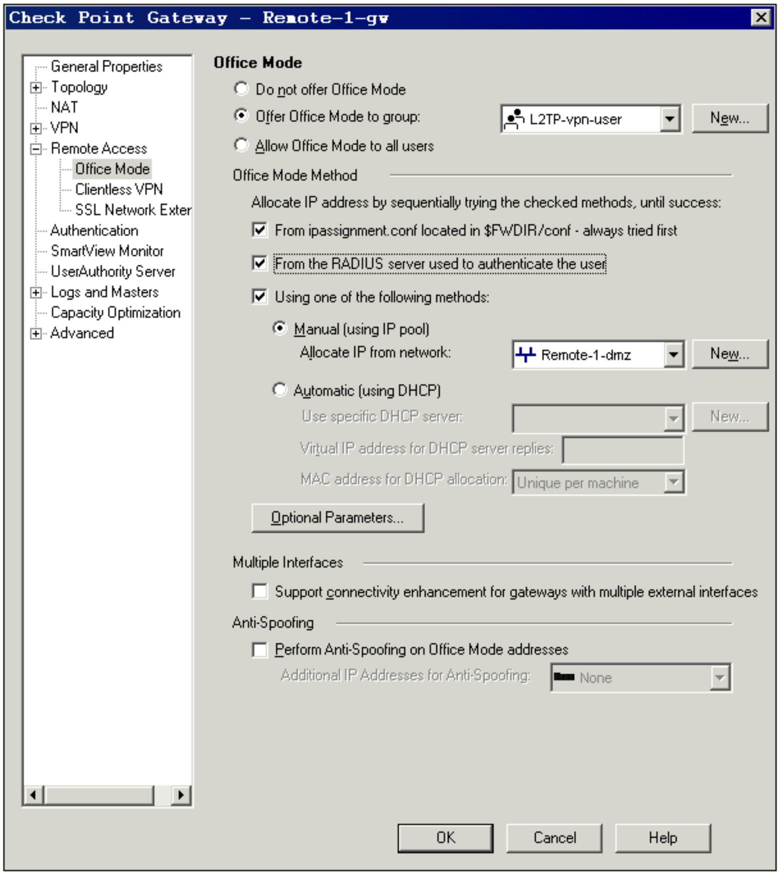
   The **RADIUS Server Properties** window appears.
3. Click the **RADIUS Accounting** tab.
4. Select **Enable IP Pool Management**.
5. Select the service the RADIUS server uses to communicate with remote users.

To configure the RADIUS server to perform authentication for remote users, proceed as follows.

In SmartDashboard:

1. Click **Manage > Network Objects**.
2. Select gateway and click **Edit**.
3. In gateway properties, select **Remote Access > Office Mode.**

Figure 15-5 Office Mode Properties Window



Office Mode

○ Do not offer Office Mode

● Offer Office Mode to group: L2TP-vpn-user

○ Allow Office Mode to all users

Office Mode Method

Allocate IP address by sequentially trying the checked methods, until success:

☑ From ipassignment.conf located in $FWDIR/conf - always tried first

☑ From the RADIUS server used to authenticate the user

☑ Using one of the following methods:

● Manual (using IP pool)

Allocate IP from network: Remote-1-dmz

○ Automatic (using DHCP)

Use specific DHCP server:

Virtual IP address for DHCP server replies:

MAC address for DHCP allocation: Unique per machine

Multiple Interfaces

☐ Support connectivity enhancement for gateways with multiple external interfaces

Anti-Spoofing

☐ Perform Anti-Spoofing on Office Mode addresses

Additional IP Addresses for Anti-Spoofing: None

4. In the **Office Mode Method** section, select **From the RADIUS server used to authenticate the user**.
5. Click **OK.**

Office Mode Configuration on SecureClient

On the client's machine the following steps should be performed in order to connect to the gateway in Office mode:

1. Right click the **SecureClient** icon in the system tray. From the pop-up menu, select **Configure.**
2. Select **Tools > Configure Connection Profile > Advanced** and select **Support Office Mode**.
3. Click **OK**, **Save** and **Close** and then select **Exit** from your **File** menu.
4. Double click your **SecureClient** icon on the bottom right side of your screen. If you're using a dial-up connection to connect to the gateway select Use Dial-up and choose the name of your dial-up connection profile from the drop-down menu (it is assumed that such a profile already exists. If dial-up is not used (i.e. connection to the gateway is done through a network interface card) proceed to step 5.
5. Select **Connect** to connect to the organization using Office Mode.

The administrator can simplify configuration, by configuring a profile in advance and providing it to the user.

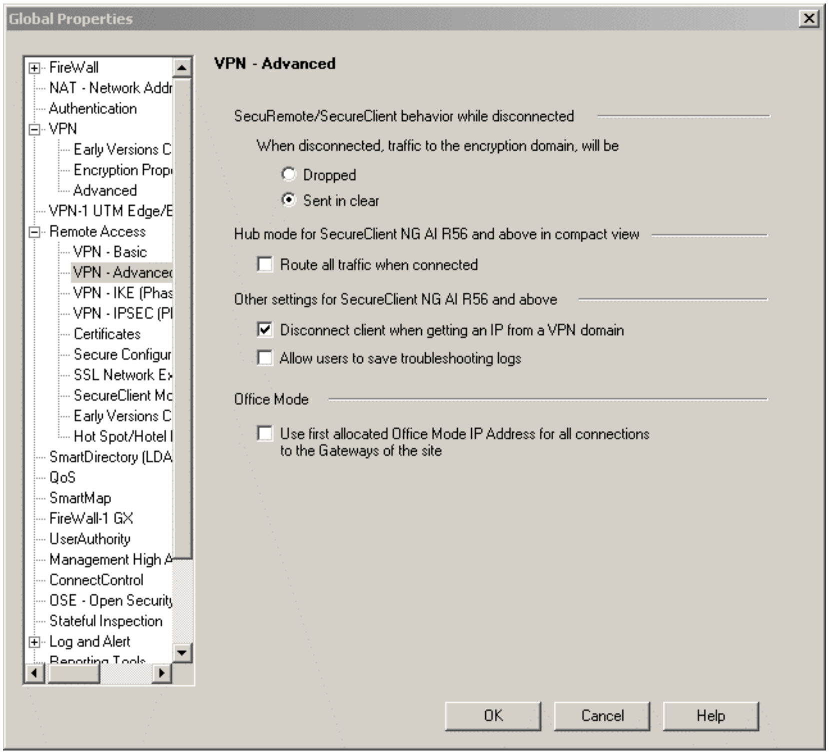
Office Mode per Site

In SmartDashboard:

1. Click **Policy > Global Properties > Remote Access > VPN - Advanced**.

   The VPN - Advanced window is displayed:

Figure 15-6 VPN - Advanced Window



VPN - Advanced

SecuRemote/SecureClient behavior while disconnected

When disconnected, traffic to the encryption domain, will be

○ Dropped

● Sent in clear

Hub mode for SecureClient NG AI R56 and above in compact view

□ Route all traffic when connected

Other settings for SecureClient NG AI R56 and above

☑ Disconnect client when getting an IP from a VPN domain

□ Allow users to save troubleshooting logs

Office Mode

□ Use first allocated Office Mode IP Address for all connections to the Gateways of the site

2. In the **Office Mode** section, select **Use first allocated Office Mode IP address for all connections to the Gateways of the site**.

3. Click **OK.**

Chapter 16
SecuRemote/SecureClient

In This Chapter

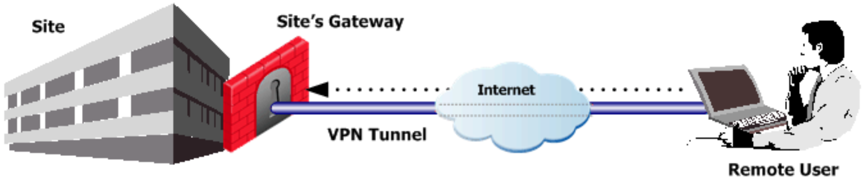
The Need for SecureClient

Anyone who wishes to send or receive e-mail while at home, or while over the weekend, needs to do so securely. When on the road, several challenges are presented by different network environments, such as a hotel Internet connection or the connection from a business partner’s network.

The Check Point Solution

VPN SecuRemote/SecureClient allows you to connect to your organization in a secure manner, while at the same time protecting your machine from attacks that originate on the Internet. You can access private files over the Internet knowing that unauthorized persons cannot view the same file or alter it. With VPN SecuRemote/SecureClient, remote users connect to the organization using any network adapter (including wireless adapters) or modem dialup. Once both sides are sure they are communicating with the intended party, all subsequent communication is private (encrypted) and secure. This is illustrated in Figure 16-1:

Figure 16-1 SecureClient connecting to Site



How it Works

SecuRemote/SecureClient provides secure connectivity by authenticating the parties and encrypting the data that passes between them. To do this, VPN SecuRemote/SecureClient takes advantage of standard Internet protocols for strong encryption and authentication. Authentication means that both parties identify themselves correctly. Encryption ensures that only the authenticated parties can read the data passed between them. In addition, the integrity of the data is maintained, which means the data cannot be altered during transit.

SCV Granularity for VPN Communities

Access can be granted to specific hosts without being verified in order to allow the remote host to become fully compliant with the networks Security Policy. For example, if the Anti-Virus software is not up-to-date on a remote host, the gateway would normally block the connection entirely. However, access can be granted to the antivirus server in order to get the appropriate updates. After the updates are retrieved and installed on the remote host, it will pass the SCV check and get full access.

SCV granularity is supported for Simplified Mode configuration only.

Blocking Unverified SCV Connections

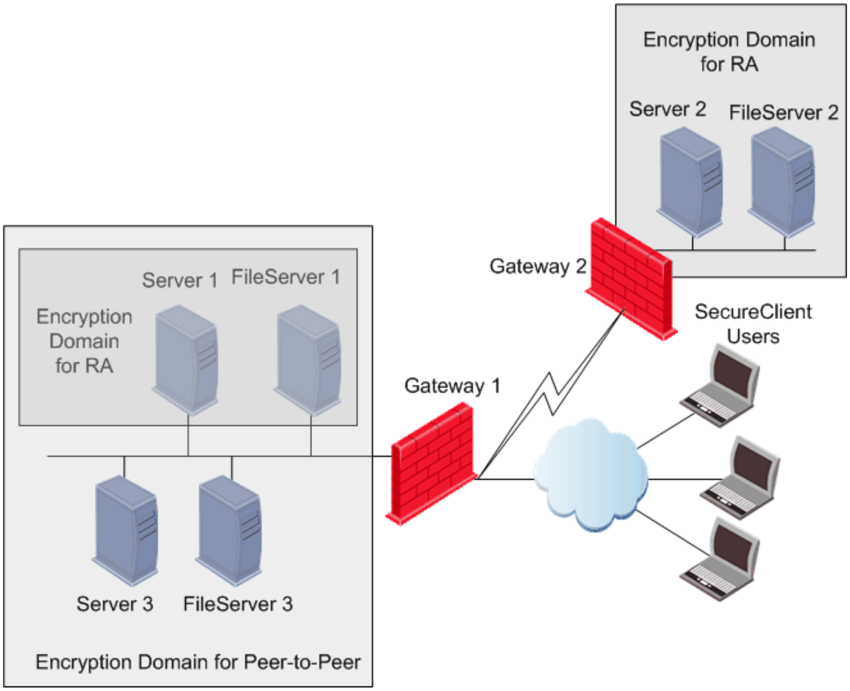
When a client becomes unverified, there is an option in the `local.scv` file to block connections that require verification: `block_scv_client_connections`. When this feature is active, and the client enters an unverified state, all SCV connections are blocked, even those which were opened during the time the client was verified. However, only SCV connections are blocked; that is, only those connections that require the client to be in a verified state. Other connections are not blocked.

Selective Routing

A VPN tunnel setup requires a configuration of a VPN domain for each participant gateway. The Selective Routing feature was designed to offer flexibility to define different encryption domains per VPN site-to-site communities and Remote Access (RA) Communities.

Remote Access VPN Dedicated Encryption Domain

Figure 16-2 Accessing Encryption domain for RA



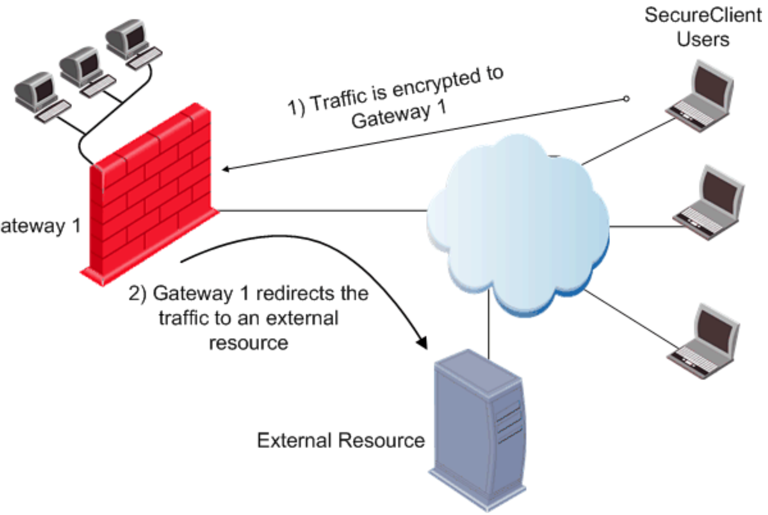
In this scenario:

- Gateways 1 & 2 are connected via a site-to-site VPN.
- Each gateway has its own encryption domain.
- Gateway 1 is also used by SecuRemote/SecureClient users.
- Using Selective Routing, a Remote Access (RA) encryption domain is configured on gateway 1 that will grant access only to Server 1 and FileServer 1.

In this case, the remote hosts are granted access to part of the encryption domain. SecuRemote/SecureClient users will only be able to access servers within the encryption domain that is permitted to them. The users will be denied access to Server 2 and FileServer 2.

Including External Resources in a Remote Access Encryption Domain

Figure 16-3 Accessing External Resources


SecureClient Users
1) Traffic is encrypted to Gateway 1
Gateway 1
2) Gateway 1 redirects the traffic to an external resource
External Resource

In this scenario:

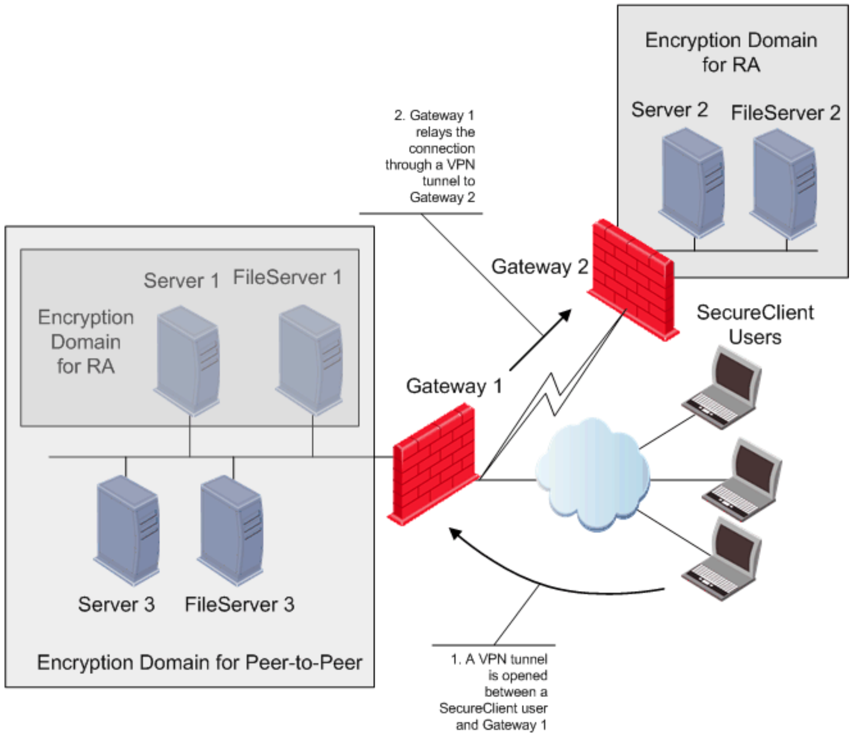
- SecureClient users connect to gateway 1.
- Gateway 1 has an encrypted domain that includes an external resource.
- Gateway 1 offers the SecureClient users access to external resources such as the Internet in addition to the VPN domain.

In the scenario depicted in Figure 16-3, an external resource is a part of the RA Encryption domain. Therefore, whenever the external resource is accessed by a remote host, the connection to that resource will be initiated by gateway 1.

The gateway also has the ability to transfer traffic from the SecureClient users to servers on the DMZ.

Providing Remote Access VPN to an External Encryption Domain

Figure 16-4 Accessing External Encryption Domain


Encryption Domain for RA
Server 2
FileServer 2
2. Gateway 1 relays the connection through a VPN tunnel to Gateway 2
Gateway 2
SecureClient Users
Server 1
FileServer 1
Encryption Domain for RA
Gateway 1
Server 3
FileServer 3
Encryption Domain for Peer-to-Peer
1. A VPN tunnel is opened between a SecureClient user and Gateway 1

In this scenario:

- Gateways 1 & 2 are connected via a site-to-site VPN.
- Each gateway has its own encryption domain.
- Gateway 1 is used by SecureClient users.

In this case, the encryption domain for remote users extends beyond one gateway. Gateway 1 relays SecureClients encrypted traffic destined to Server 2 and FileServer 2 which are located behind gateway 2. As a result, SecureClient users do not need to re-authenticate when accessing the resources behind gateway 2. This also allows for logging all the SecureClient activity to other resources behind other gateways.

Note - For remote hosts to successfully access resources behind gateway 1, either: all Office Mode IP's must be part of gateway 2's encryption domain, or Hide NAT must be enabled on gateway 1.

Desktop Security Policy

When is a Policy Downloaded?

When a user creates a site in SecureClient, a list of Policy Servers is downloaded to the client's machine. A policy will be automatically downloaded from a Policy Server when the SecureClient machine connects to the site. The automatic policy download can also be disabled — this configuration is controlled in the user's profile.

Policy Expiration and Renewal

The Desktop security policy is only valid for a certain period of time. After half of the period set has elapsed, the remote client queries the Policy Server for a renewal/update. The client tries to renew the current policy even if the previous renewal failed. If the renewal process continually fails, then the current Desktop Security Policy expires the remote client remains with the previous policy.

During the Security Policy update, the mobile users log files are being uploaded to the Policy Server.

Prepackaged Policy

SecureClient can be pre-packaged to include a default policy by:

1. Open SC tar.gz
2. Placing the policy files in the tar.gz directory (local.scv local.dt local.lp, etc.).
3. In the install section of `product.ini`, specifying `initialpolicy.bat`
4. Re-packaging the client using packing tool (or running setup from the `tar.gz`)
5. Installing SC from the generated package/tar.gz directory. The policy becomes active when the client is started for the first time.

Policy Server High Availability

When connecting to a gateway, you automatically logon to the Policy Server residing behind that gateway. If an alternative policy server was defined in the connection profile, you may logon to a Policy Server residing on another gateway by activating the Policy Server High Availability functionality by setting the `use_profile_ps_configuration` option as **true** in the `userc.c` file.

Wireless Hot Spot/Hotel Registration

Wireless Hotspot is a wireless broadband Internet access service available at public locations such as airport lounges, coffee shops and hotels.

When using Hotspot application, a user launches a web browser and attempts to connect to the Internet. When this occurs, the browser is automatically redirected by the Hotspot server to the Hotspot Welcome page for registration. during the registration process, the user fills in the required information. Once the registration is complete, the user may continue surfing the Internet.

Hotspot allows users with restrictive outbound policies and/or Hub Mode to register with Hotspot.

When a user selects to allow Hotspot, SecureClient modifies the desktop security policy and/or Hub Mode routing to enable Hotspot registration. This modification is restricted by time, number of IP addresses and ports. SecureClient records the IP addresses and ports that were accessed during the registration phase.

Enable Logging

Enabling logging will locally save all the activity on a remote host. This information is useful in tracking problems and troubleshooting. The information saved in the log files may contain confidential information and should only be sent back to the system administrator.

The Enable Logging feature can also be included in a *Prepackaged Policy*.

NAT Traversal Tunneling

The negotiation prior to the establishment of a VPN tunnel might result in the production of large packets. Some NAT devices may not fragment large packets correctly making the connection impossible. To resolve this issue, there are several methods that may be used:

- **NAT-T** - NAT-T is based on IETF RFC 3947 and 3948. When a remote user initiates a VPN session with a gateway, the remote host informs the gateway that it is able to communicate using NAT-T. During the initial negotiation, both peers attempt to detect whether the traffic passed through a NAT device. If a NAT device is detected between the peers, communication between them switches to UDP port 4500. NAT-T is not supported using Aggressive Mode. UDP port 4500 must be enabled which will be used for the entire VPN session.

  NAT-T is supported for Edge devices, L2TP clients and 3rd party gateways.
- **IKE over TCP** - IKE over TCP solves the problem of large UDP packets created during IKE phase I. The IKE negotiation is performed using TCP packets. TCP packets are not fragmented; in the IP header of a TCP packet, the DF flag ("do not fragment") is turned on. A full TCP session is opened between the remote host and the gateway for the IKE negotiation during phase I.
- **UDP Encapsulation** - This method adds a special UDP header that contains readable port information to the IPSec packet. The new port information is not the same as the original. The port number 2746 is included in both the source and destination ports. The NAT device uses the source port for the hide operation but the destination address and port number remains the same. When the peer gateway sees 2746 as the port number in the destination address, the gateway calls a routine to decapsulate the packet.

Idleness Detection

SecureClient has the ability to determine VPN traffic idleness while a user is connected using SecureClient to a Check Point gateway. A VPN tunnel is considered to be idle when no network traffic, which originated by an explicit action of a user, was sent over the VPN tunnel for a certain period of time.

In the event a VPN tunnel is considered to be idle, the user is notified that the VPN session is about to be disconnected.

An administrator can manually configure services, to be considered as implicit services which are not user initiated. Such services will be ignored by the idle detection mechanism.

Switching Modes

The VPN-1 SecureClient product has two views, compact and extended. The compact view is recommended for users that do not require multiple sites and profile management. The extended view offers profile management and multiple server definitions.

HTML Based Help

An HTML based user manual can be packaged in a SecureClient Package. The HTML help contains extensive help and graphics.

Configuring SecureClient

In This Chapter

Configuring SCV Granularity for VPN Communities

In SmartDashboard:

1. Click Policy > Global Properties.
2. Click [+] next to Remote Access to expand the branch and select Secure Configuration Verification (SCV).
3. Select the Apply SCV on Simplified Mode Security Policies checkbox and click the Exceptions button.

   The Hosts available without passing SCV verification appears.
4. Click Add to set the hosts and services to be excluded from SCV verification.

Configuring block_scv_client_connections

To block a user that becomes unverified, set the attribute `block_scv_client_connections` to *true* in the in the `local.scv file`. For more information, see "The local.scv Sets" on page 534.

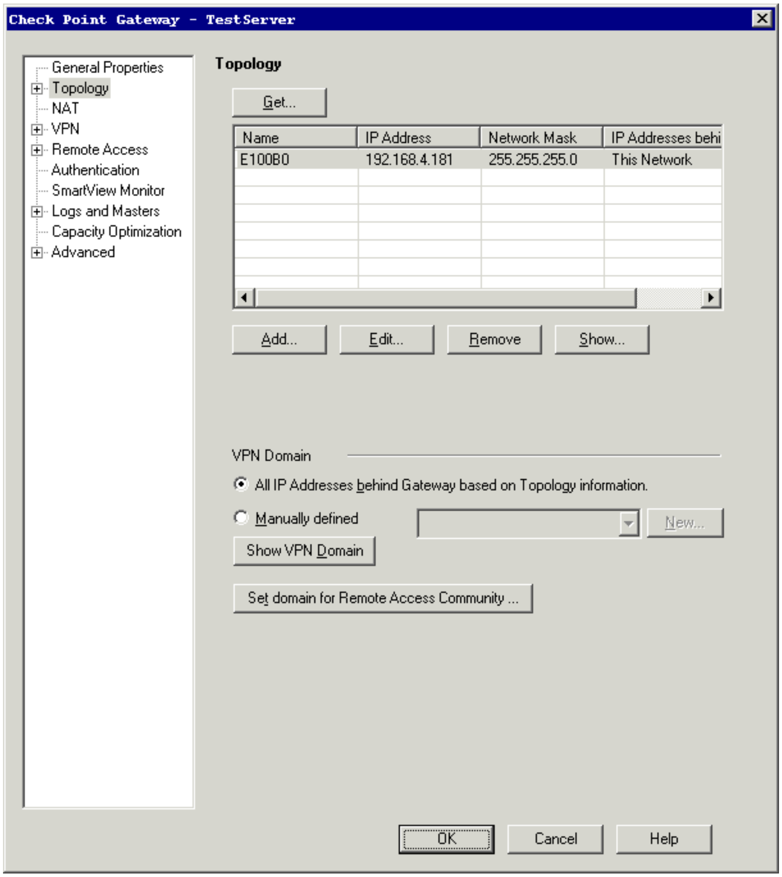
Configuring Selective Routing

From SmartDashboard, proceed as follows:

1. In the Network Objects Tree, highlight and right click the gateway to be edited.
2. Select **Edit.**

   The **Check Point Gateway** properties page appears
3. Select **Topology** to display the topology window.

Figure 16-5 Check Point Gateway Topology Window.



Check Point Gateway - TestServer

Topology

Name	IP Address	Network Mask	IP Addresses behi
E100B0	192.168.4.181	255.255.255.0	This Network

VPN Domain

● All IP Addresses behind Gateway based on Topology information.

○ Manually defined

4. Click the **Set domain for Remote Access Community** button.

   The **VPN Domain per Remote Access Community** window appears.
5. Click the **Set** button.

   The **Set VPN Domain per Remote Access Community** window appears.

6. From the drop down menu, select the object that will represent the Remote Access VPN domain.
7. Click **OK**.

Configuring Desktop Security Policy Expiration Time

1. In SmartDashboard, click **Policy > Global Properties**.

   The **Global Properties** window appears.
2. Select **Remote Access** to display the **Remote Access -VPN-1 SecuRemote/SecureClient** window.
3. In the **VPN-1 SecureClient - Desktop Security Policy expiration time** section, select the amount of time (in minutes) before the security policy will remain with the current policy.
4. Click **OK**.

Configuring Hot Spot/Hotel Registration

Enabling the Hotspot option is configured using the `userc.c` file. The Hotspot set (with defaults) is as follows:

```
:hotspot(
          :enabled (false)
          :log (false)
          :connect_timeout (600)
          :max_ip_count (5)
          :block_hotspot_after_connect (false)
          :max_trials (0)
          :local_subnets (false)
          :ports(
                                :(80)
                                :(443)
                                :(8080)
          )
     )
```

Table 16-1 Hotspot Parameters

Parameter	Default	Description
enabled	false	Set to **true** to enable a user to perform Hotspot registration
log	false	Set to **true** to send logs with the list of IP addresses and ports accessed during registration
connect_timeout	600	Maximum number of seconds to complete registration
max_ip_count	5	Maximum number of IP addresses allowed during registration
block_hotspot_after_connect	false	If set to **true** upon successful connect, the recorded ports and addresses will not remain open

Table 16-1 Hotspot Parameters

Parameter	Default	Description
max_trials	0	This value represents the maximum number of unsuccessful hotspot registration attempts that an end user may perform. Once this limit is reached, the user will not be allowed to attempt registration again. The counter is reset upon reboot, or upon a successful VPN connect. In addition, if you modify the `max_trials` value, the modification will take affect only upon successful connect, or reboot. If the `max_trials` value is set to 0, an unlimited number of trials is allowed
local_subnets	false	Restrict access to local subnets only
ports	80 443 8080	Restrict access to specific ports

Configuring Enable Logging

Enable Logging is configured in SmartDashboard and SecuRemote/SecureClient.

In SmartDashboard:

1. Go to **Global Properties > Remote Access > VPN - Advanced**.
2. Select **Allow users to save troubleshooting logs**.
3. Click **OK**.

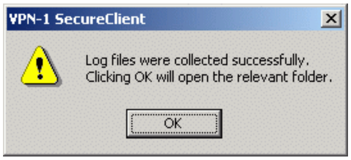
In the system tray of the desktop:

1. Right click the SecureClient icon.

   From the popup menu, select **Settings**.

2. On the **Advanced** tab, select **Enable Logging** and click **Save Logs**.

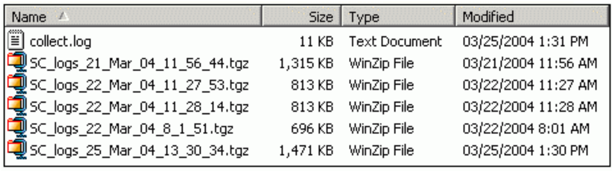
   Wait until the following message appears:



VPN-1 SecureClient

Log files were collected successfully.
Clicking OK will open the relevant folder.

3. Save the logs to the default location:



Name	Size	Type	Modified
collect.log	11 KB	Text Document	03/25/2004 1:31 PM
SC_logs_21_Mar_04_11_56_44.tgz	1,315 KB	WinZip File	03/21/2004 11:56 AM
SC_logs_22_Mar_04_11_27_53.tgz	813 KB	WinZip File	03/22/2004 11:27 AM
SC_logs_22_Mar_04_11_28_14.tgz	813 KB	WinZip File	03/22/2004 11:28 AM
SC_logs_22_Mar_04_8_1_51.tgz	696 KB	WinZip File	03/22/2004 8:01 AM
SC_logs_25_Mar_04_13_30_34.tgz	1,471 KB	WinZip File	03/25/2004 1:30 PM

NOTE: The default location is a hidden folder in windows. If you need to locate this folder, then in **Control panel > Folder Options > View** select **Show hidden files and folders**.

4. Close the location window. The file has been saved automatically.

Including Enabling Logging in a Prepackaged Policy

In the `[install]` section of the `product.ini` file add one of the following commands to either enable or disable the Enable Logging feature in a prepackaged policy:

- `logging.bat enable`
- `logging.bat disable`

Configuring NAT Traversal

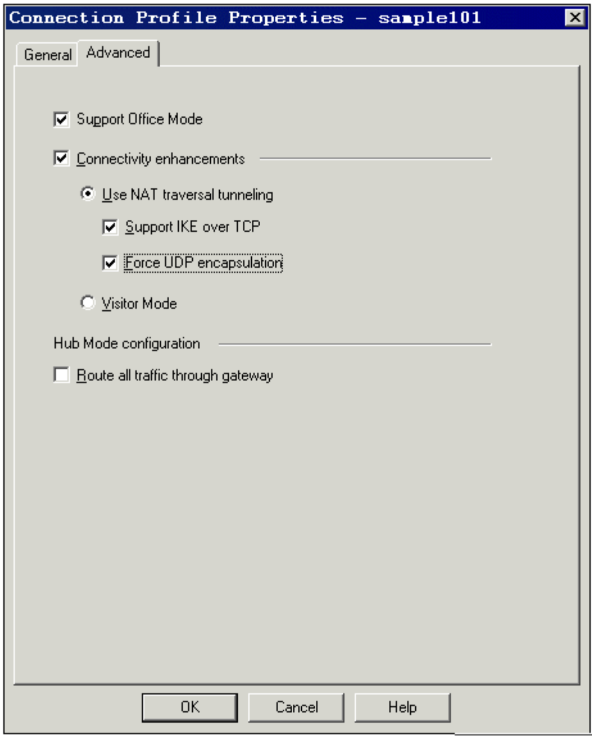
In SmartDashboard:

1. Click **Manage > Remote Access > Connection Profiles**.

   The **Connection Profiles** window appears.

2. Select Connection Profile and click **Edit.**
3. In the **Advanced** tab, click **Connectivity Enhancements**.

Figure 16-6 Connection Profile Properties - Advanced Tab



Connection Profile Properties – sample101

☑ Support Office Mode

☑ Connectivity enhancements

● Use NAT traversal tunneling

☑ Support IKE over TCP

☑ Force UDP encapsulation

○ Visitor Mode

Hub Mode configuration

□ Route all traffic through gateway

4. Select **Use NAT traversal tunneling**.
5. Select **Support IKE over TCP** and/or **Force UDP Encapsulation**.
6. Click **OK**.

Enable/Disable Switching Modes

In the `userc.c` file, set the flag `enable_mode_switching` to true.

Add HTML Help to Package

1. Open the `.tgz` distribution of SecuRemote/SecureClient.
2. Add `SR_HELP.TGZ` to the directory in which you have opened the `.tgz`.
3. Specify `sc_help_install.bat` in the install section of `product.ini` and `product.ini.simp`.
4. Re-package using packaging tool. See: "Creating a Preconfigured Package" on page 483.

Configuring Idle Detection

Configuring the idleness_detection Property

Traffic Idleness Detection is configured with the `idleness_detection` property using DBedit.

Table 16-2 Property Definition

Property Name		Default Value	Valid Values
`active`	Used to enable or disable the feature	true	true, false
`timeout`	Defines the time, in minutes, that will pass by without traffic until the VPN tunnel is considered as to have been "idled-out".	30	any positive integer value
`excluded services`	Defines a list of services that can be configured by the administrator, which will not be considered as user initiated.	None	`<port number>` - an integer denoting a port number whose traffic will be excluded. `"<port number, protocol>"` - a combination of port number and protocol for more specific exclusion. Note this has to be within quotes. `<icmp>` - the icmp protocol will be excluded on all ports.

Table 16-3 Sample Configuration Set

```
:idleness_detection (
                     :excluded services (
                               :(53)
                               :(“750,tcp”)
                               :(icmp)
                      )
                     :active (true)
                     :timeout (30)
)
```

Note - : SecureClient Control connections are automatically ignored by the idle detections mechanism. As a result, these items do not need to be configured separately.

Chapter 17

Endpoint Connect

In This Chapter:

This chapter explains how to configure the security gateway to work with the Endpoint Connect client.

Introduction

Endpoint Connect is Check Point's new lightweight remote access client. Providing seamless, secure (IPSec and SSL) VPN connectivity to corporate resources, the client works transparently with security gateways and Connectra, the Check Point remote access gateway solution.

Why Endpoint Connect?

With their requirement to repeatedly reconnect and authenticate to the corporate gateway, traditional IPSec clients can be slow and cumbersome. Even SSL VPNs with their explicit login requirements through a browser, are a less than optimal solution for highly mobile laptop users.

Providing a highly secured, low footprint VPN technology with advanced security scanning capabilities, Endpoint Connect uses intelligent Auto-Connect and roaming technologies to facilitate seamless and transparent interaction with the gateway at the perimeter of the corporate network.

Designed for corporate users who prefer to use their native desktop to launch business applications rather than the Connectra SSL portal, Endpoint Connect users do not have to authenticate each time they connect. Through interface roaming technologies, client users are always connected to the resources available behind the gateway. As corporate users move around, an auto connect mode discovers whether users are outside of a secure environment, and implements the best way to connect, using either NAT-T or Visitor Mode. In practical terms, if client users outside of the internal network open their mail programs a connection is transparently established to the mail server behind the gateway. If client users have mapped drives to servers on the internal network, those mapped drives remain functional even as users roam in and out of the network.

Note - While Endpoint Connect can reside on the same host with SecureClient or Endpoint Security, users should avoid connecting with the two VPN clients to the same network at the same time

Capabilities

Resident on the users desktop or laptop, Endpoint Connect provides various capabilities for connectivity, security, installation and administration.

Connectivity

- **Network Layer Connectivity**

  An IPSec VPN connection to the gateway for secure encrypted communication. If the network connection is lost, the client seamlessly reconnects without user intervention.

- **Intelligent Auto detect and connect**

  Whenever the gateway or client’s location changes, Endpoint Connect autodetects the best method to establish a connection, using either NAT-T or Visitor mode, intelligently auto-switching between the two modes as necessary.

- **Smart location awareness**

  Endpoint Connect intelligently detects whether it is inside or outside of the VPN domain (Enterprise LAN), and automatically connects or disconnects as required.

- **Proxy detection**

  Proxy servers between the client and the gateway are automatically detected, authenticated to, and replaced when no longer valid.

- **Transparent Network and Interface Roaming**

  If the IP address of the client changes, for example if the client is using a wireless connection then physically connects to a LAN that is not part of the VPN domain, interface roaming maintains the logical connection.

- **Multiple Sites**

  Endpoint Connect connects to any one of a number of user defined gateways.

- **Dead Gateway Detection**

  If the client fails to receive an encrypted packet within a specified time interval, it sends a special "tunnel test" packet to the gateway. If the tunnel test packet is acknowledged, then the gateway is active. If number of tunnel test packets remain unacknowledged, the gateway is considered inactive or dead.

- **Hotspot Detection and Exclusion**
- **Dialup Support**
- **Cooperative Enforcement**

Security

- **Endpoint Security on Demand**

  Provides a full, effective end point compliance check (for required software updates, Anti-Virus signatures, presence of malware) when connecting, and repeat scans at specified time intervals. Clients that fail the initial scan when connecting gain access to remediation sources.

- **Full IPSec VPN**

  Internet Key Change (version 1) support for secure authentication.

- **Support for strong authentication schemes** such as:

  a. Username and passwords (including cached passwords)

  b. SecurID

  c. Challenge-Response

  d. CAPI software and hardware tokens

- **Certificate enrollment, renewal, and auto Renewal**

- **Tunnel idleness Detection**
- **Smartcard Removal Detection**
- **Hub Mode**

  Increases security by routing all traffic, such as traffic to and from the Internet, through the gateway, where the traffic can be inspected for malicious content before being passed to the client.
- **Visitor Mode**

  When the client needs to connect through a gateway that limits connections to port 80 or 443, encrypted (IPSec) traffic between the client and the gateway is tunneled inside a regular TCP connection.

64-bit Support

The Endpoint Connect Installation package determines whether the underlying operating system is 32 or 64-bit and installs the appropriate drivers.

Installation and Use

- **Small footprint**
- **Offline and Web deployment**

  Endpoint Connect is easily distributed through the Connectra portal.
- **Automatic upgrades**

  Endpoint Connect upgrades are automatic, transparent to the user, and do not require administrator privileges or a client reboot.
- **Site and Create New Site connection wizards**

  For quickly configuring connections to corporate resources.
- **CLI Scripting**

  For automation and internal testing, and use as an embedded “headless” client.
- **OPSEC API**

  Available for embedded applications, Endpoint Connect is also designed to be part of specialized customer integrations and deployments, for example, organizations that build their own corporate presence applications that require VPN components. The client’s intelligent auto-detect and disconnect features make it ideal for remote unmanned devices that need multiple High Availability options, such as embedded Windows ATMs. For such scenarios, Endpoint Connect offers a native Command Line Interface and OPSec API for configuration and monitoring, as well as the ability to be installed and run as a service.

Administration

- **Unified Central Management**
- **Advanced User Management**
- **Unified updates**
- **Regulatory Compliance with Advanced Monitoring, Logging and Reporting**

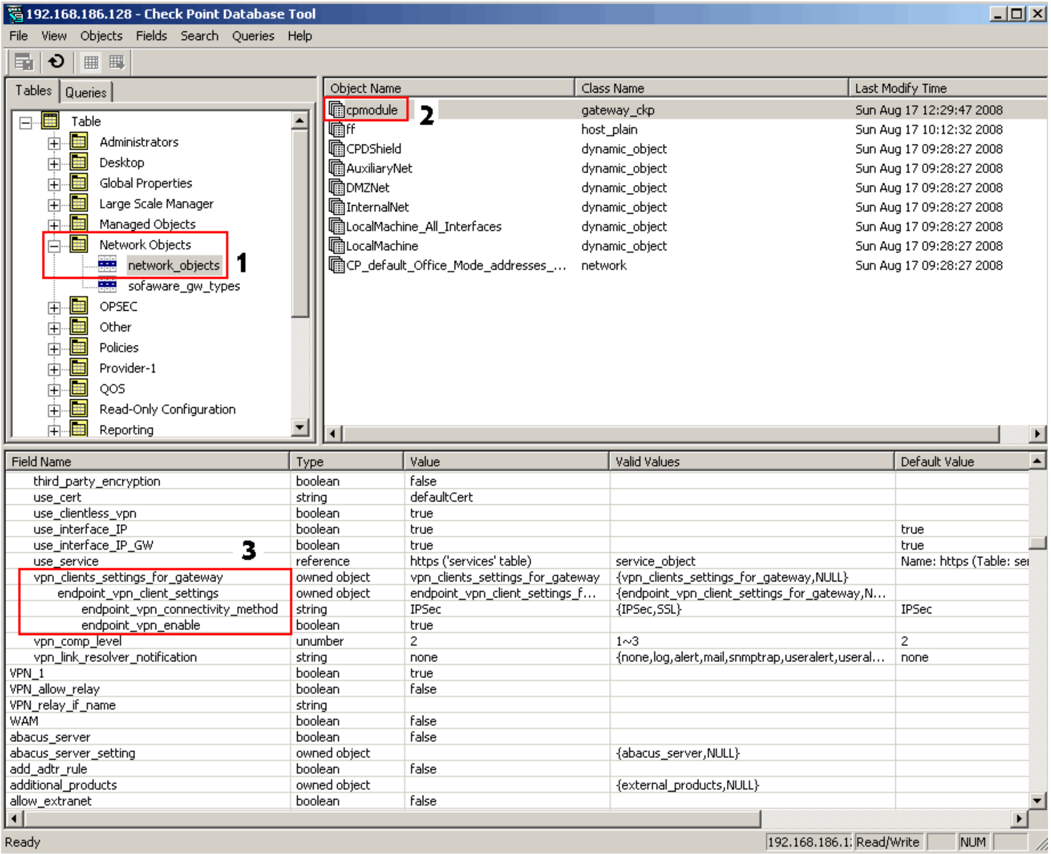
  DLL version numbers collected in a special file for troubleshooting purposes.

Enabling Endpoint Connectivity

To enable Endpoint Connect connectivity with the gateway:

1. Open **GuiDBedit**, and connect to Security Management server, as shown in Figure 17-7:

Figure 17-7 GuiDBedit



Object Name	Class Name	Last Modify Time
cpmodule	gateway_ckp	Sun Aug 17 12:29:47 2008
ff	host_plain	Sun Aug 17 10:12:32 2008
CPDShield	dynamic_object	Sun Aug 17 09:28:27 2008
AuxiliaryNet	dynamic_object	Sun Aug 17 09:28:27 2008
DMZNet	dynamic_object	Sun Aug 17 09:28:27 2008
InternalNet	dynamic_object	Sun Aug 17 09:28:27 2008
LocalMachine_All_Interfaces	dynamic_object	Sun Aug 17 09:28:27 2008
LocalMachine	dynamic_object	Sun Aug 17 09:28:27 2008
CP_default_Office_Mode_addresses_...	network	Sun Aug 17 09:28:27 2008

Field Name	Type	Value	Valid Values	Default Value
third_party_encryption	boolean	false		
use_cert	string	defaultCert		
use_clientless_vpn	boolean	true		
use_interface_IP	boolean	true		true
use_interface_IP_GW	boolean	true		true
use_service	reference	https ('services' table)	service_object	Name: https (Table: sei
vpn_clients_settings_for_gateway	owned object	vpn_clients_settings_for_gateway	{vpn_clients_settings_for_gateway,NULL}	
endpoint_vpn_client_settings	owned object	endpoint_vpn_client_settings_f...	{endpoint_vpn_client_settings_for_gateway,N...	
endpoint_vpn_connectivity_method	string	IPSec	{IPSec,SSL}	IPSec
endpoint_vpn_enable	boolean	true		
vpn_comp_level	unumber	2	1~3	2
vpn_link_resolver_notification	string	none	{none,log,alert,mail,snmptrap,useralert,useral...	none
VPN_1	boolean	true		
VPN_allow_relay	boolean	false		
VPN_relay_if_name	string			
WAM	boolean	false		
abacus_server	boolean	false		
abacus_server_setting	owned object		{abacus_server,NULL}	
add_adtr_rule	boolean	false		
additional_products	owned object		{external_products,NULL}	
allow_extranet	boolean	false		

2. On the **Tables** tab, select **Network Objects**.

3. In the **Object Name** window, select the object that represents the R65 gateway. (cpmodule in Figure 17-7).

4. In the **Field Name** table:

   a. Locate the **vpn_clients_settings_for_gateway**

   b. Select the row and right-click

   c. Select **Edit**, and click **OK** to confirm the setting.

   d. Locate the **endpoint_vpn_client_settings_for_gateway** property
   e. Select the row and right-click
   f. Select **Edit** and click **OK** to confirm the setting.
   g. Select the **endpoint_vpn_connectivity_method** property
   h. Select **Edit** and change the value to **IPSEC**
   i. Select the **endpoint_vpn_enable** property
   j. Select **Edit** and change the value to **True**.
   k. Save changes.
   l. Exit GuiDBEdit.

5. Open SmartDashboard.

   You will be prompted to download a new version of SmartDashboard.

6. Using the new SmartDashboard, configure the gateway for Endpoint connectivity. (See "Configuring the Gateway Using SmartDashboard" on page 358)

7. Install the policy.

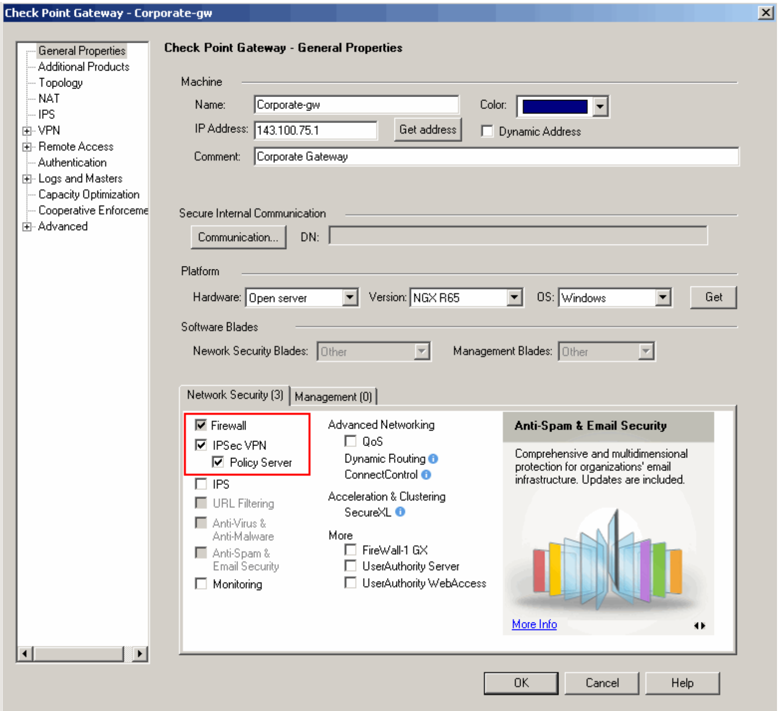
Configuring the Gateway Using SmartDashboard

In This Section:

Obligatory Settings

In SmartDashboard:

1. Open the **General Properties** window for the gateway.
2. Enable VPN:



Check Point Gateway - Corporate-gw

Check Point Gateway - General Properties

Machine

Name: Corporate-gw

Color:

IP Address: 143.100.75.1 Get address ☐ Dynamic Address

Comment: Corporate Gateway

Secure Internal Communication

Communication... DN:

Platform

Hardware: Open server Version: NGX R65 OS: Windows Get

Software Blades

Nework Security Blades: Other Management Blades: Other

Network Security (3) | Management (0)

☑ Firewall
☑ IPSec VPN
☑ Policy Server
☐ IPS
☐ URL Filtering
☐ Anti-Virus & Anti-Malware
☐ Anti-Spam & Email Security
☐ Monitoring

Advanced Networking
☐ QoS
Dynamic Routing
ConnectControl

Acceleration & Clustering
SecureXL

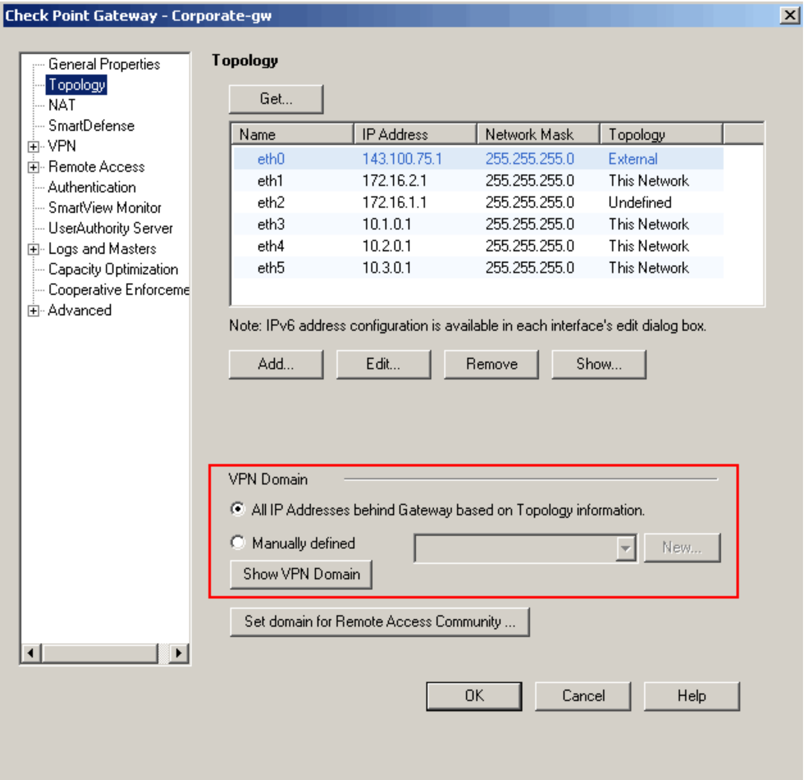
More
☐ FireWall-1 GX
☐ UserAuthority Server
☐ UserAuthority WebAccess

Anti-Spam & Email Security

Comprehensive and multidimensional protection for organizations' email infrastructure. Updates are included.

More Info

3. On the **Topology** page, create a VPN domain:



Topology

Name	IP Address	Network Mask	Topology
eth0	143.100.75.1	255.255.255.0	External
eth1	172.16.2.1	255.255.255.0	This Network
eth2	172.16.1.1	255.255.255.0	Undefined
eth3	10.1.0.1	255.255.255.0	This Network
eth4	10.2.0.1	255.255.255.0	This Network
eth5	10.3.0.1	255.255.255.0	This Network

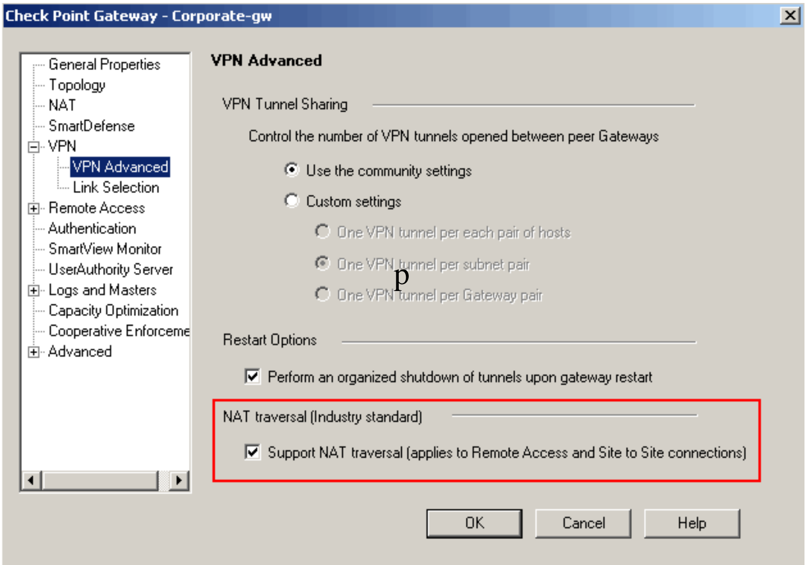
Note: IPv6 address configuration is available in each interface's edit dialog box.

VPN Domain

● All IP Addresses behind Gateway based on Topology information.

○ Manually defined

4. On the **VPN > VPN Advanced** page, enable NAT Traversal:



VPN Advanced

VPN Tunnel Sharing

Control the number of VPN tunnels opened between peer Gateways

● Use the community settings

○ Custom settings

○ One VPN tunnel per each pair of hosts

● One VPN tunnel per subnet pair

○ One VPN tunnel per Gateway pair

Restart Options

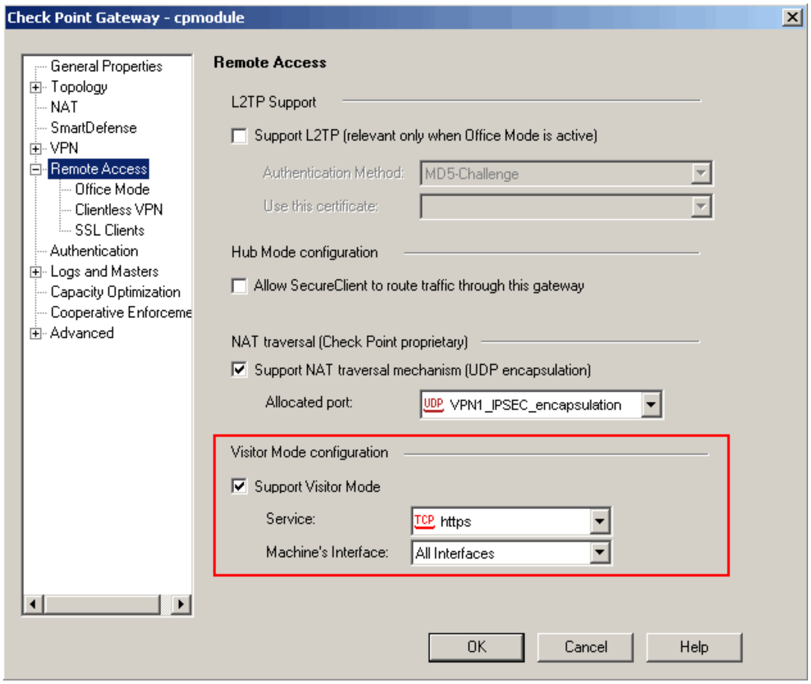
☑ Perform an organized shutdown of tunnels upon gateway restart

NAT traversal (Industry standard)

☑ Support NAT traversal (applies to Remote Access and Site to Site connections)

See “NAT Traversal” on page 378 for additional information.

5. On the **Remote Access** Page, enable Visitor Mode:



Remote Access

L2TP Support

☐ Support L2TP (relevant only when Office Mode is active)

Authentication Method: MD5-Challenge

Use this certificate:

Hub Mode configuration

☐ Allow SecureClient to route traffic through this gateway

NAT traversal (Check Point proprietary)

☑ Support NAT traversal mechanism (UDP encapsulation)

Allocated port: UDP VPN1_IPSEC_encapsulation

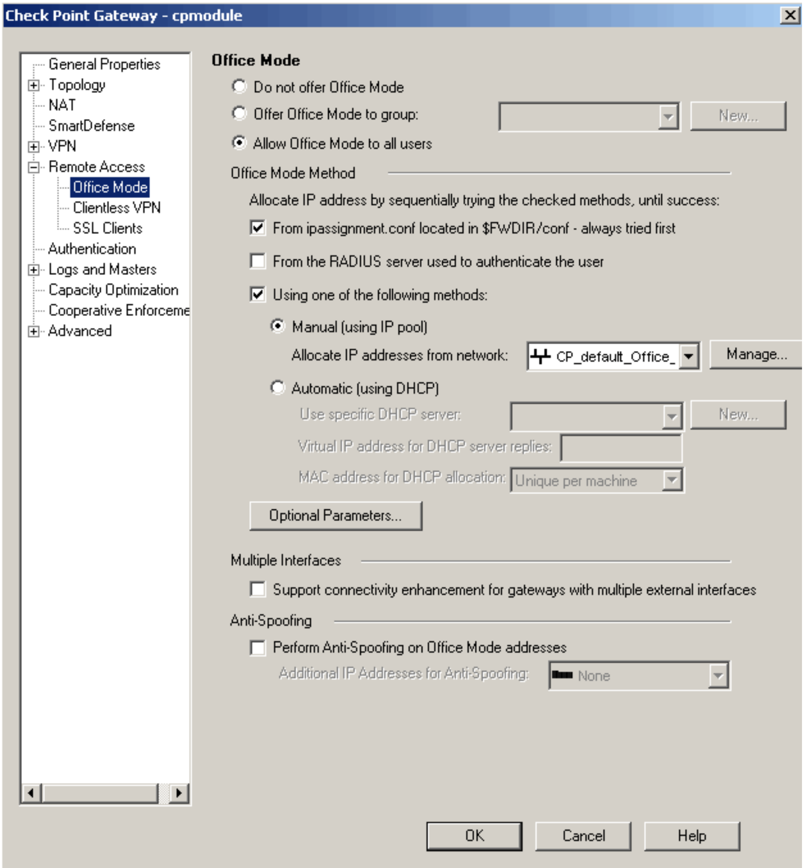
Visitor Mode configuration

☑ Support Visitor Mode

Service: TCP https

Machine's Interface: All Interfaces

6. On the **Remote Access > Office Mode** page enable Office Mode and configure the appropriate settings:



Office Mode

○ Do not offer Office Mode

○ Offer Office Mode to group:

● Allow Office Mode to all users

Office Mode Method

Allocate IP address by sequentially trying the checked methods, until success:

☑ From ipassignment.conf located in $FWDIR/conf - always tried first

□ From the RADIUS server used to authenticate the user

☑ Using one of the following methods:

● Manual (using IP pool)

Allocate IP addresses from network: CP_default_Office_

○ Automatic (using DHCP)

Use specific DHCP server:

Virtual IP address for DHCP server replies:

MAC address for DHCP allocation: Unique per machine

Multiple Interfaces

□ Support connectivity enhancement for gateways with multiple external interfaces

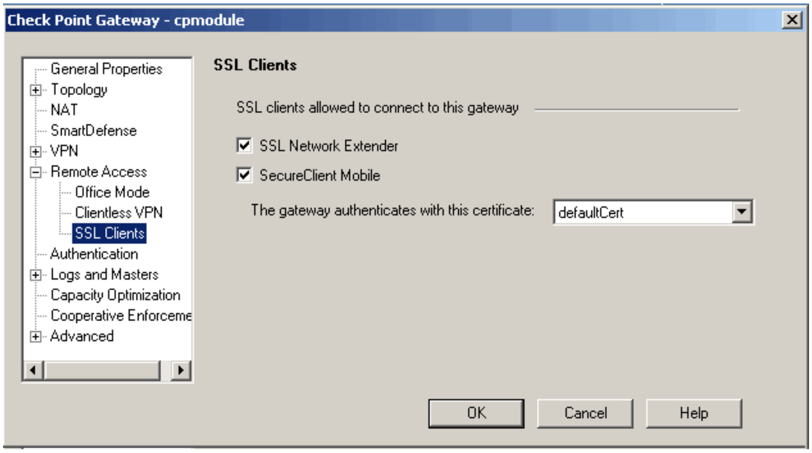
Anti-Spoofing

□ Perform Anti-Spoofing on Office Mode addresses

Additional IP Addresses for Anti-Spoofing: None

Enable Office Mode when the remote client may be working with an IP address that clashes with an IP address on the network *behind* the gateway. When working with Office Mode, Endpoint Connect takes an Office IP address from the same reserved pool of IP addresses as SecureClient Mobile or SSL Network Extender.

7. On the **Remote Access > SSL Clients** page enable SSL clients:



SSL Clients

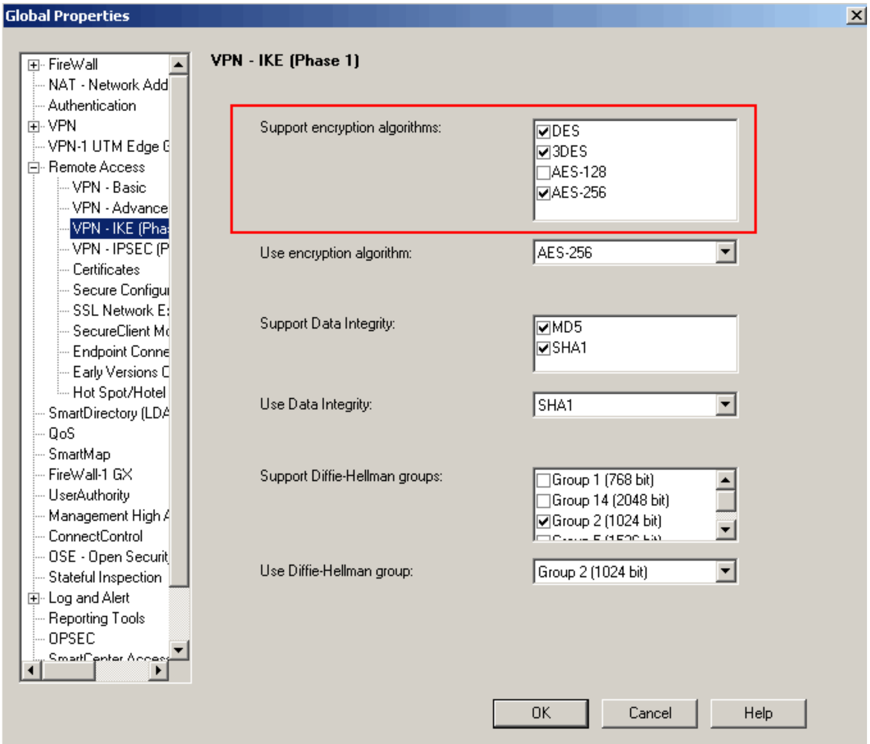
SSL clients allowed to connect to this gateway

☑ SSL Network Extender

☑ SecureClient Mobile

The gateway authenticates with this certificate: defaultCert

8. Endpoint Connect does not support the DES encryption algorithm. If you have a gateway configured to support only the DES encryption algorithm, then reconfigure the settings in **Global Properties > Remote Access VPN-IKE (phase1)**:



VPN - IKE (Phase 1)

Support encryption algorithms:
☑ DES
☑ 3DES
□ AES-128
☑ AES-256

Use encryption algorithm: AES-256

Support Data Integrity:
☑ MD5
☑ SHA1

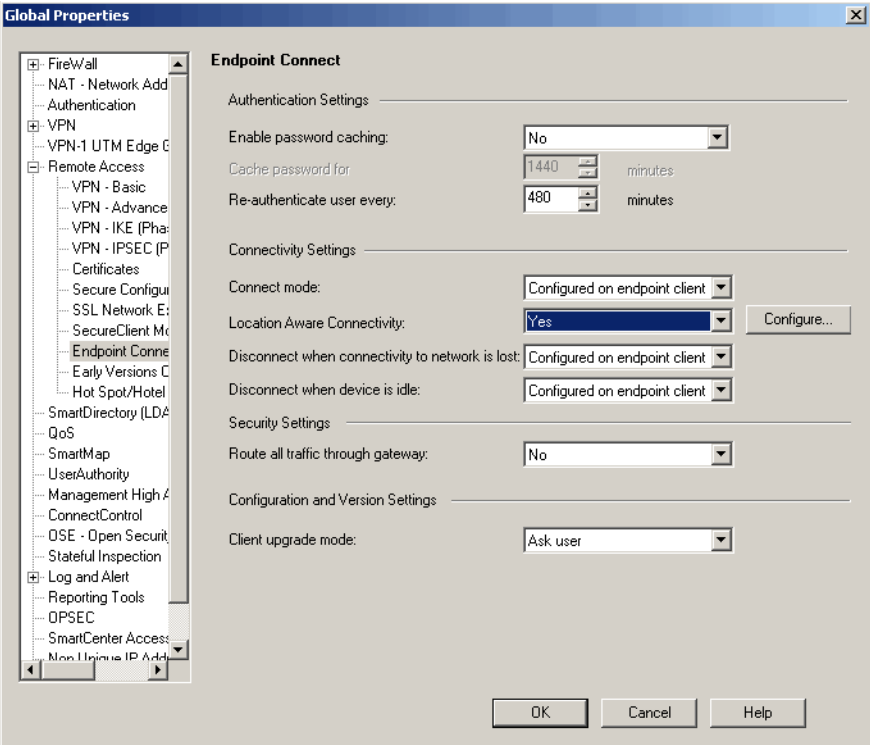
Use Data Integrity: SHA1

Support Diffie-Hellman groups:
□ Group 1 (768 bit)
□ Group 14 (2048 bit)
☑ Group 2 (1024 bit)
□ Group 5 (1536 bit)

Use Diffie-Hellman group: Group 2 (1024 bit)

Endpoint Connect Advanced Settings

1. In SmartDashboard, open **Global Properties > Remote Access > Endpoint Connect**:



Global Properties

Endpoint Connect

Authentication Settings

Enable password caching: No

Cache password for 1440 minutes

Re-authenticate user every: 480 minutes

Connectivity Settings

Connect mode: Configured on endpoint client

Location Aware Connectivity: Yes

Disconnect when connectivity to network is lost: Configured on endpoint client

Disconnect when device is idle: Configured on endpoint client

Security Settings

Route all traffic through gateway: No

Configuration and Version Settings

Client upgrade mode: Ask user

The window is divided into four sections:

- **Authentication Settings**
- **Connectivity Settings**
- **Security Settings**
- **Configuration and Version Settings**

Authentication Settings

Use the settings in this section to configure password caching, and how often the user needs to re-authenticate. If you do not open this window and configure options, then the client's default value takes affect:

Table 17-4 Default Authentication values

Option	SmartDashboard default value	Endpoint Connect default value
Enable password caching	No	No
Cache password for	1440 (minutes)	1440
Reauthenticate user every	480 (minutes)	480

Connectivity Settings

Use the settings in this section to determine connect and disconnect options. **Connect mode** covers whether the user should manually connect each time, the user is always connected, or whether the decision can be made on the client side. If the decision is left to the client, the user can select the **Enable Always Connect** option on the **Settings** tab of the site properties window.

If you do not open this window, then default values apply:

Table 17-5 Default Connectivity values

Option	SmartDashboard default value	Endpoint Connect default value
Connect mode	Client decide	Yes
Location Aware Connectivity	Client decide	Yes
Disconnect when no connectivity to network	Client decide	No
Disconnect when device is idle	Client decide	No

Location Aware Connectivity

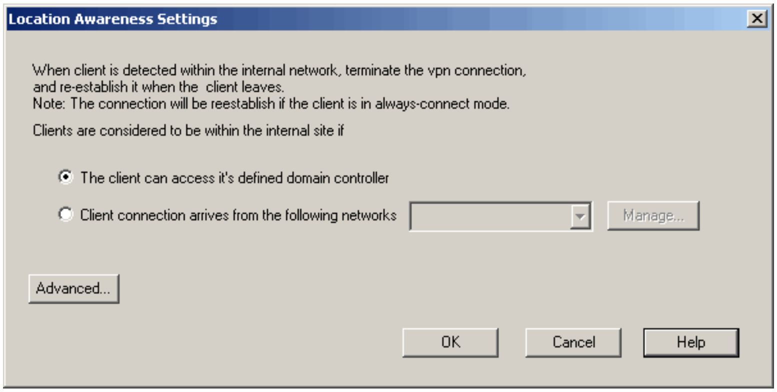
Endpoint Connect intelligently detects whether it is inside or outside of the VPN domain (Enterprise LAN), and automatically connects or disconnects as required. When the client is detected within the internal network, the VPN connection is terminated. If the client is in **Always-Connect** mode, the VPN connection is established again when the client exits.

Configuring Location Aware Connectivity

To configure Location Aware Connectivity:

a. Select **Yes** from the drop-down box and click **Configure...**.

The **Location Awareness Settings** window opens:



When client is detected within the internal network, terminate the vpn connection,
and re-establish it when the client leaves.
Note: The connection will be reestablish if the client is in always-connect mode.

Clients are considered to be within the internal site if

● The client can access it's defined domain controller

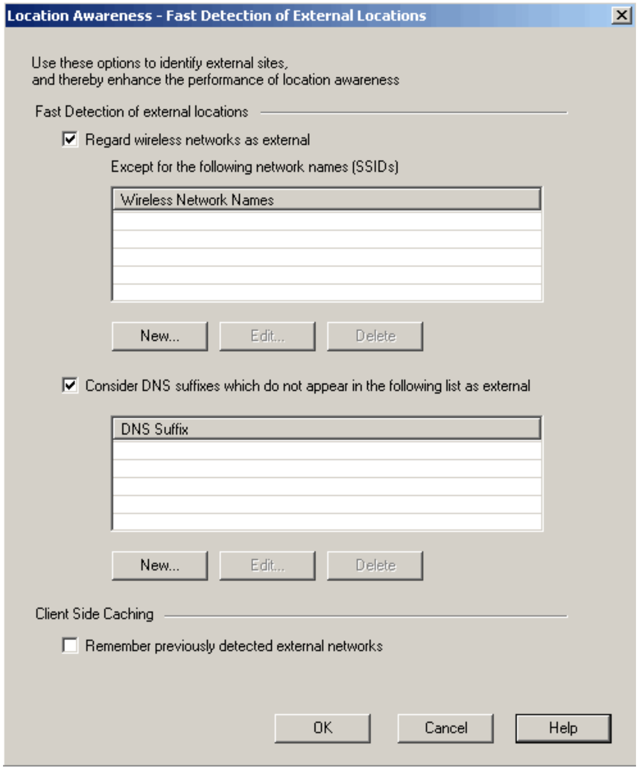
○ Client connection arrives from the following networks

b. Select **Client connection arrives from within the following network**

c. Click **Manage** and select or create a group that contains the relevant IP range.

d. Click **Advanced...**.

The **Location Awareness - Fast Detection of External Locations** window opens:



Use these options to identify external sites, and thereby enhance the performance of location awareness

Fast Detection of external locations

☑ Regard wireless networks as external

Except for the following network names (SSIDs)

☑ Consider DNS suffixes which do not appear in the following list as external

Client Side Caching

☐ Remember previously detected external networks

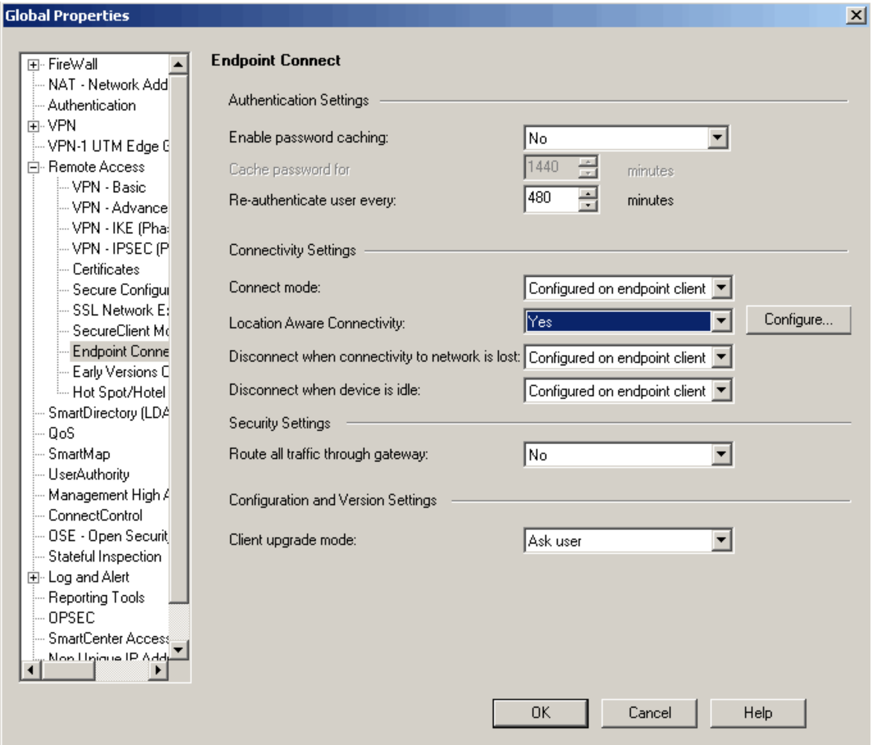
Use these options to identify external networks. For example, create a list of wireless networks or DNS suffixes that are known to be external. Or cache (on the client side) names of networks that were previously determined to be external. Selecting one or more of these options enhances the performance of location awareness.

Interface-based Location Awareness.

In this kind of location awareness, when the firewall receives a request from the remote client, the firewall uses the IP address and port number of the client together with the IP address and port of the gateway, along with the protocol type, to determine via the connection table on the gateway whether the interface is internal or external. If the firewall is unable determine this, or if the interface is listed as both internal and external, the connection is encrypted.

To configure Interface-based location awareness:

1. In SmartDashboard, open **Global Properties > Remote Access > Endpoint Connect**:



Endpoint Connect

Authentication Settings

Enable password caching: No

Cache password for 1440 minutes

Re-authenticate user every: 480 minutes

Connectivity Settings

Connect mode: Configured on endpoint client

Location Aware Connectivity: Yes

Disconnect when connectivity to network is lost: Configured on endpoint client

Disconnect when device is idle: Configured on endpoint client

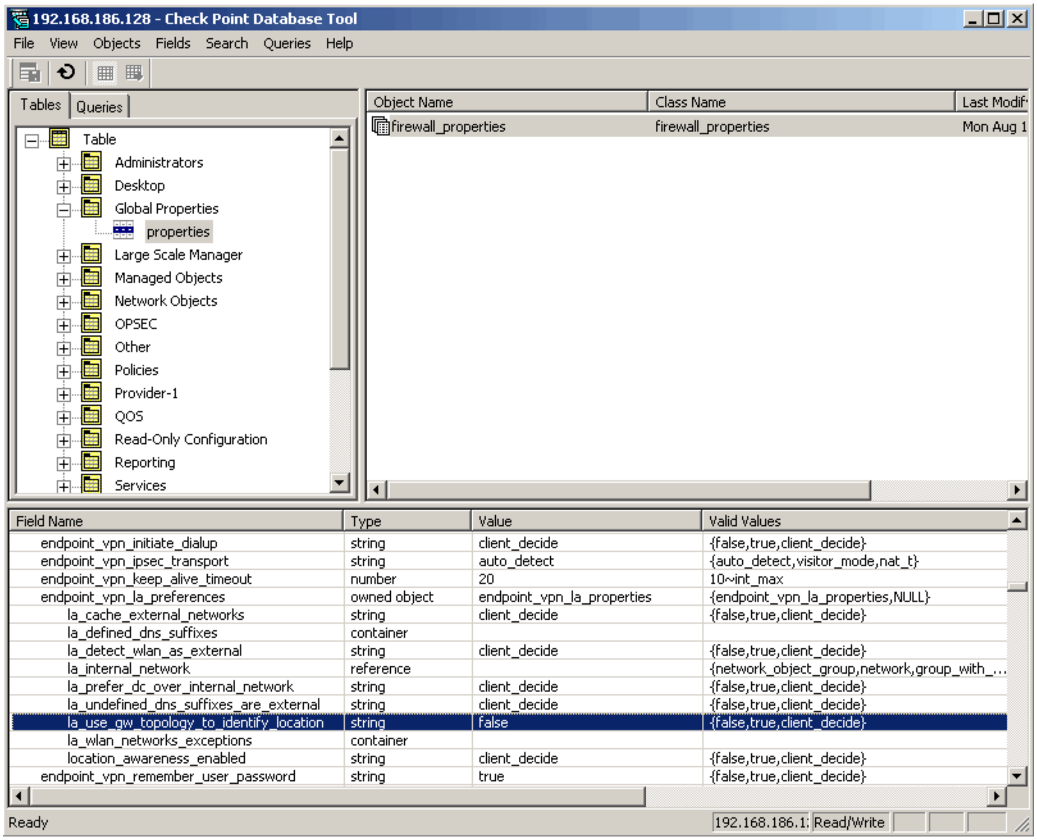
Security Settings

Route all traffic through gateway: No

Configuration and Version Settings

Client upgrade mode: Ask user

2. For **Location Aware Connectivity**, select **Yes** from the drop-down box.
3. Save the policy and close SmartDashboard.
4. Connect to Security Management server using **GuiDBedit**, the Check Point Database tool.
5. On the **Tables** tab, under **Global Properties > properties > firewall_properties** locate the **la_use_gw_topology_to_identify_location** property:



Object Name	Class Name	Last Modif
firewall_properties	firewall_properties	Mon Aug 1

Field Name	Type	Value	Valid Values
endpoint_vpn_initiate_dialup	string	client_decide	{false,true,client_decide}
endpoint_vpn_ipsec_transport	string	auto_detect	{auto_detect,visitor_mode,nat_t}
endpoint_vpn_keep_alive_timeout	number	20	10~int_max
endpoint_vpn_la_preferences	owned object	endpoint_vpn_la_properties	{endpoint_vpn_la_properties,NULL}
la_cache_external_networks	string	client_decide	{false,true,client_decide}
la_defined_dns_suffixes	container		
la_detect_wlan_as_external	string	client_decide	{false,true,client_decide}
la_internal_network	reference		{network_object_group,network,group_with_...
la_prefer_dc_over_internal_network	string	client_decide	{false,true,client_decide}
la_undefined_dns_suffixes_are_external	string	client_decide	{false,true,client_decide}
la_use_gw_topology_to_identify_location	string	false	{false,true,client_decide}
la_wlan_networks_exceptions	container		
location_awareness_enabled	string	client_decide	{false,true,client_decide}
endpoint_vpn_remember_user_password	string	true	{false,true,client_decide}

6. Change the value of the property to: **TRUE**.
7. Save and close.
8. Install a policy.

Security Settings

Use the settings in this section to determine whether or not traffic to and from Endpoint Connect is routed through the gateway, and therefore subject to content inspection.

- If the system administrator decides to **Route all traffic through gateway**, all outbound traffic on the client is encrypted and sent to the gateway but only traffic directed at site resources is passed through; all other traffic is dropped.
- If this option is *not* selected, only traffic directed at site resources is encrypted and sent to the gateway. All other outbound client traffic passes in the clear.

- For the gateway to act as a hub for content inspection of all inbound and outbound client traffic, regardless of destination, the administrator needs to a define a network application that includes the range: **0.0.0.1 > 255.255.255.254**.

  If you do not open this window, then default values apply:

Table 17-6 Default Security Settings

Option	SmartDashboard default value	Endpoint Connect default value
Route all traffic through gateway	No	No

Configuration and Version Settings

Use the settings in this section to configure how the client is upgraded. The upgrade procedure remains transparent to the user, and does not require administrator privileges on the endpoint or a reboot after the upgrade is complete.

If you do not open this window, then default values apply:

Table 17-7 Default configuration and version settings

Option	SmartDashboard default value	Endpoint Connect default value
Client upgrade mode	ask user	do not upgrade

9. Save the policy.

Configuring Optional Endpoint Connect Settings

In This Section:

If you are considering migrating SecureClient users to Endpoint Connect, then read the following sections before reaching a decision. There are a number of important differences between SecureClient and Endpoint Connect. For example:

- SecureClient supports link selection and multiple entry points. The current release of Endpoint Connect accomplishes MEP through the use of a DNS server for configured for DNS-based Geo-cluster name resolution.
- Secure Configuration Verification (SCV) for Endpoint Connect is achieved through the use of Cooperative enforcement.

Link Selection and MEP

Link Selection

Because remote sites are defined on the client according to a single IP address or resolvable (DNS) name, Link Selection is not supported. Endpoint Connect ignores SecureClient link selection settings.

Multiple Entry Points

If a site implements gateways in a cluster (load sharing) or primary-backup/first to respond (high availability) configuration, it is important that the client performs DNS resolution each time it connects to the site. Providing:

- A site is defined by a DNS resolvable name rather than IP address
- The DNS server is configured for DNS-based Geo-Cluster name resolution

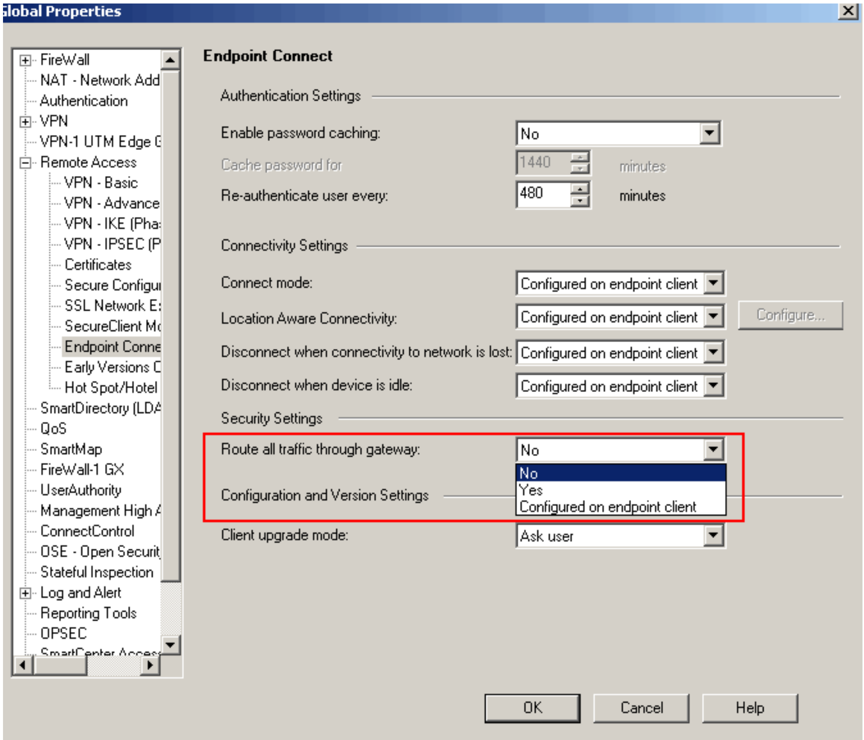
then Multiple Entry Point (MEP) for gateway redundancy can be implemented by the DNS server.

See "Disabling DNS-based Geo-Cluster Name Resolution" on page 377 for related information.

Hub Mode

Hub mode is configured in two places:

1. In **Global Properties > Remote Access > Endpoint Connect**:



Endpoint Connect

Authentication Settings

Enable password caching: No

Cache password for 1440 minutes

Re-authenticate user every: 480 minutes

Connectivity Settings

Connect mode: Configured on endpoint client

Location Aware Connectivity: Configured on endpoint client

Disconnect when connectivity to network is lost: Configured on endpoint client

Disconnect when device is idle: Configured on endpoint client

Security Settings

Route all traffic through gateway: No

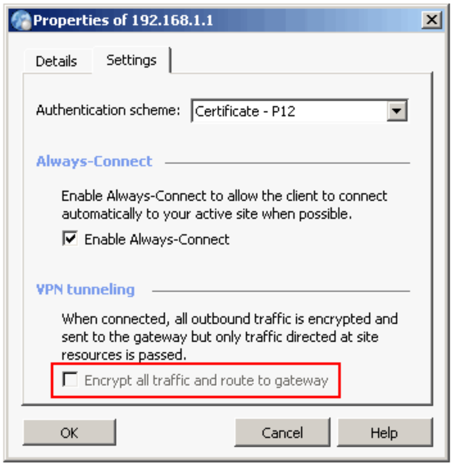
No
Yes
Configured on endpoint client

Configuration and Version Settings

Client upgrade mode: Ask user

The options are:

- Not to route all the traffic through the gateway
- To route all traffic through the gateway
- Route according to the Endpoint client configuration. If the client configuration is the determining factor, see the client's **Settings** tab for a particular site:



Properties of 192.168.1.1

Authentication scheme: Certificate - P12

Always-Connect

Enable Always-Connect to allow the client to connect automatically to your active site when possible.

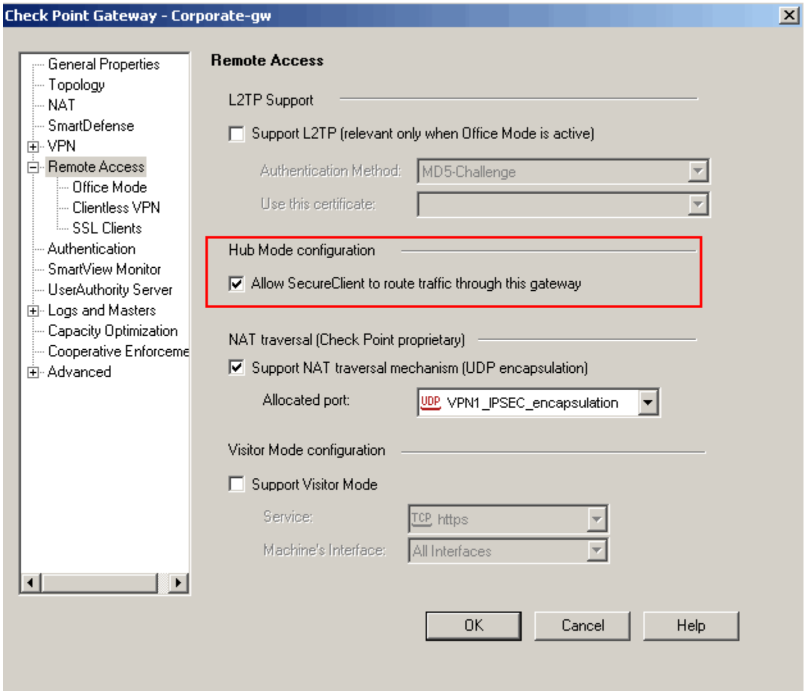
☑ Enable Always-Connect

VPN tunneling

When connected, all outbound traffic is encrypted and sent to the gateway but only traffic directed at site resources is passed.

□ Encrypt all traffic and route to gateway

2. On the **Gateway > Remote Access** page:



Check Point Gateway - Corporate-gw

Remote Access

L2TP Support

□ Support L2TP (relevant only when Office Mode is active)

Authentication Method: MD5-Challenge

Use this certificate:

Hub Mode configuration

☑ Allow SecureClient to route traffic through this gateway

NAT traversal (Check Point proprietary)

☑ Support NAT traversal mechanism (UDP encapsulation)

Allocated port: UDP VPN1_IPSEC_encapsulation

Visitor Mode configuration

□ Support Visitor Mode

Service: TCP https

Machine's Interface: All Interfaces

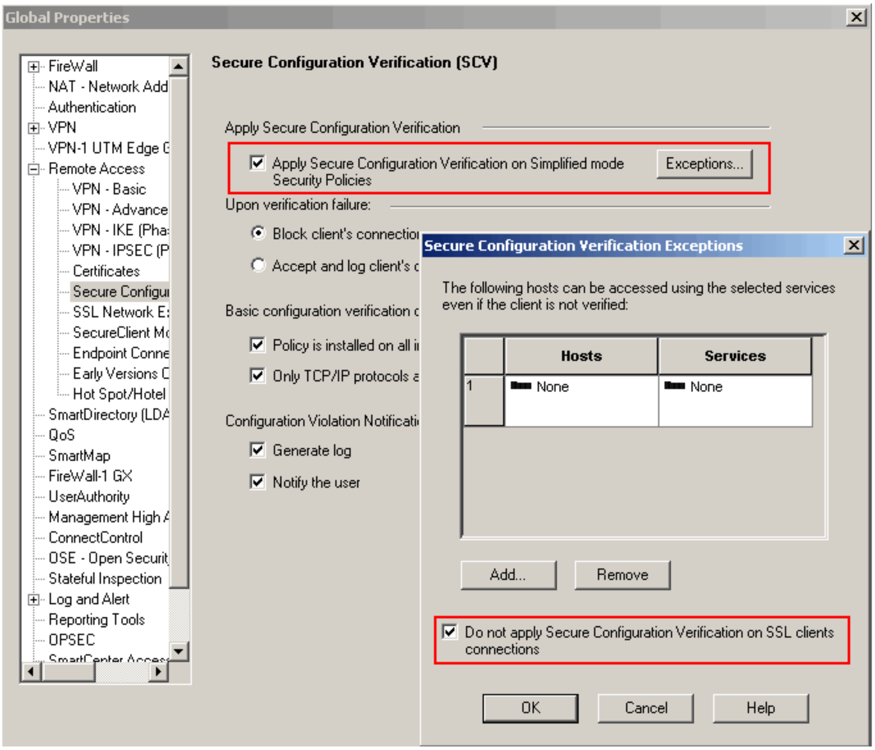
Select **Allow SecureClient to route traffic through the gateway.** The setting applies equally to Endpoint Connect.

Secure Configuration Verification

While SecureClient uses the SCV settings configured in **Global Properties**, no SCV enforcement occurs for Endpoint Connect. Endpoint Connect takes advantage of Cooperative enforcement to guarantee a secure configuration.

This means that for Endpoint Connect:

1. Bypass SVC for none SecureClient traffic by creating an exception:
   a. Open **Global Properties > Remote Access > Secure Configuration Verification**
   b. Select **Apply Secure Configuration Verification on Simplified mode Security Policies**
   c. Click **Exceptions**
   d. In the **Secure Configuration Verification Exceptions** window select **Do not apply Secure Configuration Verification on SSL client connections**:



Secure Configuration Verification (SCV)

Apply Secure Configuration Verification

☑ Apply Secure Configuration Verification on Simplified mode Security Policies

Upon verification failure:

● Block client's connectio

○ Accept and log client's c

Basic configuration verification c

☑ Policy is installed on all i

☑ Only TCP/IP protocols a

Configuration Violation Notificati

☑ Generate log

☑ Notify the user

Secure Configuration Verification Exceptions

The following hosts can be accessed using the selected services even if the client is not verified:

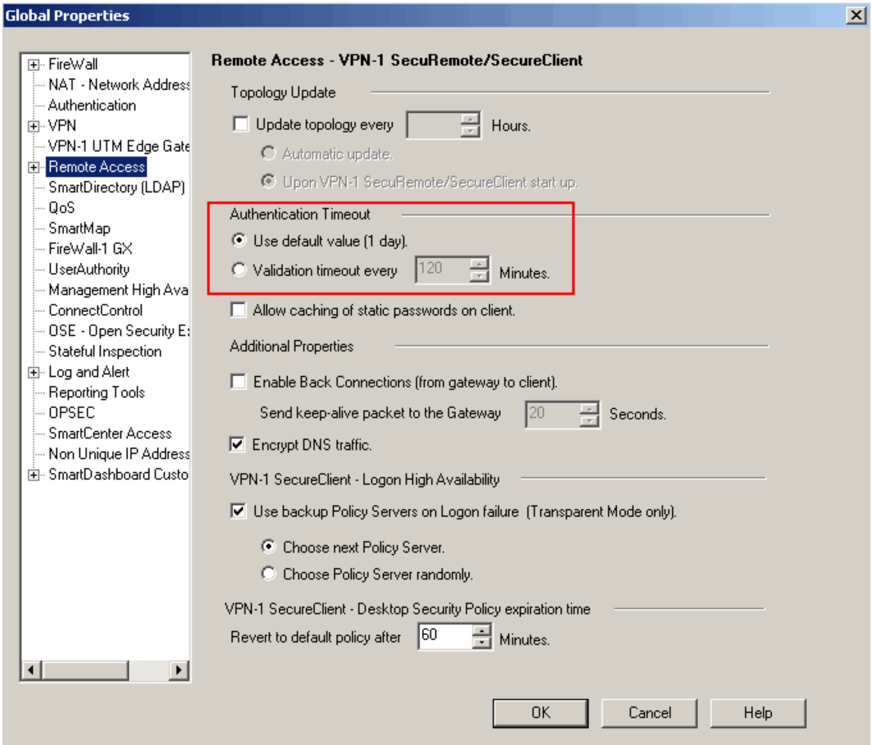
	Hosts	Services
1	None	None

☑ Do not apply Secure Configuration Verification on SSL clients connections

2. On the gateway object, turn on Cooperative Enforcement. See "Cooperative Enforcement" on page 382.

Authentication Time-out

These settings for the gateway determine how long the remote client's password remains valid, which is equal to the frequency of IKE phase 1. By default, IKE authentication is valid for one day. If you do not want to accept the IKE default, set a different value here in **Global Properties > Remote Access > Authentication Timeout**:



Remote Access - VPN-1 SecuRemote/SecureClient

Topology Update

□ Update topology every [ ] Hours.

○ Automatic update.

● Upon VPN-1 SecuRemote/SecureClient start up.

Authentication Timeout

● Use default value (1 day).

○ Validation timeout every [120] Minutes.

□ Allow caching of static passwords on client.

Additional Properties

□ Enable Back Connections (from gateway to client).

Send keep-alive packet to the Gateway [20] Seconds.

☑ Encrypt DNS traffic.

VPN-1 SecureClient - Logon High Availability

☑ Use backup Policy Servers on Logon failure (Transparent Mode only).

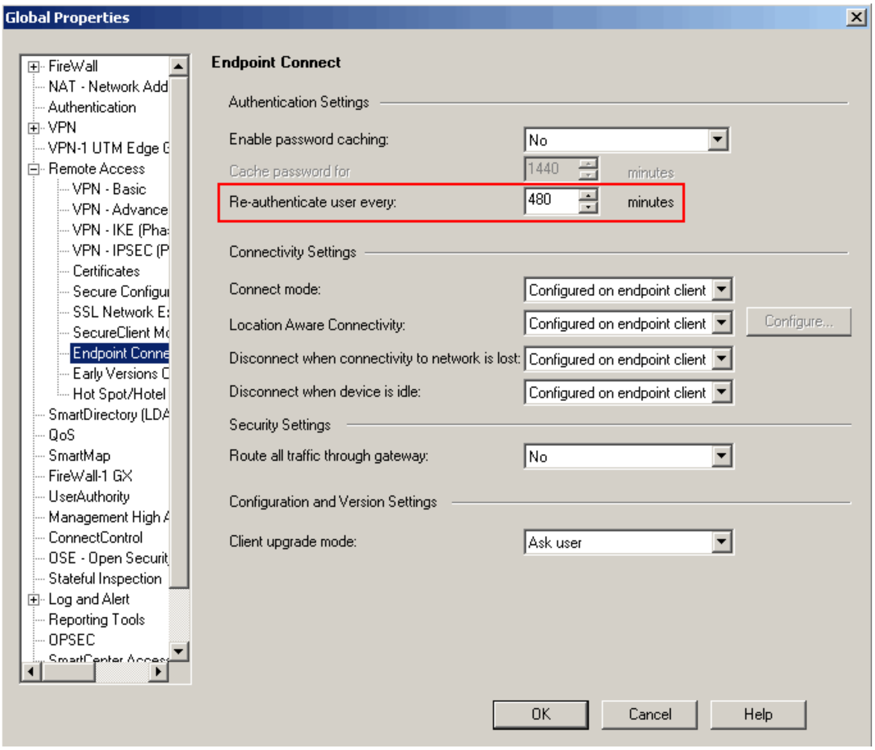
● Choose next Policy Server.

○ Choose Policy Server randomly.

VPN-1 SecureClient - Desktop Security Policy expiration time

Revert to default policy after [60] Minutes.

The client timeout is read from the Endpoint Connect Settings in **Global Properties > Remote Access**:



Global Properties

Endpoint Connect

Authentication Settings

Enable password caching: No

Cache password for 1440 minutes

Re-authenticate user every: 480 minutes

Connectivity Settings

Connect mode: Configured on endpoint client

Location Aware Connectivity: Configured on endpoint client

Disconnect when connectivity to network is lost: Configured on endpoint client

Disconnect when device is idle: Configured on endpoint client

Security Settings

Route all traffic through gateway: No

Configuration and Version Settings

Client upgrade mode: Ask user

Working with RSA Hard and Soft Tokens

If SecurID is used for authentication, you must manage the users on RSA's ACE management server. ACE manages the database of RSA users and their assigned hard or soft tokens. The client contacts the site's gateway. The gateway contacts the ACE Server for user authentication information. This means:

- Remote users must be defined as RSA users on the ACE Server.
- On the gateway, SecurID users must be placed in a group with an external user profile account that specifies SecurID as the authentication method.

For remote users to successfully use RSA's softID:

1. Create a remote users group on the Ace Server
2. Distribute the SDTID token file (or several tokens) to the remote users "out of band".
3. Instruct remote users on how to import the tokens.

Configuring Logging Options for Client Users

The **Options** window, **Advanced** tab of the client enables users to send log using their default mail client. Administrators can:

- Define an email address for these log files by modifying the **send_client_logs** attribute in `$FWDIR/conf/trac_client_1.ttm` on the gateway.

```
:send_client_logs (
                        :gateway (
                                :default ("email@example.com")
                        )
                )
```

- If an email address is not defined in `trac.client_1.ttm`, clicking **Collect Logs** in the **Options > Advanced** window collects all the client logs into a single CAB file, which the user can save and then send to the network administrator as an attachment.

Disabling CAPI Authentication

Endpoint Connect supports user authentication through the use of **PKCS#12** certificates. A **PKCS#12** certificate can be accessed directly or imported to the CAPI store and accessed from there.

If, for security reasons, you do not wish users to authenticate using certificates within the CAPI store:

1. On the gateway, open the `$FWDIR/conf/trac_client_1.ttm` file for editing.
2. Modify the **enable_capi** attribute to FALSE.

```
enable_capi (
   :gateway (
    :map (
     :false (false)
     :true (true)
     :client_decide (client_decide)
    )
    :default (true)
   )
  )
```

By default, the value is TRUE.

Modify the `:default (true)` line.

Disabling DNS-based Geo-Cluster Name Resolution

Each time Endpoint Connect connects to the security or Connectra gateway, the client performs, by default, DNS name resolution. For a deployment consisting of a single gateway, there is no need to perform DNS resolution every time — the first time the client connects, it caches the IP address of the gateway and reuses it on each subsequent connect operation. In this kind of deployment, the default behavior of the client can be safely modified.

To prevent the gateway from performing name resolution each time Connect connects:

1. On the gateway, open the `$FWDIR/conf/trac_client_1.ttm` file for editing.
2. Modify the **enable_gw_resolving** attribute to FALSE:

```
:enable_gw_resolving (
   :gateway (
    :map (
     :false (false)
     :true (true)
     :client_decide (client_decide)
    )
    :default (true)
   )
  )
```

   By default, the value is TRUE.

   Modify the `:default (true)` line.

However, in a deployment consisting of multiple gateways, for example in a cluster (load sharing) or primary-backup (high availability) configuration, it is important that the client performs DNS resolution each time it connects to the site. Based on geographical proximity or the load-sharing requirements of the gateway, the DNS server might return to the client a different IP address each time: the IP address of the *nearest available gateway*. This IP address may not be the same as the IP address cached during the first connect operation. Resolving DNS names each time:

- Enables DNS to be used for High availability (the IP address of the backup gateway is returned when the primary fails to respond)
- Adds to the client a functionality similar to MEP (Multiple Entry Points)

Note - This is not a regular cluster environment, as the two or more gateways are not synchronized.

Configuring Endpoint Compliance Checks

While Endpoint Compliance scanner provides an effective endpoint compliance check when connecting, it might prove resource intensive to the gateway. For improved speed, it is recommended to scan only for the presence of antivirus and firewall software.

NAT Traversal

When a remote user initiates a VPN (IPSec encrypted) session with the gateway, during the initial negotiation, both gateway and remote client attempt to detect whether the traffic between them passed through a NAT device.

For a number of reasons NAT is incompatible with IPSec:

- IPSec assures the authenticity of the sender and the integrity of the data by checking to see that the data payload has not been changed in transit. A NAT device alters the IP address of the remote client. The Internet Key Exchange (IKE) protocol used by IPSec embeds the client's IP address in its payload, and this embedded address, when it reaches the gateway, will fail to match the source address of the packet, which is now that of the NAT device. When addresses don't match, the gateway drops the packet.
- TCP and UDP checksums in the TCP header are sometimes used to verify the packet's integrity. The checksum contains the IP addresses of the remote client and gateway, and the port numbers used for the communication. IPSec encrypts the headers with the Encapsulating Security Payload (ESP) protocol. Since the header is encrypted, the NAT device cannot alter it. This results in an invalid checksum. The Connectra gateway again rejects the packet.

The Endpoint Connect Client resolves these and other NAT related issues by using NAT-Traversal (NAT-T) as a way of passing IPSec packets through the NAT device.

On the Connectra gateway, default ports are:

- Internet Key Exchange (IKE) - User Datagram Protocol (UDP) on port 500

  Note - only IKEv1 is supported
- IPsec NAT-T - UDP on port 4500
- Encapsulating Security Payload (ESP) - Internet Protocol (IP) on 50

If a NAT device is detected during the initial negotiation, communication between gateway and client switches to UDP port 4500. Port 4500 is used for the entire VPN session.

Note - NAT-T packets (or the packets of any other protocol) need to return to the client through the same interface they came in on. While the recommended deployment is to place the Connectra gateway in a public DMZ with a single interface for all traffic, it is also possible to deploy Connectra with inbound and outbound interfaces, the default route being the outbound route towards the Internet. Endpoint Connect *only* connects to the Connectra gateway's *default outbound interface*.

Smart Card Removal Detection

If remote users authenticate through a Smart Card, and the smart card or smart reader is removed from the USB port, the client detects that the certificate is no longer available and disconnects from the site. A **VPN tunnel has disconnected. Smart card was removed** message is displayed to the user. To configure this behavior:

On the gateway:

1. Open `$FWDIR/conf/trac_client_1.ttm` for editing.
2. Locate the line:

```
:disconnect_on_smartcard_removal (
     :gateway (
       :default (client_decide)
     )
   )
```

3. Replace "Client_decide" with either TRUE or FALSE. If leave the decision to the client, then on the client:

   a. Open `%programfiles%\checkpoint\endpoint connect\trac.defaults` for editing.

   b. Locate the line:

```
:disconnect_on_smartcard_removal  STRING   false GW_USER 0
```

   c. Replace `false` with `true`

   d. Save and close the file

Tunnel Idleness

A number of organizations may have specific security requirements, such that an open VPN tunnel should be transporting work-related traffic to the site at all times. An idle or inactive tunnel should be shut down. (Stay-alive packets or "noise" such as NetBios Broadcasts to port 83 or DNS broadcasts to port 137 are not considered "work related". A mail program such as OUTLOOK performing a send-receive operation every five minutes would be considered work-related, and the tunnel kept open.)

For this reason, a tunnel idleness interval can be configured in accordance with the company's security policy.

Tunnel idleness can be set either in the gateway policy or on the client.

Configuring Tunnel Idleness on the Client

1. Open for editing:

   `%programfiles%\checkpoint\endpoint connect\trac.defaults`.

2. Locate the line:

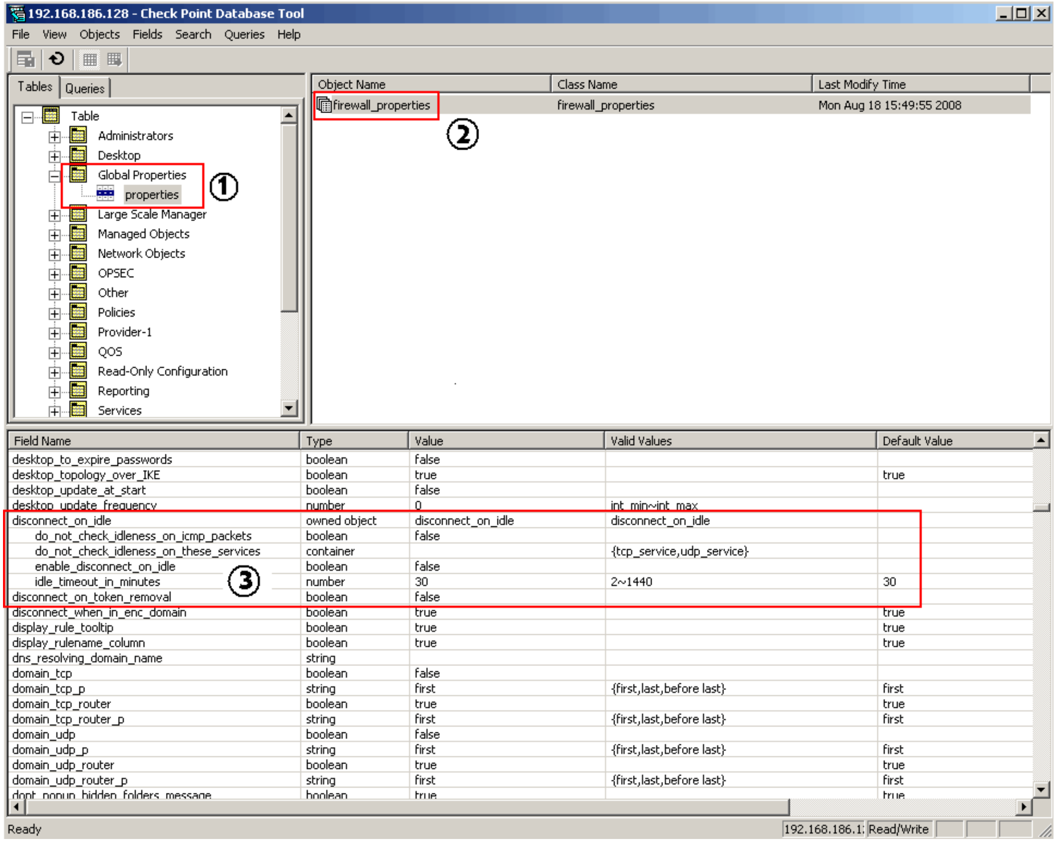
   ```
   tunnel_idleness_timeoutINT 0 GW_USER 0
   ```

3. Replace `INT 0` with a value. For example `INT 20` will shut down an inactive VPN tunnel after twenty minutes. `INT 0` means no tunnel idleness. The feature is switched off.

   This setting is only enforced if no policy regarding tunnel idleness is set on the gateway. Otherwise, the gateway policy set by Security Management server is enforced.

Configuring Tunnel Idleness on the Server:

1. Connect to the Security Management server using GuiDBedit.
2. Open the **Global Properties > properties > firewall_properties object**:



Object Name	Class Name	Last Modify Time
firewall_properties	firewall_properties	Mon Aug 18 15:49:55 2008

Field Name	Type	Value	Valid Values	Default Value
desktop_to_expire_passwords	boolean	false		
desktop_topology_over_IKE	boolean	true		true
desktop_update_at_start	boolean	false		
desktop_update_frequency	number	0	int_min~int_max	
disconnect_on_idle	owned object	disconnect_on_idle	disconnect_on_idle	
do_not_check_idleness_on_icmp_packets	boolean	false		
do_not_check_idleness_on_these_services	container		{tcp_service,udp_service}	
enable_disconnect_on_idle	boolean	false		
idle_timeout_in_minutes	number	30	2~1440	30
disconnect_on_token_removal	boolean	false		
disconnect_when_in_enc_domain	boolean	true		true
display_rule_tooltip	boolean	true		true
display_rulename_column	boolean	true		true
dns_resolving_domain_name	string			
domain_tcp	boolean	false		
domain_tcp_p	string	first	{first,last,before last}	first
domain_tcp_router	boolean	true		true
domain_tcp_router_p	string	first	{first,last,before last}	first
domain_udp	boolean	false		
domain_udp_p	string	first	{first,last,before last}	first
domain_udp_router	boolean	true		true
domain_udp_router_p	string	first	{first,last,before last}	first
dont_popup_hidden_folders_message	boolean	true		true

3. Configure the following parameters:

   - `disconnect_on_idle`
   - `do_not_check_idleness_on_icmp_packets`
   - `do_not_check_idleness_on_these_services`

     By listing services, you effectively include the associated port numbers. For example if NetBios is entered as the service, then port 83 is not monitored for tunnel idleness.

   - `enable_disconnect_on_idle`
   - `idle_timeout_in_minutes`

Set a value such as 30, for thirty minutes.

Note - If you enter a value for `idle_timeout_in_minutes`, then `enable_disconnect_on_idle` must also be set to TRUE.

4. Save and install a policy.

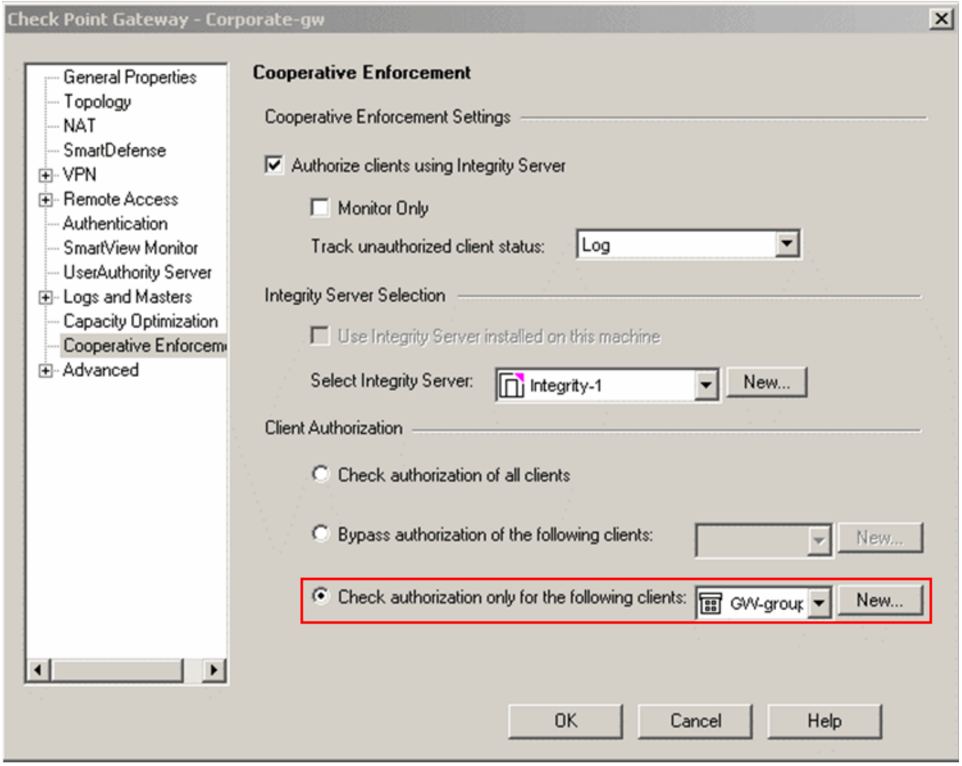
Cooperative Enforcement

Endpoint Connect works with Cooperative Enforcement technology. Cooperative enforcement relies on the Endpoint Security server compliance feature, which defines whether a remote client is secure, and blocks connections that do not meet predefined requisites. Using this feature, each firewall module queries an Endpoint Security server to determine the policy compliance of any client that attempts to open a connection through it. By enabling cooperative enforcement an administrator can allow traffic only from clients that comply with the firewall module's software configuration. This configuration is defined in the Endpoint software policy.

SmartDashboard Configuration

When configuring cooperative enforcement for Endpoint Connect clients:

- In SmartDashboard, on the gateway properties window, create a group that contains the Endpoint Connect users.



Cooperative Enforcement

Cooperative Enforcement Settings

☑ Authorize clients using Integrity Server

□ Monitor Only

Track unauthorized client status: Log

Integrity Server Selection

□ Use Integrity Server installed on this machine

Select Integrity Server: Integrity-1 New...

Client Authorization

○ Check authorization of all clients

○ Bypass authorization of the following clients: New...

● Check authorization only for the following clients: GW-grour New...

This group, when trying to open a connection through the firewall module, will be checked for Endpoint Security compliance.

Manual Configuration

The following features cannot be configured in SmartDashboard. These features can be modified by manually editing the database with tools such as `DBEdit/GUIDBEdit`.

- **Track authorized hosts**

  Supplies a log entry for any host event that is compliant and has access through the firewall. To create this log entry use the following command:

  ```
  eps_track_authorized_host
  ```

- **Connect to the host when there is no firewall / Endpoint Security server connection**

  Drops any connection from the hosts when the Endpoing Security server defined specifically for the firewall is unreachable.

  This feature is turned off by default. To turn it on use the following command:

```
eps_connect_host_integrity_down
```

- **Connect a host while waiting for status reply from the Endpoint Security server ('pending' state)**

  This feature instructs the firewall module to allow the connection from any host during the compliance validation period. This option is useful when the Endpoint Security server is loaded with compliance requests and the timeout for such replies increases.

  In this case, any connecting host will be able to open an outside connection. That is, the firewall will allow traffic even when an answer from the Endpoint Security server has not been received. The connection will be dropped only if this host is not authorized.

  To enable this feature use the following command:

  ```
  eps_connect_host_pending
  ```

- **Track authorized hosts**

  Supplies a log entry regarding a successful host authorization by the Endpoint Security server (with Endpoint Security Client installed). Any further traffic from this host will not be logged by this log type. To prevent excessive logs from being written to the Security Management server, this log type is turned off be default. This type of log can be activated with the following command:

  ```
  eps_track_authorized_host
  ```

- **Enforce external interface**

  By default, cooperative enforcement only checks clients that arrive from the internal gateway's interface. This parameter enables checking of all remote clients that open VPN connections to the gateway’s external interface. The option is turned off by default.

  To enable this feature use the following command:

  ```
  eps_enforce_external_if
  ```

Note - This option must be set for Endpoint Clients to successfully connect with the gateway.

- **Track dropped connections**

  By default, any connection that is dropped as a result of a SAM rule is not tracked.

When activated, the SmartView Tracker will issue a log of any dropped connection from an unauthorized client (that is, according to EPS definitions).

To track any connection that is dropped as a result of a SAM rule use the following command:

```
eps_track_dropped_conn
```

For further information, see the Endpoint Security documentation available on your R70 installation CD, or visit the Check Point support center:

https://supportcenter.checkpoint.com

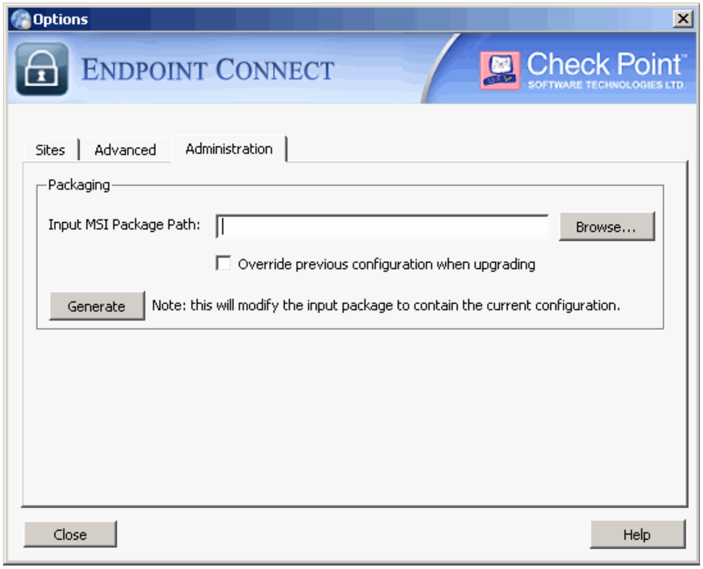
Using the Packaging Tool

Endpoint Connect supports a special administration mode that enables the creation of preconfigured packages. The administrator opens one instance of the client, configures all settings then saves the client as an **.msi** package for further distribution to end point users.

To create a preconfigured package:

1. Open the Endpoint Connect client in administration mode:
   - Click on AdminMode.bat file in `c:\Program Files\Checkpoint\Endpoint Connect`, or:
   - From a command prompt, run: `c:\Program Files\Checkpoint\Endpoint Security\trgui.exe /admin`
2. Right-click the client icon in the system tray, and select **VPN Options**.

   The VPN Options window opens showing the administration tab:


Options
Endpoint Connect
Check Point
SOFTWARE TECHNOLOGIES LTD.
Sites
Advanced
Administration
Packaging
Input MSI Package Path:
Browse...
☐ Override previous configuration when upgrading
Generate
Note: this will modify the input package to contain the current configuration.
Close
Help

3. Using the options on the **Site** and **Advanced** tabs, configure:
   - Site definitions
   - Authentication method
   - Logging
   - Proxy server settings
   - Always-connect mode
   - VPN tunneling
4. On the **Administration** tab:
   a. Select a folder for the new package
   b. Decide whether to override the previous configuration when upgrading
   c. Click **Generate** to create the **.msi** package in the designated folder.
5. Distribute this package to Endpoint Connect users.

   Users can download the package directly from the gateway by entering the following URL into a browser:

   https://<gateway ip>/CSHELL/CheckPointEndpointConnect.msi.

Chapter 18

Endpoint Connect API

In This Chapter:

This section covers the OPSEC API for embedded custom client integrations. The API contains functions exported by the **TrAPI.dll** library, an API infrastructure employed to transfer messages between the client and the **tracsrvwrapper** service. The API exposes functions that form synchronic actions, for example retrieving the status for a specific connection. The API also contains functions that enable the client to register to receive various notifications from the service. Because the notifications can arrive at any time, these functions are considered asynchronic. API calls to the client block the client until the function completes. When the API calls any API function, the API infrastructure sends the corresponding message to the service and waits for the service's response.

Function prototypes are defined in the **TrAPITypes.h** header file. To use the client API, first download the client zip file from the Check Point Support Center. The zip file contains the library file **TrAPI.dll**, and the header files **TrAPITypes.h** and **TrAPI.h**.

Introduction to the Client OPSEC API

The client API is C based. Exported functions must have a C-style declaration. To access these API functions from C++, use the extern “C” declaration. The API supports:

Two General Functions for Error Tracing

- TrInitNewExceptionFilter
- TrCloseExceptionFilter

Use these functions to print the stack when a process terminates unexpectedly.

Functions to transfer messages to the service:

- TrAPIInit
- TrAPIInitDebug
- TrAPIDebug
- TrStart
- TrStop
- TrIsTracActive
- TrConnEnum
- TrConnGetInfo
- TrConnConnect
- TrConnCancelConnect
- TrConnCreate
- TrConnDelete
- TrGetInformation
- TrGetConfiguration
- TrSetConfiguration
- TrSendLogs
- TrGetStatus
- TrAPIGetVersion
- TrSendNotification

- TrRegisterErrorCallback

Functions to Receive Notifications from the Service

Use the following functions to receive notifications from the trac service. All notifications are described in **TrAPIType.h**. The client can register with the service to receive only specific notifications. By default, the client receives all notifications.

- TrRegisterNotificationCallback
- TrUnregisterNotificationCallback

Function Return Codes

The return codes and numerical equivalents for API Functions are as follows:

Function Return Code	Equivalent	Meaning...
TrOK	0	Function executed without error.
TrFAIL	-1000	Function failed.
TrConnAlreadyConnected	-999	Connect function failed because the client is already connected.
TrConnNameAlreadyExisted	-998	Connect function failed because a site with this name already exists.
TrConnAddrAlreadyExisted	-997	Connect function failed because a site with this IP address already exists.
TrParamsFAIL	-996	Function failed because a wrong parameter was passed to it.
TrAllocFAIL	-995	Function failed because of a memory shortage.
TrComSendFAIL	-994	Function failed to establish communication with the service
TrAPIInitFAIL	-993	Function failed to communicate with the service

Function Return Code	Equivalent	Meaning...
TrICSNoCompliance	-992	Function failed because the user failed the end point compliance test.
TrProxyAuthFailed	-991	Function failed because proxy authentication failed
TrProxyAuthRequired	-990	Function failed because proxy authentication credentials were not presented.

Client Functions Communicating with Service

In This Section:

TrAPIInit

The first function called after loading **TrAPI.dll**. Only run once and before calling any other function. If the service goes down, the function needs to be initialized again.

Prototype

```
TRAPI_CPAPI TrStatus TrAPIInit();
```

TrAPIInitDebug

This function creates logs.

Prototype

```
TRAPI_CPAPI TrStatus TrAPIInitDebug(TrString filename,int max_size, int
max_files,int TopicLevel);
```

Arguments

Argument	IN/OUT	Meaning...
`filename`	in	The name of the log file.
`max_size`	in	Maximum size of log in Bytes.
`max_files`	in	Maximum number of files.
`TopicLevel`	in	The number of topics the logs should contain.

TrAPIDebug

This function writes a text message to the log file.

Prototype

```
TRAPI_CPAPI void TrAPIDebug(const char *TopicNames,int TopicLevel,int
err, const char *fmt,...);
```

Arguments

Argument	IN/OUT	Meaning...
`TopicNames`	in	the names of topics used in the logs.
`TopicLevel`	in	the number of topics.
`err`	in	Error level number, for example fatal error=1, informative error message=5.
`fmt`	in	The text message to be inserted in the log file.

TrStart

Starts the service.

Prototype

```
TRAPI_CPAPI TrStatus TrStart();
```

TrStop

Stops the service.

Prototype

```
TRAPI_CPAPI TrStatus TrStop();
```

TrIsTracActive

Checks whether the trac service is active.

Prototype

```
TRAPI_CPAPI bool TrIsTracActive();
```

TrConnEnum

Ennumerates all configured sites. Returns a connection handle according to the given index, starting from zero. When there are no more sites in the list, a NULL value is returned.

Prototype

```
TRAPI_CPAPI TrConn TrConnEnum(int connIndex);
```

Arguments

Argument	IN/OUT	Meaning...
`connIndex`	in	the connection handle representing the connection for the site

TrConnGetInfo

According to a given connection handle, this function retrieves information from the connection STRUCT.

Prototype

```
TRAPI_CPAPI TrStatus TrConnGetInfo(TrConn connHandle, TrConnStruct*
connStruct);
```

Arguments

Argument	IN/OUT	Meaning...
connHandle	in	The handle for the connection
TrConnStruct	out	The connection information as contained in the STRUCT: • char mDisplayName[PARAM_MAX_LEN]; the same of the site, as given by the user. • char mGwIP[PARAM_MAX_LEN]; the IP address of the site's gateway. • char mGwHostname[PARAM_MAX_LEN]; the FQDN of the site. • int mConnStatus; the status of the connection: connecting, connected, reconnecting, or terminated (when the service is down). Idle=0. • bool mIsActiveSite; TRUE if this connection is the active site, meaning the last site to which the user successfully connected. • TrAuthInformation mAuthInfo; the authentication scheme for the given site. • TrConn mConnHandle; the connection handle.

TrConnConnect

Connects to the site according to the given connection handle. Also checks to see whether the user cancels the action at any point.

Prototype

```
TRAPI_CPAPI TrStatus TrConnConnect(IN TrConnStruct * connStruct);
```

Arguments

Argument	IN/OUT	Meaning...
connStruct	in	Specifically, only the connhandle and authentication information inside the STRUCT are required.

TrConnCancelConnect

Cancels the connection to the given site.

Prototype

```
TRAPI_CPAPI TrStatus       TrConnCancelConnect(TrConn connHandle);
```

Arguments

Argument	IN/OUT	Meaning...
connHandle	in	Handle of the site to cancel.

TrConnCreate

Creates a new site according to the data given in connStruct, and returns a connection handle.

Prototype

```
TRAPI_CPAPI TrStatus TrConnCreate(IN TrConnStruct * connStruct);
```

Arguments

Argument	IN/OUT	Meaning...
connStruct	in	the STRUCT that contains the display name, IP address of the site's gateway, and the FQDN.

TrConnDelete

Deletes a site according to the given connection handle.

Prototype

```
TRAPI_CPAPI TrStatus TrConnDelete(TrConn connHandle);
```

Arguments

Argument	IN/OUT	Meaning...
`connHandle`	in	the handle of the site to be deleted.

TrGetInformation

This function returns a list of all Domain Names. The service obtains the list of DNs from certificates in the certificate store.

Prototype

```
TRAPI_CPAPI TrStatus TrGetInformation(TrParam paramType, TrMsg**
pParamValue);
```

Arguments

Argument	IN/OUT	Meaning...
`paramType`	in	Two types are available: • TR_USE_DN_LIST. Returns list of DNs. • TR_ICS_REPORT_FILENAME. Returns location of the compliance check report.
`pParamValue`	out	the returned message.

TrGetConfiguration

Retrieves information related to site variables. The function expects an argument list. The first argument must be the IP address of the gateway if referring to a specific gateway, otherwise an empty string. Each argument must be a string that holds the name of the requested configuration variable.

Prototype

```
TRAPI_CPAPI TrStatus    TrGetConfiguration(TrMsg* pParams, TrMsg **
pConfiguration);
```

Arguments

Argument	IN/OUT	Meaning...
`pParams`	in	Message that contains attributes to retrieve, such as default time out.
`pConfiguration`	out	Returns requested attribute

TrSetConfiguration

This function saves the user's configuration as an attribute / value pair, for example:

Attribute	Value
IP Address	192.168.x.x
Authentication scheme	

The function expects an argument list. The first argument must be the IP address of the gateway if referring to a specific gateway, otherwise an empty string. Each argument must be a string that holds the name of the attribute.

Prototype

```
TRAPI_CPAPI TrStatus TrSetConfiguration(TrMsg* pConfiguration);
```

Arguments

Argument	IN/OUT	Meaning...
`pConfiguration`	in	configuration to be stored.

TrAPIGetVersion

Returns the client version.

Prototype

```
TRAPI_CPAPI TrStatus TrAPIGetVersion(TrVersion* version);
```

Arguments

Argument	IN/OUT	Meaning...
`version`	out	Returns major version, minor version, and build number

TrSendNotification

Sends a notification from the client to the service. All notifications are described in **TrAPIType.h**. The client can register with the service to receive only specific notifications. By default, the client receives all notifications.

Prototype

```
TRAPI_CPAPI TrStatus TrSendNotification(TrNotification *
pClientNotification);
```

Arguments

Argument	IN/OUT	Meaning...
`pClientNotification`	in	Notifications to send to the service.

TrRegisterErrorCallback

When communication with the service is lost, the client registers a callback to be called by **TrAPI.dll**.

Prototype

```
TRAPI_CPAPI void TrRegisterErrorCallback(ErrorCbFunctor cb, void*
clientOpaque);
```

Arguments

Argument	IN/OUT	Meaning...
`ErrorCbFunctor cb`	in	the registered callback
`clientOpaque`	in	client's opaque to the callback

Notification Identifiers

TrNotificationID

Identifiers for each notification.

Prototype

```
enum TrNotificationID
```

NotificationID	Meaning and Format...
`TR_NOTIFICATION_NETWORK_OUT`	The client is located outside of the VPN domain
`TR_NOTIFICATION_NETWORK_IN`	The client is located within the VPN domain
`TR_NOTIFICATION_NETWORK_NO_NETWORK`	No network available
`TR_NOTIFICATION_CONNECTION_DISCONNECTED`	Connection disconnected Disconnect reason: type - eTrArgTypeStr val - a string representing the disconnect reason default_text - NULL
`TR_NOTIFICATION_CONNECTION_RECONNECTING`	Reconnecting Reconnecting reason: type - eTrArgTypeStr val - a string representing the reconnecting reason default_text - NULL

NotificationID	Meaning and Format...
TR_NOTIFICATION_TRAC_STOP	Service is stopped
TR_NOTIFICATION_LOG	Logs message Log string: type - eTrArgTypeStr val - the log's string default_text - NULL
TR_NOTIFICATION_UPGRADE	Client upgrade is required. upgrade string type - eTrArgTypeStr val - the upgrade's string default_text - NULL
TR_NOTIFICATION_CLIENT_UPGRADE	Upgrade notification sent by the client to the service. Perform upgrade type - eTrArgTypeInt32 val - an integer represents whether the user wishes to upgrade: 1 for upgrade, 0 for no_upgrade. default_text - NULL
TR_NOTIFICATION_ICS_NO_COMPLIANCE	End point failed the endpoint compliance test

NotificationID	Meaning and Format...
TR_NOTIFICATION_AUTH_SUPPLY_CREDS	Supply authentication credentials. The number of arguments depends on the authentication scheme: 1. GW: type - eTrArgTypeStr val - a string representing the gateway's name default_text - NULL 2. Authentication type (TrAuthType): type - eTrArgTypeInt32 val - an integer represents the authentication type default_text - NULL 3. Number of parameters: type - eTrArgTypeInt32 val - an integer represents the number of parameters (e.g. 2 for username+password, 1 for certificate dn, 3 for username+pin+passcode) default_text - NULL #) Param number # type - eTrArgTypeStr val - a string representing the parameter (e.g. "username" or "passcode", etc) default_text - NULL

NotificationID	Meaning and Format...
TR_NOTIFICATION_CLIENT_CREDENTIALS	Authentication credentials sent from the client to the service. The number of arguments depends on the authentication scheme. 1. Gateway type - eTrArgTypeStr val - a string representing the gateway's ip address default_text - NULL 2. Authentication type (TrAuthType): type - eTrArgTypeInt32 val - an integer represents the authentication type default_text - NULL 3. Number of values: type - eTrArgTypeInt32 val - an integer represents the number of values (e.g. 2 for username+password, 1 for certificate dn, 3 for username+pin+passcode) default_text - NULL #) Value number # type - eTrArgTypeStr val - a string representing the value (e.g. the username value, the pin code value, etc) default_text - NULL

NotificationID	Meaning and Format...
TR_NOTIFICATION_CONNECTION_PROGRESS	Progress of the connection operation. Takes six arguments: 1. Flow's type: type - eTrArgTypeInt32 val - an integer indicating the flow type: PRIMARY_CONN_FLOW = 0 RECONNECT_FLOW = 1 DISCONNECT_FLOW = 2 DOWNLOAD_CL_SETTINGS_FLOW = 3 default_text - NULL 2. Step's status: val - an integer indicating the TrStatus of the step default_text - NULL 3. Step's name: type - eTrArgTypeStr val - a string representing the step's name default_text - NULL 4. Step's reason for error: type - eTrArgTypeStr val -a string representing the reason for the step's failure. This value is only relevant when the step fails. If the step's status is "success", this value equals to the empty string. default_text - NULL

NotificationID	Meaning and Format...
	5. Total progress: type - eTrArgTypeInt32 val - an integer indicating the connect progress in percentages default_text - NUL 6. Next step’s name: type - eTrArgTypeStr val - a string representing the next step's name (empty string if this is the last step) default_text - NULL

Functions from Service to Client

In This Section:

The various types of notifications are described in **TrAPITypes.h**.

TrRegisterNotificationCallback

This function registers with the service notifications to be sent to the client.

Prototype

```
TRAPI_CPAPI TrStatus TrRegisterNotificationCallback(NotificationCbFunctor
cb,void* clientOpaque, int eNotificationType = TR_NOTIFICATION_ALL);
```

Arguments

Argument	IN/OUT	Meaning...
`NotificationCbFunctor cb`	in	the registered callback
`clientOpaque`	in	client's opaque.
`eNotificationType`	in	the notification type: • TR_NOTIFICATION_NETWORK_TYPE = (1<<16) • TR_NOTIFICATION_CONNECTION_TYPE = (1<<17) • TR_NOTIFICATION_SUGGEST_CONNECT_TYPE = (1<<18) • TR_NOTIFICATION_TRAC_STOP_TYPE = (1<<19) • TR_NOTIFICATION_LOG_TYPE = (1<<20) • TR_NOTIFICATION_AUTH_TYPE = (1<<21) • TR_NOTIFICATION_DOWNLOAD_TYPE = (1<<22) • TR_NOTIFICATION_CLIENT_TYPE= (1<<23) • TR_NOTIFICATION_ICS_TYPE = (1<<24) • TR_NOTIFICATION_ALL = 32767 << 16 For example, to receive only notifications of type network, connection and stop notifications then **eNotificationType** should be equal to: TR_NOTIFICATION_NETWORK_TYPE | TR_NOTIFICATION_CONNECTION_TYPE | TR_NOTIFICATION_TRAC_STOP_TYPE

TrUnregisterNotificationCallback

Unregisters the notification callback.

Prototype

```
TRAPI_CPAPI TrStatus TrUnregisterNotificationCallback();
```

TrMsgCreate

This function creates an array of parameters included in the message.

Prototype

```
TRAPI_CPAPI TrMsg*  TrMsgCreate(int version, char *ID, char *def_msg,
unsigned int arguments_num,...);
```

Arguments

Argument	IN/OUT	Meaning...
`version`	in	version number of the message
`ID`	in	ID of the message
`def_msg`	in	The message text
`arguments_num,...`	in	Number of parameters

Currently, these arguments should be zero or empty strings.

TrMsgConstruct

This function creates a message without arguments.

Prototype

```
TRAPI_CPAPI TrMsg *TrMsgConstruct(int version, char *ID, char *def_msg,
unsigned int arguments_num);
```

Arguments

Argument	IN/OUT	Meaning...
`version`	in	Version number of the message
`ID`	in	ID of the message
`def_msg`	in	The message text
`arguments_num`	in	Number of parameters

Currently, these arguments should be zero or empty strings.

TrMsgDestroy

This function destroys a given message.

Prototype

```
TRAPI_CPAPI void  TrMsgDestroy(TrMsg *message);
```

Arguments

Argument	IN/OUT	Meaning...
`message`	in	the message to destroy

TrMsgGetVersion

This function gets the version of a given message.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgGetVersion(TrMsg *message, int *version);
```

Arguments

Argument	IN/OUT	Meaning...
`message`	in	the message to destroy
`version`	out	version number of the message

Currently, these arguments should be zero or empty strings.

TrMsgGetID

This function gets the ID of the message.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgGetID(TrMsg *message, char **ID);
```

Arguments

Argument	IN/OUT	Meaning...
message	in	the message
ID	out	ID of the message

Currently, these arguments should be empty strings.

TrMsgGetDefaultMsg

This function fills the given message, and returns the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgGetDefaultMsg(TrMsg *message, char **def_msg);
```

Arguments

Argument	IN/OUT	Meaning...
message	in	Given message
def_msg	out	Return message

Currently, these arguments should be empty strings.

TrMsgArgIterCreate

This function creates an iterator for a given message, returns NULL for failure.

Prototype

```
TRAPI_CPAPI TrMsgArgIter *TrMsgArgIterCreate(TrMsg *message);
```

Arguments

Argument	IN/OUT	Meaning...
`message`	in	Given message

TrMsgArgIterDestroy

This function destroys an iterator.

Prototype

```
TRAPI_CPAPI void TrMsgArgIterDestroy(TrMsgArgIter *iter);
```

Arguments

Argument	IN/OUT	Meaning...
`iter`	in	the iterator

TrMsgArgIterGetArgNum

This function fills the argument number, and returns the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgArgIterGetArgNum(TrMsgArgIter *iter, int
*arg_num);
```

Arguments

Argument	IN/OUT	Meaning...
`iter`	in	the iterator to get
`arg_num`	out	number of arguments

TrMsgArgIterGetNextArg

This functions fills the next TrArg in theTrMsg. If there are no more TrArgs, fills arg with NULL, and the return code is TrOK. If the function fails, an appropriate TrStatus error code is returned, and arg is NULL.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgArgIterGetNextArg(TrMsgArgIter *iter, TrArg
**arg);
```

Arguments

Argument	IN/OUT	Meaning...
`iter`	in	the iterator
`arg`	out	the next argument

TrMsgSetIntArg

This function sets the argument in the given position to int argument, and overrides the current argument that exists in the given position.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgSetIntArg(TrMsg *message,int pos, int val, char
* default_txt);
```

Arguments

Argument	IN/OUT	Meaning...
`message`	in	the message
`pos`	in	position of the message
val	in	value of the message
default_text	in	the message text

TrMsgSetStrArg

This function sets the argument in the given position to a str argument, and overrides the current argument that exists in the given position.

Prototype

```
TRAPI_CPAPI TrStatus TrMsgSetStrArg(TrMsg *message, int pos, char * val,
char * default_txt);
```

Arguments

Argument	IN/OUT	Meaning...
`message`	in	the message
`pos`	in	the position of the message
val	in	the value of the message
default_text	in	the message text

TrNotificationConstruct

This function creates a new TrNotification, and return NULL upon error.

Prototype

```
TRAPI_CPAPI TrNotification*  TrNotificationConstruct(TrNotificationID ID,
unsigned int arguments_num);
```

Arguments

Argument	IN/OUT	Meaning...
`ID`	in	ID of the notification
`arguments_num`	in	number of arguments

TrNotificationGetID

This function fills the notification ID, and return the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrNotificationGetID(TrNotification *notification,
TrNotificationID *ID);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the given notification
`ID`	out	the ID of the notification

TrNotificationClone

This function clones a given TrNotification

Prototype

```
TRAPI_CPAPI TrNotification*   TrNotificationClone(TrNotification
*notification);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the notification to be cloned

TrNotificationDestroy

This function destroys a given TrNotification

Prototype

```
TRAPI_CPAPI void TrNotificationDestroy(TrNotification *notification);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the notification to be destroyed

TrNotificationArgIterCreate

This function creates a TrNotificationArgIter for a given notification, and returns NULL on failure.

Prototype

```
TRAPI_CPAPI TrNotificationArgIter
*TrNotificationArgIterCreate(TrNotification * notification);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the given notification

TrNotificationArgIterDestroy

This function destroys a given TrNotificationArgIter.

Prototype

```
TRAPI_CPAPI void TrNotificationArgIterDestroy(TrNotificationArgIter
*iter);
```

Arguments

Argument	IN/OUT	Meaning...
`iter`	in	the iterator

TrNotificationArgIterGetArgNum

This function fills the argument number, and returns the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrNotificationArgIterGetArgNum(TrNotificationArgIter
*iter, int *arg_num);
```

Arguments

Argument	IN/OUT	Meaning...
iter	in	the iterator
arg_num	in	the number of arguments

TrNotificationArgIterGetNextArg

This function fills the next TrArg in the TrNotification. When there are no more TrArgs, arg is filled with NULL, and the return code is TrOK. When failure occurs, the function returns the appropriate TrStatus error code, and arg is NULL.

Prototype

```
TRAPI_CPAPI TrStatus
TrNotificationArgIterGetNextArg(TrNotificationArgIter *iter, TrArg
**arg);
```

Arguments

Argument	IN/OUT	Meaning...
iter	in	the iterator
arg	in	the argument

TrNotificationSetIntArg

This function sets the argument in the given position to int argument, and overrides the current argument that exists in the given position.

Prototype

```
TRAPI_CPAPI TrStatus TrNotificationSetIntArg(TrNotification
*notification,int pos, int val, char * default_txt);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the given notification
`pos`	in	position of the notification
`val`	in	the value of the notification
`default_text`	in	notification text

TrNotificationSetStrArg

This function sets the argument in the given position to str argument, and overrides the current argument that exists in the given position.

Prototype

```
TRAPI_CPAPI TrStatus TrNotificationSetStrArg(TrNotification
*notification, int pos, const char * val, char * default_txt);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the given notification
`pos`	in	position of the notification
`val`	in	value of the notification
`default_text`	in	notification text

TrNotificationSetDoubleArg

This function sets the argument in the given position to double argument, and overrides the current argument that exists in the given position.

Prototype

```
TRAPI_CPAPI TrStatus TrNotificationSetDoubleArg(TrNotification
*notification, int pos, double val, char * default_txt);
```

Arguments

Argument	IN/OUT	Meaning...
`notification`	in	the given notification
`pos`	in	position of the notification
`val`	in	value of the notification
`default_text`	in	notification text

TrArgGetType

This functions fills the TrArg type, and returns the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrArgGetType(TrArg *arg, TrArgType *type);
```

Arguments

Argument	IN/OUT	Meaning...
`arg`	in	the argument
`type`	out	the argument type: string, int, double etc.

TrArgGetIntVal

This functions fills the int value, and returns the status of the operation. If TrArg is not an int, an error is returned.

Prototype

```
TRAPI_CPAPI TrStatus TrArgGetIntVal(TrArg *arg, int *val);
```

Arguments

Argument	IN/OUT	Meaning...
`arg`	in	the argument
`val`	out	value of the argument

TrArgGetDoubleVal

This function fills the double value, and return the status of the operation. If TrArg is not double, an error is returned.

Prototype

```
TRAPI_CPAPI TrStatus TrArgGetDoubleVal(TrArg *arg, double *val);
```

Arguments

Argument	IN/OUT	Meaning...
`arg`	in	the argument
`val`	out	value of the argument

TrArgGetStrVal

This function fills the string value, and returns the status of the operation. If TrArg is not a string, an error is returned.

Prototype

```
TRAPI_CPAPI TrStatus TrArgGetStrVal(TrArg* arg, char **str);
```

Arguments

Argument	IN/OUT	Meaning...
`arg`	in	the argument
`str`	out	the value of the string

TrArgGetDefText

This function fills the TrArg default text, and returns the status of the operation.

Prototype

```
TRAPI_CPAPI TrStatus TrArgGetDefText(TrArg *arg, char **def_text);
```

Arguments

Argument	IN/OUT	Meaning...
`arg`	in	the argument
`def_text`	out	the default text

Chapter 19
SecureClient Mobile

In This Chapter

Overview of SecureClient Mobile

SecureClient Mobile is a client for mobile devices that includes a VPN and a firewall. It replaces SecureClient for PocketPCs. The client works on various platforms and enables easy deployment and upgrade.

SecureClient Mobile's VPN is based on SSL (HTTPS) tunneling and enables handheld devices to securely access resources behind Check Point gateways.

The client can be triggered/controlled by 3rd party applications by exporting a programmable and extensible interface.

SecureClient Mobile has the following two modes of operation:

- **Centrally Managed Mode**: The client connects to a gateway (module) configured for SecureClient Mobile and downloads a set of policies that were sent to the gateway from the Security Management server. The client then enforces these policies.

  For this mode to work, a security gateway and the Security Management server must be configured to support the client. This mode is supported on the following gateway versions: R65, R62, R61_HFA1 and above, R60_HFA4 and above. A patch to the Security Management server/P1 is also needed on versions earlier than R65. The patch extends the schema (database) with relevant additions. Connectra gateway R62 and above supports this mode too.
- **SSL Network Extender Mode**: The client connects to a gateway configured only for SSL Network Extender. In this mode, the client does not download policies, but enforces a set of policies predefined upon client installation. The client works with any gateway configured for SSL Network Extender Network mode (available on Check Point VPN-1 Pro R55 HF10 versions and above, and on Connectra 2.0 versions and above). This is a backward compatibility mode that enables the running of a subset of the client features without upgrading the corporate infrastructure. For additional information on how to configure SSL Network Extender mode on a Check Point Security Gateway, refer to the *VPN User Guide*. For additional information on how to configure SSL Network Extender mode on a Connectra gateway, refer to the *Connectra Web Security Gateway* guide.

SecureClient Mobile is supported on the Windows Mobile 2003/SE/5.0 operating system.

Connectivity Features

When users access their organization from remote locations, it is essential that not only are the normal requirements of secure connectivity met, but also the following requirements of remote clients:

- **Secure Connectivity**: Secure connectivity is guaranteed by the combination of authentication, confidentiality and data integrity procedures employed for every connection.
- **Application Connectivity**: The remote client must be able to access the organization from various locations, even if it is behind a NAT device, proxy or firewall. The range of applications available must include web, mail-push, VoIP, and file sharing applications, in addition to other more specialized applications required by the corporation. Once authenticated, remote users begin a session. The session provides the context for which all encrypted traffic is processed until the user logs out (disconnects), or the session ends due to a timeout.
- **Usability**: There are several options that enable seamless connectivity for the end user. The client GUI and visual elements were designed with user friendliness and minimal interaction.

Session Continuation and Timeout

Once authenticated the client is assigned a session-id that is valid for a configurable duration. During this time if the client's VPN connection is dropped due to various networking conditions, the client uses its session-id to seamlessly reconnect the VPN connection, without disturbing the overall user experience. For example, this occurs if a user goes through a tunnel or enters an elevator. If user "moves" its internet connectivity from one interface to another, for instance from GPRS/UMTS to WiFi, SecureClient Mobile seamlessly reconnects maintaining connections that are open "above" the VPN tunnel.

The user may be prompted for authentication credentials five minutes before the session is timed-out. Once these credentials are accepted, the time-out interval is initialized. If the user does not provide in time it is disconnected from the server and must reconnect and re-authenticate the client manually. The user can also manually end its session by disconnecting the client.

This feature is on by default and can be configured using the `neo_user_re_auth_timeout` and `neo_implicit_disconnect_timeout` properties.

Initiate Dialup

SecureClient Mobile can be configured to initiate a dialup connection (for example, GPRS) if the device is not elsewhere connected to the Internet. Configure this feature using the `neo_initiate_dialup` property found in Table 19-1 on page 457.

Always Connected

SecureClient Mobile can be configured to automatically establish the VPN tunnel with the last gateway to which it was connected when one of the following conditions are met:

- The device has a valid IP address. For example, when turning WLAN/WiFi on.
- The device exits standby mode or loads after a softreset/shutdown.
- After the condition that caused the device to automatically disconnect ceases to exist (for example, when the client automatically disconnects (as a result of putting the device in ActiveSync), then when the device is released from ActiveSync the client automatically connects.)

This mode of operation is highly advised as the default mode of operations as it relieves the end user from the need to manually establish the VPN tunnel whenever connectivity is needed. Combining this mode with Initiate Dialup makes the client connected at all times. This allows push protocols, VoIP, etc.

Configure this feature using the `neo_always_connected` and `neo_disconnect_when_idle/timeout` properties found in.

Authentication Schemes

The client supports most of the common authentication schemes supported by Check Point gateways including Active Directory, RADIUS, etc. There are several ways to authenticate the user and the connected device:

- Client Certificates (X.509, Smartcards, etc.)
- One Time Password (RSA, SecureID, SoftID, etc.)
- User/Password combinations and multi-challenge-response ("legacy")

A connectivity policy downloaded to the device enables the administrator to define the amount of user interaction required to carry out the authentication process.

Some of these schemes can be configured to achieve seamless authentication (no user prompt for credentials):

- Certificates (through CAPI).
- User/Pass with credential caching. As long as the password cached, the user is not prompted to re-enter it when the client authenticates.
- OTP with SoftID and credential caching: The client reads the token code from the SoftID application transparently and the PIN can be cached.

Warning - When credential caching is enabled, the password/PIN is stored locally on the device. This poses a security threat because the password can be retrieved if the device is lost, stolen or hacked.

Some of these schemes can be configured to provide 2 factor authentication: Certificates with device-login (authenticate user to device login first, then use CAPI installed certificates to authenticate the device to CA), OTP Tokens including SoftID.

SecureClient Mobile supports a secure authentication (SAA) OPSEC interface that allows third party-extensions to the standard authentication schemes. This includes 3-factor, biometrics, and authentication.

Configure this feature using the `neo_remember_user_password/timeout`, `neo_user_auth_methods`, `neo_user_re_auth_timeout`, and `neo_saa_guilibs/url` properties.

Note - RSA SoftID is an authentication method that generates a unique, onetime passcode every 60 seconds used for secure access over the Internet. The passcode is generated using the PIN and obtained automatically. SecureClient Mobile gets the passcode from SoftID by communicating directly with the SoftID application. The SoftID application must be installed on the device but does not have to be running.
When the user has no PIN, either because the tokencard is new or the administrator reset the PIN, the PIN field is left blank. Once logged in, the user is directed to a page that requires a PIN to be created. This PIN is used for subsequent logins.
Prior to logging in, the token file (containing the shared token) must be imported into the SoftID application. This file is required for the authentication between the Authentication server (ACE/Server) and the SoftID application. The token file is protected by the pass_phrase. The pass_phrase may be obtained from the system administrator.

Support for Alternate Gateway

The client supports two methods for high availability and load sharing.

1. Clustering: Two or more gateways are logically placed behind one IP address providing full session continuation and load balancing in case of hardware failure.
2. DNS based clustering: Two or more gateways that provide the same topology and named with one DNS name. SecureClient Mobile can resolve a DNS name to all of its IP addresses. SCM can attempt to connect to up to 10 IP addresses per DNS name. SCM attempts to connect to each IP address until it succeeds. Support for this feature is transparent to any algorithm used by the DNS server for load sharing or priority ordering (e.g Round Robin) enabling effective load-sharing. This type of clustering does not provide session-continuation in case of a gateway failure.

Note - SecureClient Mobile does not support traditional MEP (Multiple Entry Points), but the equivalent of MEP with fully-overlapping-encryption-domain can be achieved using the DNS based clustering.

Gateway History

SecureClient Mobile retains the details of the gateways to which it was previously connected. This enables users to more readily access a gateway without having to re-enter the gateway’s information.

Allow Clear Traffic During ActiveSync and When Disconnected

Corporate users, who use SecureClient Mobile to access their corporate network from home or from the road with their mobile devices, may also wish to use SecureClient Mobile in the office where network traffic encryption is not necessary.

SecureClient Mobile can be configured to allow clear (not encrypted) traffic while in ActiveSync (the device is “cradled” to the PC, which serves as a kind of NAT access the network).

Traffic may also need to be sent unencrypted ("in the clear") when the mobile device is located in a private network inside the encryption domain. For example, when a Wi-Fi base station is located inside the corporate network.

Configuring the client not to allow clear traffic in ActiveSync effectively makes the client auto-disconnect if it is placed in cradle. And would also prevent the user from manually connecting as long as the device is cradled.

Configure this feature using the `neo_clear_in_activesync`, `neo_allow_clear_while_disconnected` and `neo_disconnect_when_in_enc_domain` properties.

Secure Configuration Verification (SCV) Traversal

SecureClient Mobile users can connect to a gateway that requires SCV validation. In cases where a gateway is configured to only allow access to clients that have passed the SCV checks, an exception can be made not to apply the SCV check to SSL clients. This includes SecureClient Mobile and SSL Network Extender (SNX). To enable this feature, navigate to **Global Properties > Remote Access > Secure Configuration Verification (SCV) > Exceptions**. Select the **Do not apply Secure Configuration Verification on SSL clients connections** checkbox.

Topology and Split Tunneling

A topology is the collection of enabled VPN links in a system of gateways, their VPN domains, hosts located behind each gateway, and the remote clients external to each gateway.

The administrator defines the list of networks and hosts accessible for the client once connected to the gateway. This list, the encryption domain, or VPN domain, is downloaded to the client after the initial connection and is used by the client to define what network traffic should be tunneled, encrypted to the gateway, and what traffic should not.

Hub Mode (VPN Routing for Remote Access)

VPN routing for remote access clients can be enabled through Hub Mode. In this mode, all traffic from the device is directed through the connected gateway. The gateway acts as a router for the remote client. When traffic from remote access clients is routed through the gateway, subsequent traffic can be filtered, inspected and kept in compliance.

Office Mode

This mode enables connections from within the corporate network to the remote access device and client-to-client connectivity (for example, P2P and VoIP protocols, back connections, and "push" technologies). In Office mode the connected gateway assigns an IP address to a remote client. This IP address is only used internally for secure encapsulated communication with the home network and is not visible in the public network. The IP address assignment takes place once the user connects and authenticates. The assignment lease is renewed so long as the user is connected. The address may be selected either from a general IP address pool, an IP address pool specified by the user group using a configuration file, A DHCP server or a RADIUS server. Office Mode is required for a correct configuration of SecureClient Mobile gateway.

Visitor Mode (SSL Tunnel)

Visitor mode enables the tunneling of all client-to-gateway communication through a SSL/TLS connection on port 443. Visitor mode is designed to traverse firewalls and proxy servers servers and NAT devices when the client needs to bypass the server to get "out" to the Internet and be able to reach the corporate gateway. Visitor Mode is required for a correct configuration of SecureClient Mobile gateway.

Security Policies and Client Decide

SecureClient Mobile supports several centrally managed policies that define the behavior of the client through sets of properties. When the client connects to the organization's gateway to establish a VPN tunnel, the policies are downloaded to the device and enforced by the client. The policies that are actually enforced by the client are those downloaded from the last connected gateway and they are enforced regardless if the client is currently connected, or not.

The administrator may wish to enforce on the clients some of the properties, while others may be left for the end user to define. This is enabled by the "client_decide" option available in most of the properties that can be applied to the client. This option's meaning is that the administrator does not wish to enforce this property, or behavior, on its clients. These can also be modified using the client CLI and programmables interface (API). Some of the properties are exposed for the end-user control through the options dialog.

If the gateway does not offer the client to download policies (see SSL Network Extender mode above) then the client enforces the packaged predefined policy. If that policy does not exist it enforces a "master" non-configurable pre-defined policy.

The client updates its policies each time it connects to a gateway. A timeout is applied for the policy validity, after which the client would pull the gateway for an updated policy.

IP Firewall Policy

SecureClient Mobile has a built in IP firewall, which supports predefined security policies One of these policies can be applied for the client to enforce:

- **Allow All**: All traffic is allowed. The client will still be protected by implicit firewall rules.
- **Allow Outgoing and Encrypted**: Permits incoming and outgoing encrypted traffic to and from the VPN domain. Also permits outgoing non-VPN connections that are initiated from the handheld. This policy is the recommended setting.
- **Allow Outgoing Only**: All outbound connections are permitted and all inbound connections are blocked. This policy will prevent incoming connections from being established from both the non-VPN hosts and VPN hosts.
- **Allow Encrypted Only**: Only VPN traffic originating from or destined to the encryption domain are permitted (and only when the client is connected).

The administrator can also define a few other rules to apply on the client firewall:

1. A policy to allow/disallow ActiveSync (device to PC sync) communications. This is useful when it is required that the users do not sync their device with an unauthorized PC.
2. A policy to drop all non-encrypted traffic destined to the VPN domain. This is mostly combined with Hub Mode and Allow Encrypted Only to achieve maximum security and stop address-leak prevention.
3. A policy that disables packet forwarding on the device. This disables IP Forwarding done by the device IP stack to disable the usage of the device as a router.

Configure these features using the `neo_enable_firewall_policy`, `neo_firewall_policy`, `neo_enable_activesync`, `neo_enable_ip_forwarding`, `neo_policy_expire`, and `neo_allow_clear_while_disconnected` properties.

Connectivity Policy

Allows the admin to define the way the client operates its connectivity features. This includes the Always-connected, Disconnect in ActiveSync, Hub Mode as described above. Some other properties include: ability to connect to a (new) gateway that was not pre-defined by the admin; the requirement for gateway finger-print approval etc.

General "GUI" Policy

Allows the administrator to define some general settings and behaviors of the client. These include: Running the client on device boot, The ability of the end-user to quit the client; to generate debug logs; to have the Today/Home item and the amount of interaction, pop-ups and warnings the client produces.

Configure these features using the `neo_run_client_on_device_startup`, `neo_enable_kill`, `neo_allow_client_debug_logs`, `neo_allow_client_db_export`, `neo_show_taskbar/today_item`, and `neo_flash_icon_on_encrypting/drop` properties.

Client Deployment, Repackaging and Upgrade

SecureClient Mobile comes packaged as a self-installing CAB signed for integrity. The CAB package can be customized before it is distributed to users to include predefined topology, settings, credentials, and a default firewall policy. The CAB can be re-packaged with a supplied MSI package (Windows Installer) to be installed through ActiveSync's installer service from a Windows PC. During version upgrades, the installer preserves the existing client policies and credentials unless they are specifically overridden by the upgrade package.

When the client is installed on the mobile device, another applet called Certificate Import Wizard is also installed. This applet enables the importing of PKCS#12 certificate files to the device CAPI store for use for client authentication.

Configure the upgrade is done using the `neo_upgrade_mode`, `neo_upgrade_version`, and `neo_upgrade_url` properties.

Installing SecureClient Mobile

SecureClient Mobile Gateway Side Installation

Full SecureClient Mobile (SCM) support is available starting from R65 for VPN1 gateways and Security Management server and from R62 for Connectra standalone gateways. For R62 Connectra gateway installation and configuration please refer to Connectra documentation.

For earlier gateways and Security Management server versions one should follow the below installation instructions.

In order to centrally manage the gateways, a management patch should be installed on the Security Management server earlier than R65, although this is not mandatory. If SCM support is only installed on the gateways, then configuration must be applied to each gateway individually.

A SecureClient Mobile user can connect to a gateway that does not have SCM support installed, or to a gateway with SCM support installed but not enabled, through the SSL Network Extender settings.

Module Support

SecureClient Mobile (SCM) support is built-in on these gateway versions:

- R60 HFA_04 or later
- R61 HFA_01 or later
- R62 or later.

Note - After installing the HFA one further "manual" step is needed in most cases. When HFA is installed it does not override any existing configuration files. Instead the configuration file are copied to the conf folder with a "_HFA" appended to the file name. Such configuration files should manually be renamed after copying any relevant configuration data into them. There are 3 configuration files that are part of the SCM support and should be renamed: `$FWDIR/conf/*_HFA.ttm` > `$FWDIR/conf/*.ttm`

Downloading HFAs

If you do not have an HFA installed, download the latest HFA for your gateway version from:

http://www.checkpoint.com/downloads/latest/hfa.html

Security Management Server Support

To centrally manage SCM supporting gateways you need the following Security Management server versions:

- R65 or later
- R62 with the SCM management patch
- R61 with the SCM management patch
- R60 with the SCM management patch

Downloading SCM Management Patch

The the relevant patch version from: http://www.checkpoint.com/downloads/

Product: SecureClient Mobile, Version: R60/R61/R62, OS: Windows Mobile

Choose the relevant add-on: 'Add-on for NNNN Based Security Management server' where NNNN is your Security Management server OS version.

Management Patch Installation

To install the management patch:

1. At the command prompt on the Security Management server, type:

   ```
   fw1_HOTFIX_AAA_HF_HANN_NNN_NN
   ```

2. When prompted, type **y** to continue with the installation.
3. At the command prompt on the Security Management server, run the following commands:

   ```
   cpstop
   ```

   ```
   cpdb scheme_adjust
   ```

   ```
   cpstart
   ```

4. Install policy..

Note - A separate patch is available for installation on a Provider-1. Please refer to Sk32210: SecureClient Mobile cannot be centrally managed using Provider-1/SiteManager-1.

Gateway Patch

This patch is required only for R60 HFA_02.

To install the gateway patch (for each gateway):

1. At the command prompt, type:

   ```
   fw1_HOTFIX_DAL_HF_HA02_129_591129NNN_N
   ```

2. When prompted, type **y** to continue with the installation.

After the installation is complete, reboot the machine.

Client Side Installation

There are two ways to install SecureClient Mobile:

- Self-installing CAB Package: This file is installed directly on the mobile device.
- Self-installing MSI Package: This file is installed on the user's personal computer. During installation, the installer extracts a CAB file package from within the MSI package and installs it on a connected mobile device using ActiveSync services.

Hardware and Software Requirements

Operating System

- Windows Mobile 2003/SE PocketPC
- Windows Mobile 5.0 PocketPC/Smartphone

Processor

- Intel ARM/StrongARM/XScale/PXA Series Processor family
- Texas Instrument OMAP Processor family

Check Point Certificates and Locked Devices

To install the client on a Widows Mobile device that is One/Two-Tier Mobile2Maket Locked you have to first install Check Point certificates into the device Trusted Certificates and SCP Store. This "locked" configuration is common especially with SmartPhone devices. This allows the client installer and executables, which are signed by Check Point certificates, to be installed and run. The Check Point certificates are packaged in a small CAB installer that needs to be installed once on the target device before any Check Point product is installed.

1. The installer, cpcert.cab, is found in the client ZIP package under the 'unlock_smartphone' folder.
2. Copy the .cab file to the device while in ActiveSync
3. Run the .cab on the device using File Explorer.
4. At the end of the installtion you should get a message: "cpcert.cab was successfully installed on your device."

CAB Package

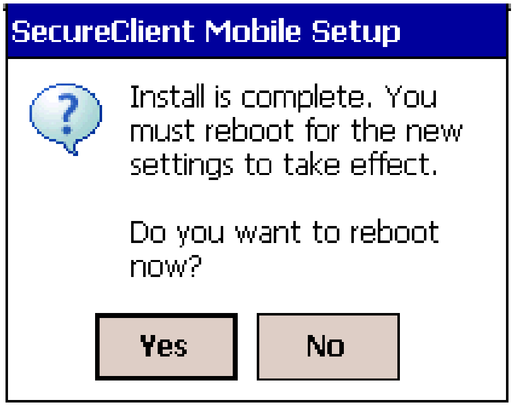
The .cab file is provided by the administrator and may be stored anywhere on the mobile device or an attached storage card. The installation can be automated using configuration tools such as Over The Air (OTA).

Installation

To install the CAB package:

1. On the mobile device in the **File Explorer** window, navigate to the folder where the .cab file was placed and select the .cab file.
2. If prompted to select an installation location. Select **Device**. (Installing on storage cards is not supported).

   The **SecureClient Mobile Setup** window opens.



SecureClient Mobile Setup

Install is complete. You must reboot for the new settings to take effect.

Do you want to reboot now?

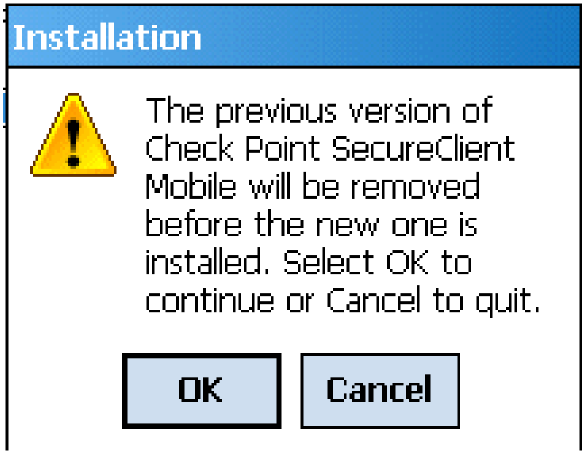
3. Tap **Yes** to reboot.

Upgrade

To upgrade the CAB package:

1. From the **File Explorer** window, select the .cab file.

The **Installation** window opens.



Installation

The previous version of Check Point SecureClient Mobile will be removed before the new one is installed. Select OK to continue or Cancel to quit.

2. Tap **OK** to install the new version. Existing configuration settings are not lost during upgrade, they are transferred to the new version.
3. If prompted to select an installation location. Select **Device.** (Installing on storage cards is not supported).
4. Tap **Yes** to reboot.

Uninstall

To uninstall the CAB package:

1. Select **Start > Settings > System Tab > Remove Programs.**
2. Highlight **Check Point SecureClient Mobile**, and then tap **Remove.**

MSI Package

The `.msi` file package is provided by the administrator and may be stored anywhere on the PC. The installation can be automated using tools such as Microsoft SMS Server.

Installation

To install the MSI package:

1. On the Windows PC machine, run the `.msi` file provided by your administrator.

2. Follow the instructions in the wizard to complete the installation. During installation, the ActiveSync service prompts users to install the software on the device.

Upgrade

To upgrade the MSI package:

1. Run the `.msi` file provided by your administrator.
2. Follow the instructions in the wizard to complete the installation. During installation, the ActiveSync service prompts users to install the software on the device.
3. Click **OK** to install the new version. Existing configuration settings are not lost during upgrade, but transferred to the new version.

Uninstall

To uninstall the MSI package:

1. Click **Start > Settings > Control Panel > Add Remove Programs**.
2. Highlight **Check Point SecureClient Mobile**, and then click **Remove**.
3. Follow the instructions in the wizard to complete the uninstallation. If you want to remove the client from the device, see "Uninstall" on page 440 to follow the CAB Package uninstall procedure.

Configuring SecureClient Mobile

In order for SecureClient Mobile clients to work in centrally managed mode, the following configuration is required:

- Configure a remote access community that includes all the supporting gateways.
- Enable load sharing and high availability features for each gateway.
- Configure Office Mode for each gateway.
- Define a topology for remote access for each gateway.
- Set global properties for SecureClient Mobile (neo properties).
- Establish connectivity settings.
- Define a security policy.
- Advanced properties can be set with GuiDBEdit tool.
- Enable and configure support for SecureClient Mobile on each gateway.
- Install a SecureClient Mobile License (SKU: CPVP-SCM-NNN) or eval license (SKU: CPVP-EVAL-SCM-25-30/1)

Note - When a gateway is configured for both SecureClient Mobile support and for SSL Network Extender support and a matching property is configured with different settings, the SSL Network Extender settings are applied (e.g. user authentication method, gateway certificate, SSL encryption method, etc.)

Note - In some of the following configuration points specific Security Management server database flags/properties are mentioned. These can be edited using the R65 SmartDashboard and/or GuiDBEdit. Please refer to the '"Advanced Configuration" on page 455' for details about GuiDbEdit.

Configuring a Gateway to Support SecureClient Mobile

There are two ways to configure a gateway to enable SCM support:

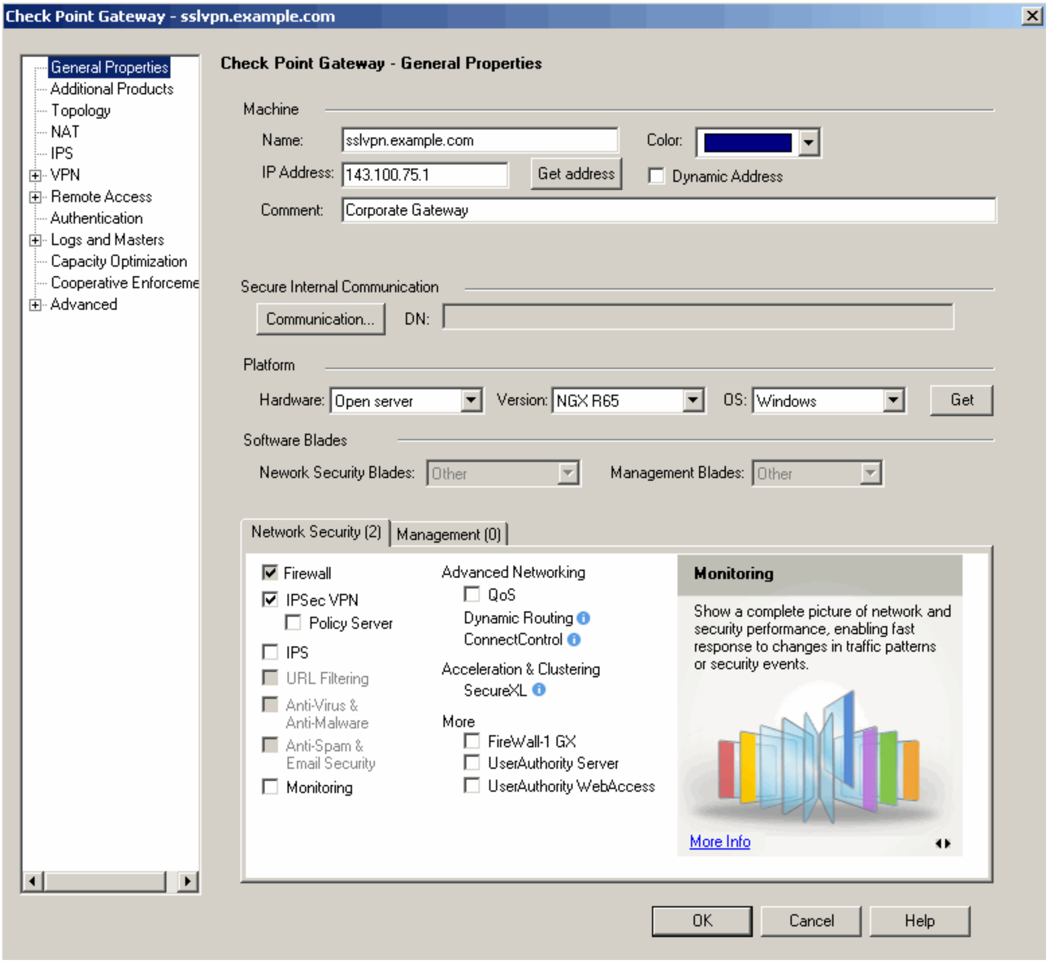
1. Using SmartDashboad, R65 and above: On the gateway object, navigate to **Remote Access > SSL Clients** check the **SecureClient Mobile** checkbox.
2. Enabling the `neo_enable` property on each gateway object using the GuiDBedit tool. (This method is available on R65 management, or if the SCM management patch was installed).
   A. Go to **Network Object > network_objects**.
   B. Select a gateway and search for the `ssl_ne` set within the VPN set. If the `ssl_ne` properties (such as `neo_enable` and `ssl_enable)` are not displayed, set the value of `ssl_ne` to `ssl_network_extender`. These properties are then displayed.
   C. Within the set, change the value of `neo_enable` to **true** to enable and **false** to disable support..
   D. Save the changes to install the policy.

Configuring the Gateway as a Member of a Remote Access Community

To configure a gateway as a member of a remote access community:

1. On **SmartDashboard**, select the **Gateway Object** from the **Network Object** tab of the **Objects Tree**. The **General Properties** window opens.

Figure 19-1 General Properties Window



Check Point Gateway - General Properties

Machine

Name: sslvpn.example.com

Color:

IP Address: 143.100.75.1

□ Dynamic Address

Comment: Corporate Gateway

Secure Internal Communication

DN:

Platform

Hardware: Open server

Version: NGX R65

OS: Windows

Software Blades

Nework Security Blades: Other

Management Blades: Other

☑ Firewall

☑ IPSec VPN

□ Policy Server

□ IPS

□ URL Filtering

□ Anti-Virus & Anti-Malware

□ Anti-Spam & Email Security

□ Monitoring

Advanced Networking

□ QoS

Dynamic Routing

ConnectControl

Acceleration & Clustering

SecureXL

More

□ FireWall-1 GX

□ UserAuthority Server

□ UserAuthority WebAccess

Monitoring

Show a complete picture of network and security performance, enabling fast response to changes in traffic patterns or security events.

2. Verify that **VPN** is selected.
3. Select **VPN** from the menu on the left.
4. Verify that the gateway participates in the remote access community. If not, add the gateway to the remote access community.

5. From the **Gateway Properties** page, in the **Topology** tab, configure the VPN domain for SecureClient Mobile in the same way that it was configured for SecureClient.

Note - The VPN domain can be used to configure SecureClient Mobile to work in hub mode, where all traffic is directed through a central hub.

The "Set domain for Remote Access Community ..." button on the **Topology** tab can also be used to create a different encryption domain for remote access clients that connect to the gateway.

6. Configure visitor mode, as described in the *Resolving Connectivity Issues* chapter in the *VPN Guide*. Configuring visitor mode does not interfere with regular SecureClient user functionality, but permits SecureClient users to enable visitor mode.

Note - The SecureClient Mobile uses TCP 443 (SSL) to establish a secure connection with the VPN SecurePlatform and the Nokia platform, and for remote administration purposes. Another port may be assigned to the SecureClient Mobile, however, this is not recommended, as most proxies do not allow ports other than 80 and 443. Instead, it is recommended that you assign SecurePlatform, or the Nokia platform web user interface, to a port other than 443.

7. On SecurePlatform, perform one of the following procedures:

   To change the webui port, run: `webui enable <port number>`. (For example, `webui enable 444`.)

   To disable the webui port, run: `webui disable`.

8. To change a Voyager port on a Nokia platform, run:

   `voyager -e x -S <port number>` (`x` represents the encryption level).

   For more information, run: `voyager -h`

9. Select **Remote Access > Office Mode**.

10. Configure office mode, as described in *Chapter 15, "Office Mode" on page 299.*

Note - Office mode support is mandatory on the gateway side.

11. Configure users and authentication.

Load Sharing Cluster Support

SecureClient Mobile provides load sharing cluster support.

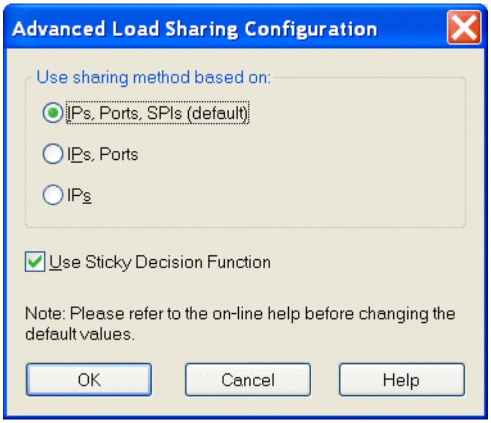
To enable load sharing cluster support:

1. Double-click the **Gateway Cluster Object** from the **Network Object** tab of the **Objects Tree.** The **Gateway Cluster Properties** window opens.

Note - A load sharing cluster must be created before you can configure the sticky decision function.

2. Select **Cluster XL**. The **Cluster XL** tab opens.
3. Click **Advanced.** The **Advanced Load Sharing Configuration** window opens.

Figure 19-2 Advanced Load Sharing Configuration window



Advanced Load Sharing Configuration

Use sharing method based on:

● IPs, Ports, SPIs (default)

○ IPs, Ports

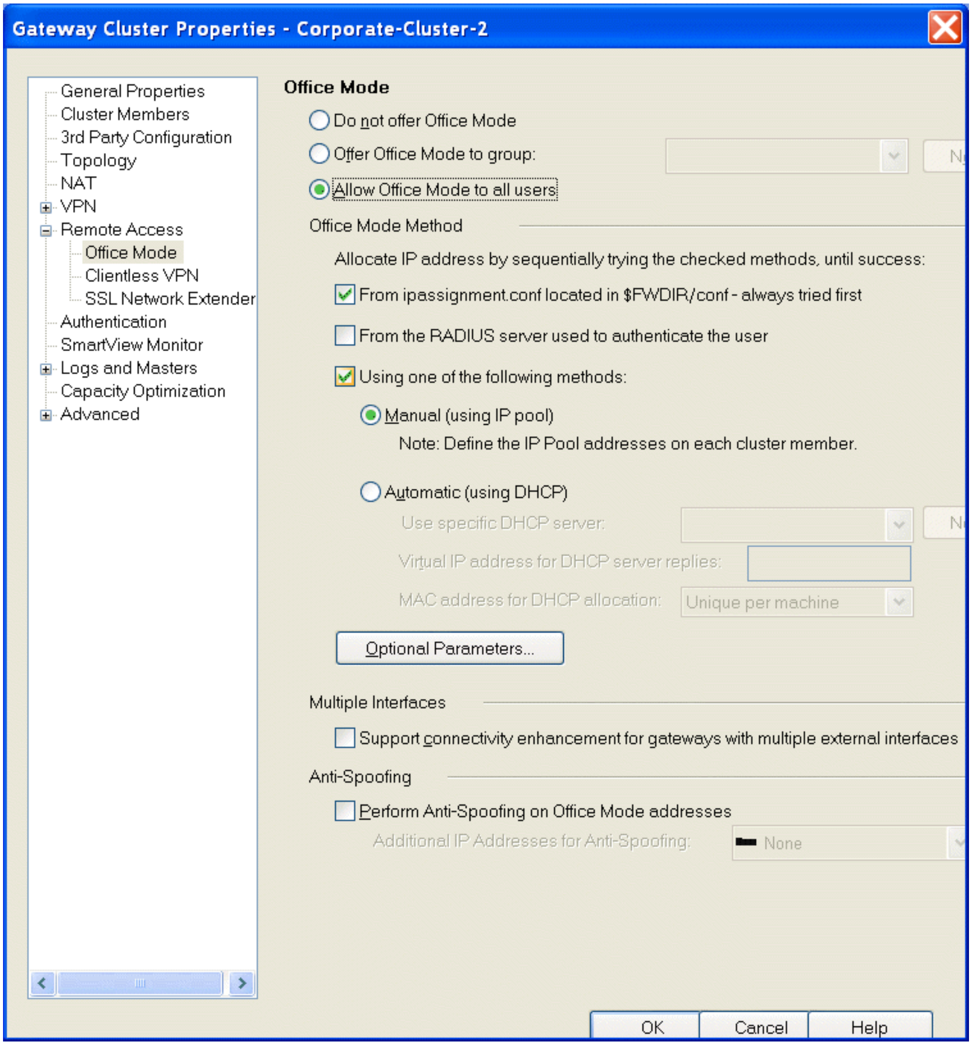
○ IPs

☑ Use Sticky Decision Function

Note: Please refer to the on-line help before changing the default values.

4. Select **Use Sticky Decision Function.** Using this function, when the client connects to the cluster, all of its traffic passes through a single gateway. If the member gateway fails, the client reconnects to another cluster member and resumes its session.

5. Select **Gateway Cluster Object > Remote Access > Office Mode**. When defining office mode for use with load sharing clusters, only the **Manual (using IP pool)** method is supported.



Office Mode

○ Do not offer Office Mode

○ Offer Office Mode to group:

● Allow Office Mode to all users

Office Mode Method

Allocate IP address by sequentially trying the checked methods, until success:

☑ From ipassignment.conf located in $FWDIR/conf - always tried first

□ From the RADIUS server used to authenticate the user

☑ Using one of the following methods:

● Manual (using IP pool)

Note: Define the IP Pool addresses on each cluster member.

○ Automatic (using DHCP)

Use specific DHCP server:

Virtual IP address for DHCP server replies:

MAC address for DHCP allocation: Unique per machine

Multiple Interfaces

□ Support connectivity enhancement for gateways with multiple external interfaces

Anti-Spoofing

□ Perform Anti-Spoofing on Office Mode addresses

Additional IP Addresses for Anti-Spoofing: None

Authentication Schemes

There are four ways to identify and authenticate a remote user

- **Certificate**: The system authenticates the user through a certificate. Enrollment is not permitted.
- **Certificate with enrollment**: The system authenticates the user through a certificate. Enrollment is permitted. If the user does not have a certificate, enrollment is permitted using a registration key provided by the system administrator.
- **Legacy**: The system authenticates the user through their username and password as well as other challenge-response options (for example, SecurID).
- **Mixed**: The system attempts to authenticate the user through a certificate. If the user does not have a valid certificate, the system attempts to authenticate the user through one of the legacy methods.

Note - The Certificate with enrollment feature is currently not implemented by the client. It will have the same effect as selecting Certificate option.

SecureClient supports a secure authentication (SAA) OPSEC interface that allows third party-extensions to the standard authentication schemes. . For this scheme to work Legacy scheme should be selected here. The `neo_saa_guilibs` peopery should also be updated with the SAA DLL name using the GuiDBEdit tool.

Configuring the Authentication Method

There are two methods used to configure the authentication method:

1. In SmartDashboard, click **Policy > Global Properties > Remote Access > SecureClient Mobile.** Select an authentication scheme from the **User authentication method** drop down menu.
2. Configure the authentication method using the `neo_user_auth_methods` property described in Table 19-2.

Re-authenticate Users

There are two methods used to configure the re-authenticate user value:

1. In SmartDashboard, click **Policy > Global Properties > Remote Access > SecureClient Mobile > Advanced.** Enter a value in the **Re-authenticate user every** field.
2. Configure the re-authenicate user value using the `neo_user_re_auth_timeout` property described in Table 19-2.

Configuring Encryption Methods

There are two encryption methods available:

- **3DES only**: (Default) The SecureClient Mobile client only supports 3DES.
- **3DES or RC4**: The SecureClient Mobile client supports both the RC4 and the 3DES encryption methods. (RC4 is a faster encryption method.)

There are two methods to configure the encryption method:

1. In SmartDashboard, click **Policy > Global Properties > Remote Access > SecureClient Mobile.** Select an encryption method from the **Supported encryption methods** drop down menu.
2. To determine whether the SecureClient Mobile client supports the RC4 or the 3DES encryption method, use the `neo_encryption_methods` property listed in Table 19-2.

Certificates

The Security Management server uses the same certificate for both SSL Network Extender and SecureClient Mobile clients when SSL Network Extender is enabled.

1. In SmartDashboard, open the gateway object and navigate to **Remote Access > SSL Clients.** Select the appropriate certificate from the **The gateway authenticates with this certificate** drop down menu.
2. This feature is configured using the the registry. See "Configuring a Non-Centrally Managed Gateway" on page 466. for more information.

Certificate Nickname

To view the certificate nickname

1. On **SmartDashboard**, open the **VPN** tab of the relevant network object.
2. In the **Certificates List** section, the nickname is listed next teach certificate.

Management of Internal CA Certificates

If the administrator has configured **Certificate with Enrollment** as the user authentication scheme, the user can create a certificate by using a registration key provided by the system administrator.

To create a user certificate for enrollment:

1. Follow the procedure described in “The Internal Certificate Authority (ICA) and the ICA Management Tool” in the *Security Management Server Administration Guide.*

Note - In this version, enrollment to an External CA is not supported.

2. Browse to the ICA Management Tool site, `https://<mngmt IP>:18265`, and select **Create Certificates**.
3. Enter the username, and click **Initiate** to send a registration keyto the user.

   When the user connects using SecureClient Mobile without a certificate, the **Enrollment** window opens, and the user can create a certificate by entering the registration key they received from the system administrator..

Note - The system administrator can direct the user to the URL, http://<IP>/registration.html, to receive a registration key and create a certificate even if they do not wish to use the SSL Network Extender at that time.

Importing a Certificate

To import a certificate using SecureClient Mobile, the certificate must already be on the Pocket PC and located in the `My Documents` directory.

To import a certificate:

1. Select **Start** > **Programs** > **Connection** > **CertImport**.

2. Click the certificate to be imported.
3. Enter the certificate password.
4. Select **Import issuer to Root CA** to import the certificate of the CA that was issued for the imported certificate. Use this feature when user and server certificates are issued by the same CA, for example a Check Point internal CA.
5. To view the additional certificate, select **Start > Settings > System > Certificates > Root**.
6. To view the personal certificate, select **Start > Settings > System> Certificates > Personal**.
7. Click **OK**. A window opens indicating that the certificate was imported successfully.
8. Click **OK**.

Topology Update

Topology updates are downloaded or updated to the client automatically each time that a user connects to a gateway and when a user reconnects after an authentication timeout occurs. It is also updated on a regular basis, as defined by the administrator. To define the frequency with which updated site details are downloaded to the client

1. In **SmartDashboard**, select **Policy > Global Properties > Remote Access**.
2. In **Topology Update**, select **Update topology every ... hours**.
3. Enter the frequency (in hours) with which the policy should be updated.

Security Policy

A security policy is created by the system administrator in order to regulate incoming and outgoing traffic. If a client connects and SecureClient Mobile support is not enabled, a default policy is enforced as defined in the client package. Use one of the following methods (listed in order of priority) to configure a security policy:

1. Using SmartDashboad, click **Policy > Global properties > Remote Access > SecureClient Mobile** and select **Yes** in the **Enable firewall policy** drop down menu. Set the **Firewall policy** and **Enable Microsoft ActiveSync** policy to the required values.

2. Using the GUIDBEdit tool on the Security Management server. For additional information, refer to "Advanced Configuration" on page 455.
3. Modifying the TTM files on each gateway. For additional information, refer to "Transform Template Files (TTM)" on page 466.
4. Modifying the `startup.C` file in a package.

When there are conflicting settings, that is one setting is configured differently in two locations, the settings configured in the highest priority location are applied. For example, if `neo_remember_user_password` is set to `true` in dbedit and `false` in the TTM file, SecureClient Mobile treats the property as `true`.

Route All Traffic (Hub Mode)

When Hub Mode is enabled, the gateway agrees to act as a VPN router for the client. All connections the client opens, either to the internal network or to other parts of the Internet, pass through the gateway. The packets are encrypted between the client and the gateway but pass "in clear" between the gateway and the client's peer. In addition, if the final destination of the connection is a machine behind another gateway, and a VPN link is defined between both gateways, the connection is routed along this link.

To enable Hub Mode in SmartDashboard:

1. On the gateway, open the **Remote Access** page and select **Allow SecureClient to route traffic through this gateway**.
2. Click **Policy > Global Properties > Remote Access > SecureClient Mobile** and select **Yes** in the **Route all traffic through gateway** drop down menu.

Alternatively, use the GUIDBEdit tool set the property `neo_route_all_traffic_through_gateway` to **true**.

Note - For more options, refer to sk31873 and sk31367.

Client Side Configuration

Connecting to a Site

To connect to a gateway:

1. On the toolbar, tap **Menu > Connect > New**. The **Connect to a new Server** window opens.
2. In the **Server address or name** field, enter the gateway information. If you are using Visitor mode to connect to a port other than the default, enter "<gateway information>:<port>".
3. Tap **OK**. The first time you connect to a server, the credentials need to be verified.
4. When prompted, enter your credentials.

Note - If you connected to a gateway, then tap **Connect** on the toolbar to connect to the most recently connected gateway.

To connect to the most recently connected gateway:

1. On the toolbar, tap **Connect**.
2. Select the server name or IP address of the gateway, or tap **Connect** on the toolbar to connect to the most recently connected gateway.

Configuring Display Settings

To configure display settings on the mobile device:

1. Select **Menu > Options...**.
2. Scroll down to **Display Settings**, and configure the following:
   - **Notification Level**: Select from the drop down menu one of the following options:
     - **All** - All popups from all categories (information, progress, warnings and errors) sent by the client are allowed.
     - **Progress, Warnings and Errors** - Select this option to eliminate Information popups from appearing.

    - **Warnings and Errors** - Select this option to allow only Warnings and Errors popups to appear.
    - **Errors only** - Select this option to eliminate all popups except those in the Errors category from appearing.
    - **None** - Select this option to prevent all popups from appearing.
- **Show Today Item**: Select this option to display SecureClient Mobile in the **Today Item** menu.
- **Show Taskbar icon**: Select this option to display the SecureClient Mobile icon on the taskbar when the client is running.
- **Flash icon on encrypting**: Select this option to display the **i** in the icon on the taskbar, which flashes when information is sending or receiving.
- **Flash icon on firewall packet drop**: Select this option to display the lock in the icon on the taskbar, which flashes when packets are dropped.

Status Page

The status page has two views, basic details and more details.

Basic details view contains:

- **Status**: Displays whether the client is connected to a gateway.
- **Server ID**: Displays the gateway name or IP address of the current connection.
- **Firewall policy**: Displays whether the firewall policy is enabled or disabled.

More details view contains:

- **Status**: Displays whether the client is connected to a gateway.
- **Server ID**: Displays the gateway name or IP address of the current connection.
- **Office mode IP**: Displays the office mode IP address that was assigned by the gateway.
- **Duration**: Displays the duration of the current session.
- **Firewall policy**: Displays whether the firewall policy is enabled or disabled.
- **ActiveSync policy**: Displays whether the ActiveSync policy is enabled or disabled.

Advanced Configuration

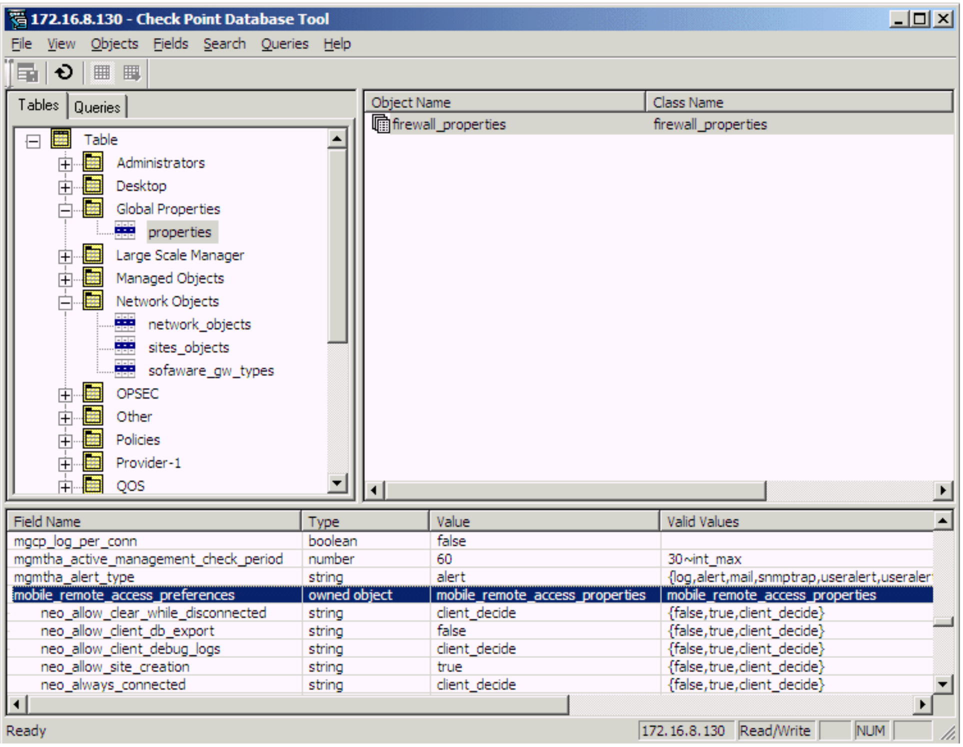
When the management patch is installed, the security policy is configured on the Security Management server using dbedit.

To configure the security policy using GuiDBEdit tool perform the following. This tool is available on the SmartGUIs installation in the following location: Program Files\CheckPoint\SmartConsole\<version>\PROGRAM\GuiDBEdit.exe (refer to SK13009).

1. Select **Global Properties > properties > firewall_properties**.
2. In the **Field Name** column, find **mobile_remote_access_properties**. The **SecureClient Mobile** properties appear below this property.
3. Customize the properties to meet the requirements.
4. Save the changes and select install policy.

The changes are not enforced until install policy is run. The policy is delivered to all gateways. Refer to Table 19-1 for a list of the properties used to configure the security policy.

Figure 19-3



Object Name	Class Name
firewall_properties	firewall_properties

Field Name	Type	Value	Valid Values
mgcp_log_per_conn	boolean	false	
mgmtha_active_management_check_period	number	60	30~int_max
mgmtha_alert_type	string	alert	{log,alert,mail,snmptrap,useralert,useraler
mobile_remote_access_preferences	owned object	mobile_remote_access_properties	mobile_remote_access_properties
neo_allow_clear_while_disconnected	string	client_decide	{false,true,client_decide}
neo_allow_client_db_export	string	false	{false,true,client_decide}
neo_allow_client_debug_logs	string	client_decide	{false,true,client_decide}
neo_allow_site_creation	string	true	{false,true,client_decide}
neo_always_connected	string	client_decide	{false,true,client_decide}

The security policy is configured using the properties described in:

- Table 19–1 *VPN Properties*
- Table 19–2 *Gateway Properties*
- Table 19–3 *Firewall Properties*
- Table 19–4 *General Properties*

Table 19-1 VPN Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_remember_user_password`	Remembers the user password/PIN (password caching). So long as the password is cached, the user should not be prompted to enter a password when the client connects, reconnects or re-authenticates.	**false**, true, client_decide
`neo_remember_user_password_timeout`	The password/PIN caching timeout (in minutes) since the user has entered their credentials. An authentication attempt after this timeout expires requires the user to re-enter their credentials.	-1 (infinite), 1 - MAX_INT, **1440**
`neo_always_connected`	Always connected. The client automatically connects to the last connected gateway: When the device has a valid IP address. When the device "wakes up" after it had low-power and after a soft-reset. After the condition that caused the device to automatically disconnect ceases to exist (Allow clear traffic during ActiveSync, Disconnect when idle).	false, true, **client_decide**

Table 19-1 VPN Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_always_connected_retry`	The always connected retry timeout (in minutes). If an automatic connection fails, the client tries to reconnect again and again on an interval set by this value. The client also tries to reconnect after the IP address of the client changes, or if the user manually requests a connection.	**1 (default)** -MAX_INT
`neo_initiate_dialup`	This flag instructs the client to automatically initiate an existing dialup connection (for example, GPRS). When the always connected flag is set to true, the user requests a connection, and there is no valid IP on the machine.	false, true, **client_decide**
`neo_disconnect_when_idle`	Disconnect when idle. Automatically disconnects the tunnel when there is no user interaction over a defined time period. A message balloon appears when the client disconnects.	false, true, **client_decide**
`neo_disconnect_when_idle_timeout`	Disconnect when idle timeout (in minutes).	**1 (default)** -MAX_INT

Table 19-1 VPN Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_user_approve_server_fp`	Requests user approval of server Finger Print (FP) before the client enters its credentials. The server FP is part of the gateway certificate provided in the SSL interaction with the client. The following options are available: **Once**: If the FP is seen for the first time by the client and not stored in the client database. **Always**: Prompts the user to approve the FP for every connection. **Never**: Always accepts the FP.	once, always, never, **client_decide**

Table 19-1 VPN Properties

Property	**Description**	**Valid Values (Default value in bold)**
neo_allow_site_creation	Enables the client to connect to a new gateway. When this flag is set to false, the client can only connect using the list of gateways configured in the client setup package.	false, true, **client_decide**
neo_block_conns_on_erase_ passwords	Blocks a connection upon the removal of passwords. If set to true, when the user clears the **Remember Password** option in the **Options** dialog, or selects the **Erase Passwords** menu option, the tunnel is automatically disconnected. A message balloon appears when the client disconnects.	false, true, **client_decide**
neo_disconnect_when_in_en c_domain	If the client is connected to a site, and an interface appears with an IP address located within one of the VPN encryption domains, the client disconnects. A message balloon appears when the client disconnects.	false, true, **client_decide**

Table 19-2 Gateway Properties

Property	**Description**	**Valid Values (Default value in bold)**
neo_enable	A gateway property which activates neo support.	**false**, true
neo_user_auth_methods	Client authentication methods.	certificate, certificate with enrollment, **legacy**, mixed
neo_encryption_methods	Client encryption methods.	**3DES only**, 3DES or RC4

Table 19-2 Gateway Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_upgrade_mode`	Client upgrade mode.	no upgrade, **ask user**, force upgrade
`neo_upgrade_version`	The client required version.	a number in hexadecimal format
`neo_upgrade_url`	Client download absolute URL.	
`neo_keep_alive_timeout`	The frequency with which the client sends keep-alive packets (in seconds).	10-MAX_INT, **20 (default)**
`neo_package_id`	The gateway allows only clients with these package IDs to connect (comma separated list).	
`neo_user_re_auth_timeout`	The session validity timeout (in minutes).	10~1440, **480 (default)**
`neo_saa_guilibs`	The DLL name or full path that is loaded for authentication with the server.	
`neo_saa_url`	The absolute URL for SAA authentication.	

Table 19-3 Firewall Properties

Property	Description	Valid Values (Default value in bold)
`neo_enable_firewall_policy`	Enables the firewall policy.	false, true, **client_ decide**
`neo_firewall_policy`	The supported firewall policies: Allow-all Outgoing only Outgoing and encrypted Encrypted only Block all (never disabled)	allow_all, outgoing_only, **outgoing_and_encrypted**, encrypted_only, block_all
`neo_enable_activesync`	Enables ActiveSync to PC (disabled if firewall is not installed).	false, true, **client_ decide**
`neo_enable_ip_forwarding`	Enables IP forwarding (when firewall is enabled).	**false**, true, client_ decide
`neo_enable_automatic_policy_update`	Automatically update the policy when it expires.	false, **true**, client_ decide
`neo_policy_expire`	The policy expiration timeout (in minutes).	-1 (infinite); 10-MAX_INT, **525600**
`neo_automatic_policy_update_frequency`	frequency with which the client updates policy files (in minutes).	5-MAX_INT, **120**
`neo_request_policy_update`	If set to true, the client prompts the user to update the policy upon policy expiration (`automatic_policy_update_frequency`). If the client is disconnected, the client attempts to update the policy after a connection is made.	false, **true**, client_ decide

Table 19-3 Firewall Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_route_all_traffic_through_gateway`	Routes all traffic through a gateway (in hub mode). This flag sets the routing in the IP routing table to send all traffic to the connected gateway, which results in all traffic leaving the machine (except for specific routes) to be encrypted and possibly re-routed from the gateway to the outside Internet. It allows for the inspection of all client data received that is examined by the connected gateway. This will only work if the gateway also supports routing all traffic. To configure the gateway, see"Route All Traffic (Hub Mode)" on page 452.	**false**, true, client_decide

Table 19-3 Firewall Properties

Property	**Description**	**Valid Values (Default value in bold)**
`neo_implicit_disconnect_timeout`	Retry to establish tunnel until this timeout elapse (in minutes).	1-MAX_INT, **2 (default)**
`neo_clear_in_activesync`	Enables clear traffic during ActiveSync. When set to **true** if the device is cradled (for example, when ActiveSync is activated to a PC using Bluetooth), the client automatically disconnects and the firewall settings permit clear traffic to exit the device to the encryption domain. This is required when the connected PC is located inside the encryption domain and the encryption of data is not necessary. A message balloon appears when the client disconnects.	false, true, **client_decide**
`neo_allow_clear_while_disconnected`	Enables clear traffic to the encryption domain when the client is disconnected. The client prevents clear traffic to the encryption domain from exiting the machine at all times except if this flag is set to true. Note: In an IPSEC client, this functionality is achieved using the VPN chain in the firewall. In SecureClient Mobile, this functionality is achieved using the firewall rule setting.	false, true, **client_decide**

Table 19-4 General Properties

Property	Description	Valid Values (Default value in bold)
`neo_run_client_on_device_startup`	Runs the client on device startup.	false, **true**, client_decide
`neo_enable_kill`	Specifies whether the user can stop the client. If this option is set to false, the quit option does not appear in the client menu.	false, true, **client_decide**
`neo_allow_client_debug_logs`	Enables the client troubleshooting window.	false, true, **client_decide**
`neo_allow_client_db_export`	Enables the client to export its local database to a clear text file is used to create a customized installation package.	**false**, true, client_decide
`neo_show_today_item`	Displays the today item.	false, true, **client_decide**
`neo_show_taskbar_item`	Displays the taskbar icon.	false, true, **client_decide**
`neo_flash_icon_on_encrypting`	Displays the flash icon, which monitors VPN tunnel activity (traffic).	false, true, **client_decide**
`neo_flash_icon_on_fw_packet_drop`	Displays the flash icon, which monitors firewall packet dropping activity.	false, true, **client_decide**

Configuring a Non-Centrally Managed Gateway

Configuring a Gateway to Support SecureClient Mobile:

This method must be performed on each gateway. To enable support:

1. From the command prompt, run `#ckp_regedit -a SOFTWARE\\CheckPoint\\VPN1 neo_enable 1`
2. Run `cpstop` and `cpstart` to enable support.

To disable support:

1. From the command prompt, run

   ```
   #ckp_regedit -d SOFTWARE\\CheckPoint\\VPN1 neo_enable
   ```

To configure a certificate to be used by gateway SSL (only if SSL Network Extender is disabled), proceed as follows:

1. Add the `neo_gw_certificate` key to `SOFTWARE/CheckPoint/VPN1` in the registry on each gateway.

To add the certificate, run the following command from the command prompt:

```
#ckp_regedit -a SOFTWARE/CheckPoint/VPN1 neo_gw_certificate
"cert_nickname"
```

To remove the certificate, run the following command from the command prompt:

```
#ckp_regedit -d SOFTWARE/CheckPoint/VPN1 neo_gw_certificate
```

If SSL Network Extender is disabled, and no certificate for SecureClient Mobile clients is defined, a certificate issued by the internal CA is used.

Transform Template Files (TTM)

The security policy is defined on each gateway individually using the TTM files when the management patch is not installed on the Security Management server. TTM files are found on each gateway in the `$FDIR/conf/` folder.

There are three types of TTM files:

- `vpn_client_1.ttm` (Refer to Table 19-1 for details.)
- `fw_client_1.ttm` (Refer to Table 19-3 for details.)
- `neo_client_1.ttm` (Refer to Table 19-4 for details.)

To configure the security policy using TTM files:

1. Open a TTM file using any text editor.
2. Set the default value for the property you are changing, for example:

```
:neo_request_policy_update ( :gateway ( :default (true)))
```

or

```
:neo_request_policy_update (
        :gateway (
                :map (
                        :false (false)
                        :true (true)
                        :client_decide (client_decide)
                )
                :default (true)
        )
)
```

3. Change the default setting, `true`, to create a new default setting for the security policy.
4. Save the file and select install policy.
5. The following property is used to set the policy expiration timeout for all policies, except the firewall policy: `:expiry ( :gateway ( :default (100)))`.

   The following property is used to set the firewall policy expiration timeout: `:expiry ( :gateway (neo_policy_expire :default (100)))`.

Configuration in a Mixed SecureClient and SecureClient Mobile Environment.

When connecting SecureClient Mobile to a security gateway that has already been setup to handle SecureClient connection (Windows, Macosx and 4.1) consider the following issues:

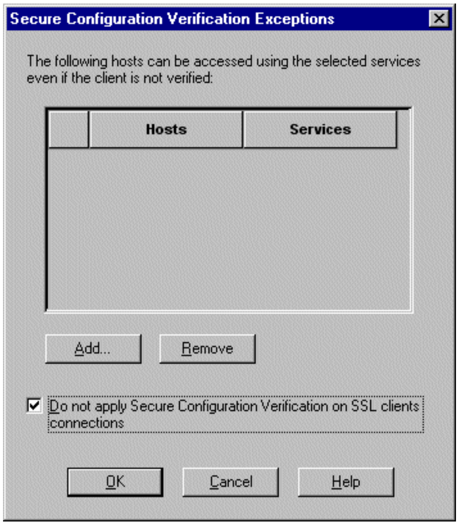
1. Topologies with several gateways and Multiple Entry Points (MEP): In some configurations the remote-access topology ("domain for remote access") is based on the gateway-to-gateway topology. If there are a few gateways with non-fully-overlapping encryption domain this may be a challenge for SecureClient Mobile as the topology downloaded to this client it only "knows" of the remote-access encryption domain of the (one) connected gateway. SCM does not support MEP too. To overcome this limitation it is advanced to use Hub Mode when connecting to any of the gateways. The client's packets will be routed through the connected gateway to other parts of the network using site-to-site VPN and will be routed to the Internet, as necessary.

2. Enable Secure Configuration Verification (SCV) Traversal: When enforcing SCV for remote-access on the gateways it is necessary to exempt SSL Clients from the SCV check, proceed as follows:

   A. In SmartDashboard, click **Policy > Global Properties > Remote Access > Secure Configuration Verfication (SCV)**.

   B. Select the **Apply Secure Configuration Verfication on Simplified mode Security Policies** option.

   C. Click **Exceptions**. The **Secure Configuration Verfication Exceptions** screen appears:



Secure Configuration Verification Exceptions

The following hosts can be accessed using the selected services even if the client is not verified:

	Hosts	Services

☑ Do not apply Secure Configuration Verification on SSL clients connections

   D. Select **Do not apply Secure Configuration Verification on SSL Clients Connections**.

Click **OK**.

Client Deployment Overview

SecureClient Mobile is packaged as a self-installing CAB (cabinet) or MSI (Microsoft Installer) file package. Users can install either package without specifying configuration details. This ensures the proper configuration of SecureClient Mobile software.

A CAB file package contains compressed files, which are mainly used to distribute software. The CAB file package is installed directly on the mobile device and has a `.cab` file extension.

The MSI package is installed on the user's personal computer and has a .msi file extension. It is used for a silent (unattended) installation. During installation, the CAB file package is extracted from the MSI package and installed on a connected mobile device using ActiveSync services.

Package Customization

The administrator obtains the SecureClient Mobile distribution package from the Check Point Download Center. The distribution package is located in a `.zip` file, which contains the folders/file listed in Table 19-5.

Table 19-5

Folder/File	Explanation
client_api	SecureClient Mobile API plus sample code.
client_pkg	SecureClient Mobile components for package customization.
tools	SecureClient Mobile tools (for instance MSI package and MSI packaging tool).
unlock_smartphone	This folder contains the Check Point certificate. Installing cpcert.cab enables the device to trust Check Point software, and successfully deploy SecureClient Mobile on locked Windows Mobile devices.
readme.txt	Readme file.

The unpacked client files are the same as those in the CAB package. The administrator can customize and package these files into a new CAB or MSI file package before distributing it to users. The customized package can include predefined topology and credentials, a default firewall policy and other settings.

During version upgrades, the installer retains the existing client policies and credentials that were not predefined in the upgrade package. The administrator can client upgrade using the `neo_upgrade_mode`, `neo_upgrade_version`, and `neo_upgrade_url` flags.

When the client is installed on the mobile device, another applet, called Certificate Import Wizard, is also installed. This applet enables you to import PKCS#12 certificates to the device.

The CAB and MSI packages can be edited by the administrator to customize the settings for SecureClient Mobile. The administrator can edit the package:

- Adding a file to the CAB package, for example, a user certificate file or a Secure Authentication (SAA) plug-in. For additional information, refer to "Adding a File to a CAB Package" on page 470.
- Deleting a file from the CAB package, for example, the `Cert_import` utility may not be needed for some configurations. For additional information, refer to "Deleting a File from a CAB Package" on page 471.
- Preconfiguring the client database parameters. For additional information, refer to "Exporting the Client Configuration" on page 472.
- Defining the client installation version. For additional information, refer to "Defining the Client Installation Version" on page 473.

Adding a File to a CAB Package

To add a file to a CAB package:

1. Obtain the SecureClient Mobile distribution `.zip` file from the Check Point Download Center site or from the CD.
2. Save the distribution `.zip` file to your local machine and extract its contents. One of the files is the `SecureClient_Mobile_<build number>.zip` file.
3. Extract `SecureClient_Mobile_<build number>.zip` to a folder (for example, `SCM)`. This creates a number of subfolders.
4. Copy and paste the file(s) to be included in the package to the `conf` folder (one of the extracted subfolders created in step 3).
5. In the `SCM` folder, open the `SecureClient_Mobile_Setup_<build number>.inf` file using a text editor.
6. In the `SecureClient_Mobile_Setup_<build number>.inf` file, add the name(s) of the file(s) to be included in the package to the following sections:

- In the `[conf]` section, add the name(s) of the file(s) starting on the line immediately after `startup.C`.
- In the `[SourceDisksFiles]` section, add the name(s) of the file(s) starting on the line immediately after `startup.C`. Every file in this section ends with an equal sign and a number, for example, `startup.C=7`. Add an equal sign and a number to the end of each file name that is added. The number represents the folder the file is placed in and corresponds to the `[SourceDisksNames]` section numbers.

7. Save the file.
8. Continue to "Creating a CAB Package" on page 473.

Deleting a File from a CAB Package

To delete a file from a CAB package:

1. Obtain the SecureClent Mobile distribution `.zip` file from the Check Point Download Center site or from the CD.
2. Save the distribution `.zip` file to your local machine and extract its contents. One of the files is the `SecureClient_Mobile_Setup_<build number>.zip` file.
3. Extract `SecureClient_Mobile_Setup_<build number>.zip` to a folder (for example, `SCM)`. This creates a number of subfolders.
4. In the SCM folder, open the `SecureClient_Mobile_Setup_<build number>.inf` file using a text editor.
5. In the `SecureClient_Mobile_Setup_<build number>.inf` file, delete references to the file(s) to be deleted in the following sections:
   - In the `[conf]` section, delete the name(s) of the unwanted file(s).
   - In the `[SourceDisksFiles]` section, delete the name(s) of the unwanted file(s).
6. Save the file.
7. Continue to "Creating a CAB Package" on page 473.

Exporting the Client Configuration

The administrator can provide all users with customized settings that are pre-configured on a SecureClient Mobile installation package. This is done by exporting an existing client configuration into a config file that is then added to a customized client CAB package. This customized package can then be installed/upgraded to the connecting devices.

To export the client configuration:

1. Install the client on a handheld device.
2. Configure a client with the required configuration, for example, configure the client's firewall options and connection to the gateways. You can now export the database with the current client configuration settings.
3. To export the database, locate the `database.C` file on the client, and in the `global properties` section of `database.C`, change the value of the property `neo_allow_client_db_export` to `true`.
4. Copy the `database.C file` to the `client` folder.
5. Restart the client.
6. In SecureClient Mobile, select **Menu > Help > Export db**. This exports the current settings to the `startup.C` file, which contains the nonconfidential data in the database.
7. Replace the `startup.C` file that is located in the `conf` folder of the preconfigured package. This file may be edited manually using a text editor in order to add or remove flags.

Note - Exporting `startup.C` will also export the global property `neo_allow_client_db_export` with the value set to **true**. To restrict users from exporting the client configuration, edit the `startup.C` and remove the property or set it to **false**.

Defining the Client Installation Version

The default client installation version is the client build number defined by Check Point.

To change the client installation version:

1. Obtain the SecureClient Mobile distribution `.zip` file from the Check Point Download Center site or from the CD.
2. Save the distribution `.zip` file to your local machine and extract its contents. One of the files is the `SecureClient_Mobile_Setup_<build number>.zip` file.
3. Extract `SecureClient_Mobile_Setup_<build number>.zip` to a folder (for example, `SCM`). This creates a number of subfolders.
4. In the SCM folder, open the `SecureClient_Mobile_Setup_<build number>.inf` file using a text editor.
5. Change the following attribute to the desired build number:
6. `NEO_VERSION_NUMBER= <build number>`
7. Save the file.
8. Continue to "Creating a CAB Package" on page 473.

Creating a CAB Package

A CAB package is created from the application files using the Cabwiz utility. Cabwiz can be downloaded and installed from the Microsoft Pocket PC 2003 SDK.

To create a CAB package:

1. To obtain the Cabwiz utility
   - Download the Microsoft Pocket PC 2003 SDK from here.
   - Install the SDK on your PC. After the SDK is installed, the Cabwiz utility is normally located at: `C:\Program Files\Windows CE Tools\wce420\POCKET PC 2003\Tools`.
2. Edit the package by exporting the client configuration and removing and/or adding files (for additional information, refer to "Adding a File to a CAB Package" on page 470, "Deleting a File from a CAB Package" on page 471 and "Exporting the Client Configuration" on page 472).
3. Copy the `Cabwiz.exe` and the `Cabwiz.ddf` files to the `SCM` folder created when extracting the `SecureClient_Mobile_Setup_<build number>.zip` file (this file was originally extracted from the SecureClient Mobile distribution `.zip` file).

4. Copy the `makecab.exe` from the Windows system directory (by default: `C:\WINDOWS\system32`) to the `SCM` folder.
5. Run the `Cabwiz SecureClient_Mobile_Setup_<build number>.inf` file. The created CAB package has a `.cab` extension.

Creating an MSI Package

The customizable install package (ZIP file) includes a Windows Installer package (MSI file) with a placeholder for a new client installer (CAB file). Once a client CAB is attached to the MSI file it can be installed on a Windows machine (PC) - the attached CAB would then be installed onto a connected PDA though the ActiveSync service. To attach a client CAB file to a MSI file, proceed as follows:

1. Obtain the SecureClient Mobile distribution `.zip` file from the Check Point Download Center site or from the CD.
2. Save the distribution `.zip` file to your local machine and extract its contents. One of the files is the SecureClient Mobile MSI file.
3. Run /tools/neo_msi_tool.exe SecureClient_Mobile.MSI SecureClient_Mobile.CAB

Configuring the SAA Plugin

Enabling the SAA plugin enables the ability to implement additonal authentication schemes (for example SoftID.) The plugin also allows customizing the login page.

To enable the SAA plugin using GuiDBedit:

1. Set the property `neo_saa_guilibs` to the SAA plugin name, for example `SAAPlugin.dll`.
2. Save the change and exit GuiDBEdit.
3. Install the updated policy.

Once the SAA plugin is enabled on the gateway, the client can be configured in one of two ways:

1. Manually
2. Using a predefined package

Configuring the SAA Plugin on the Client Manually

On the device:

1. Copy the SAA plugin into the following folder:

   ```
   \Program Files\CheckPoint\SecureClient_Mobile
   ```

2. Connect to the gateway. During the connection process, the defined SAA plugin pop-up appears.

In the event you receive the following error message, "Configuration Error: Failed to load SAA plugin," use the client login page (username-password) to connect. Once connected, quit and relaunch the client again.

Using a Predefined Package

This configuration is for situations where all the users use the SAA plugin to connect. In the event that only certain users are required to use the plugin, set the `neo_saa_guilibs` property to an empty string after you complete the creation of the customized package. As a result, only the users using the customized package will be using the SAA plugin.

1. Follow the steps described in "Configuring the SAA Plugin on the Client Manually" on page 475.
2. Establish a connection with each gateway that will be included in the package. This will store each gateway into the clients database.
3. Export the client configuration. To export the database, see "Exporting the Client Configuration" on page 472.

Use the exported `startup.C` to create the customized CAB file (include the SAA plugin in the CAB too). To create a CAB file, see "Creating a CAB Package" on page 473.

Troubleshooting

Enabling Log Files

Log files are files that records client activity, which are useful when troubleshooting various issues.

To enable log files:

- From the SecureClient Mobile GUI, select **Menu > Help > Troubleshooting**.
- Log files may be enabled for **Client**, the **VNA Kernel** (Virtual Network Adapter) and the **FW Kernel**.

Routing Table

The routing table is used by the TCP/IP stack to route IP packets on the device.

IP Configuration

The IP configuration page displays the IP addresses of the various interfaces.

Error Messages

Table 19-6 provides a list of error messages, their possible cause and a solution.

Table 19-6 Error Messages Troubleshooting

Error Message	Possible Cause	Solution
Cannot find the server (server name). Please check the server name and try again.	There is an error resolving the server name.	Check the server name and verify that the IP address is valid.
Error while negotiating with the server (server name). Please try again.	Error in client-server negotiation.	Try to connect again.
You are not permitted to access the server.	The user is not authorized.	Check that the user certificate is installed and is valid.

Table 19-6 Error Messages Troubleshooting

Error Message	Possible Cause	Solution
Your device is not connected to any network.	The network is not available for connection.	Connect the device to a network.
Your device is not connected to any network. Dialup connection is not available.	The network is not available for connection and dialup cannot be initiated. The settings may not be configured properly.	Check that your dialup settings are configured properly.
Access denied. Wrong username or password.	Wrong credentials supplied.	Ensure that the credentials are current and retry. If the credentials are cached, use the **clear passwords** button.
User is not permitted to have an office mode IP address.	The user attempting to connect is not configured to have an office mode IP address and therefore the connection failed.	Ensure that the user is configured to receive an office mode IP address.
The certificate provided is invalid. Please provide the username and password.	Invalid certificate provided.	Either install a new user certificate or connect with a username and password.
Connection to the server (server name) was lost.	There is no connection to the server, and the client disconnected.	Try to reconnect.
Security warning! Server fingerprint has changed during connection. Contact your administrator.	Server validation failed and therefore the connection failed.	Contact your administrator.

Additional Resources

For additional resources on setting up SecureClient Mobile, refer to:

How to add your own root certificate via CAB file.

How to add root certificates to Windows Mobile 2003 Smartphone and to Windows Mobile 2002 Smartphone.

Windows Mobile 5.0 Security Model FAQ.

ActiveSync 4.x Troubleshooting Guide.

Chapter 20

Packaging SecureClient

In This Chapter

Introduction: The Need to Simplify Remote Client Installations

As remote access to organizations becomes more widespread, administration of the remote client software becomes more difficult. Users often lack the technical expertise to configure the software themselves, requiring administrators to provide support for large numbers of users, many of whom may be geographically dispersed and using a wide variety of platforms. The administrator’s task is even more difficult if the organization has several groups of users, each of which requires a different configuration.

Administrators need a tool to automate the configuration of software to large user communities. This tools must enable the administrator to preconfigure the software, so that users do not have to do this themselves.

The Check Point Solution - SecureClient Packaging Tool

In This Section

Overview

The SecureClient Packaging Tool enables the administrator to create pre-configured SecureClient installation packages. Users can then use the configured package to install the software without being required to configure details, ensuring that users cannot inadvertently misconfigure their SecureClient software.

Pre-packaging can be done using either the:

- Check Point Packaging Tool Wizard
- MSI Packaging

The benefits of packaging are:

- Configuration (site creation, connection and encryption parameter specification, *etc*.) is performed by professional administrators, rather than by unsophisticated and error-prone users.
- Installation and support overhead are greatly reduced.
- Users' security configurations are more uniform across the organization, because they are pre-defined by the administrator rather than specified by each user individually.
- The administrator can more quickly respond to security threats by automatically updating remote users' security software.

How Does Packaging Tool Work?

Packaging Tool combines a client installation package (for example, the generic SecureClient installation package) with a package profile to create a preconfigured SecureClient package. The administrator can then distribute the package to the users.

The administrator can pre-configure the client's installation and configuration settings, such as the connection mode to the VPN gateway (Connect/Transparent), encryption properties and more. These settings are saved in a package profile, and can then be used for configuring packages.

The administrator can create different package profiles for different user groups. For example, the administrator can create one profile with the configuration parameters for Windows XP users, and another for Windows 98 users. The administrator can save all the profiles in a central database.

To allow the client to connect to the organization from the moment it is installed, the administrator can specify Partial Topology information for a site, that is, the IP address of the site or of its Security Management server. This information is included in the package. The first time the user connects to and authenticates to the site, the site's full topology is downloaded to the client.

The SecureClient package can also include scripts to be run after the installation of SecureClient.

The MSI Packaging Solution

MSI is a standard file format for application distribution in a Windows environment. Once a profile is created, it is saved and may be distributed to SecuRemote and SecureClient users.

The MSI package installs SecuRemote/SecureClient Extended View with default settings and can be customized using the command line based tool - `cpmsi_tool`.

Split Installation

When used with 3rd party software distribution systems, the connection to the distribution server is broken once the SecuRemote/SecureClient kernel is installed; the result is that the distribution server is not aware that the installation ended.

In order to resolve such cases a Split Install feature is available.

Creating a Preconfigured Package

In This Section

The Packaging Tool wizard guides an administrator through the process of creating a preconfigured SecureClient installation package. Each package can contain a different combination of a SecureClient version and a pre-configured profile.

You create a package in two essential stages:

1. Configuring and saving a package profile. The profile contains all the settings to be installed by the package by default.
2. Applying the profile to an exisitng installation package, thus creating a properly preconfigured package.

Creating a New Package Profile

1. To create a new profile, Select **Profile > New**. Enter the profile details and press **Next**.
2. The Packaging Tool wizard will guide you through the next several windows, in which you should configure different parameters regarding the user's profile, such as policy, encryption, topology (including Partial Topology information), certificates, client installation and logon parameters (SDL/Gina DLLs). For information about these features, see the relevant chapters in the documentation.
3. After pre-configuring all of the client's settings, you will be presented with the **Finish** screen. In this screen you can decide if Packaging Tool should continue to create a new package containing the changes as they appear in the profile or finish the process of profile generation without creating a new package. The options in this screen are:
   - **No, Create Profile only** — The profile will be created according to the setting you have pre-defined in the wizard and you will be returned to the main Packaging Tool window.

- **Yes, Create profile and generate package** — If you choose this option the profile you've created will be saved and you will be taken to the package generation wizard. For instructions regarding this wizard, proceed to "Generating a Package" on page 484.

You can always create packages from a saved profile at a later time.

Generating a Package

This section describes how to generate a SecureClient package according to the settings defined in a package profile.

Preparation

If you have not already prepared a base package, do so now, as follows:

1. Obtain an original SecureClient installation package. This package will be the base package, upon which the Packaging Tool will create the new custom SecureClient package.
2. Copy the clean SecureClient package to an empty directory. If the package is zipped or tarred you should unpack the package to the empty directory.

Once you have a base package, proceed as follows:

3. Run the SecureClient package generation wizard. You can run the wizard immediately after creating a new package profile (by selecting **Yes, Create profile and generate package**), or from the main Packaging Tool window by highlighting a previously created profile and selecting **Profile>Generate.**
4. You will be asked to enter a package source and destination folders.

   Under **Package source folder**, select the directory in which the original SecureClient installation you prepared in step 2 is located. Make sure you select the directory in which the SecureClient setup files actually exist and not a higher level directory.

   Under **Package destination folder and file name**, select an empty directory to which the new package will be copied, and enter a name for the file being generated.

   Press **Next** to continue to the next window.
5. If the package details can not be extracted from the package, you will be prompted to enter the package details (operating system type, SecureClient version and service pack). If the package details conflict with another package

(for example, you've updated the Windows 98 SecureClient package from FP3 to NG with Application Intelligence), you will be prompted to approve the replacement of the older package with the newer one.

The Packaging Tool will perform the actions you requested.

Adding Scripts to a Package

To specify that a script should be run after the user installs or uninstalls SecureClient, proceed as follows:

1. Edit the `product.ini` file.
2. To specify a post-installation script, add the file's name to the `[install]` section.
3. To specify a post-uninstallation script, add the file's name to the `[uninstall]` section.

The script should be accessible through the OS `PATH` variable.

The script is not part of the package, and should be transferred to the client separately.

Configuring MSI Packaging

To customize a profile used for remote users save the `.msi` file provided by Check Point. Once the file is saved, configurable files may be extracted from the file, customized, and then placed back into the file. To edit one of the configurable files:

1. Use `cpmsi_tool <SC-MSI-package-name> out <file-name>` to extract the file from the package.
2. Customize the file.
3. Use `cpmsi_tool <SC-MSI-package-name> in <file-name>` to insert the file back into the package.

The configurable files are:

- `product.ini`
- `userc.c`
- `userc.set`
- `reg.ini`
- `SecuRemoteAuthenticate.wav`
- `SecuRemoteConnected.wav`
- `SecuRemoteDisconnected.wav`
- `SecuRemoteFailed.wav`
- `logo.bmp`
- `logging.bat`
- `install_boot_policy.bat`
- `collect.bat`
- `scvins.bat`
- `scvuins.bat`
- `msfw.bat`
- `harden.bat`

Add and Remove Files in Package

To add new files to the package:

```
cpmsi_tool <SC-MSI-package-name> add <file-name>
```

To remove a newly added file:

```
cpmsi_tool <SC-MSI-package-name> remove <file-name>
```

Installation Command Line Options

The following are the command line parameters.

Table 20-1 Installation Command Lines

Parameter	Description
/i pkg_name	Install
/x pkg_name	Uninstall
/q	Quiet installation
/l*v log_file_name	Collect logs

Split Installation

To activate:

1. Set `SplitKernelInstall=0` in the `product.ini` file.
2. Install the product except for the kernel.
3. An automatic reboot, initiated by the end user, will occur.
4. After the reboot, the automatic kernel installation takes place.
5. A second automatic reboot will occur

Debug

In order to debug the MSI installation, run the `/l*v log_file_name_parameter`. `log_file_name` and `install_securemote.elg` are used for troubleshooting.

Zone Labs Endpoint Security Client

When installing the SecureClient MSI package with Zone Labs integration, use the following syntax:

```
msiexec /i <package_name> [ZL=1] [INSTALLDIR=<install_dir>]
[/qr|/qb|/qb!]
```

- package_name - the SecureClient msi package name
- ZL=1 - install with Zone Labs configuration
- INSTALLDIR=<install_dir> - the folder where the package is installed
- [/qrl/qbl/qb!] - standard MSI UILevel support used for silent installation.

Using this command, the `product.ini` file is automatically modified.

Chapter 21 Desktop Security

In This Chapter

The Need for Desktop Security

A security gateway protects a network by enforcing a Security Policy on the traffic to and from that network that passes through the security gateway. A remote client, located outside the protected network, is vulnerable to attack because traffic to the remote client does not pass through the security gateway — no Security Policy is enforced on this traffic.

There is a further danger: an attacker might gain access to a protected network by compromising a remote client, which may in turn compromise the protected network (for example, by relaying a virus through the VPN tunnel). Even if the security gateway enforces a very restrictive Security Policy, the LAN remains vulnerable to attacks routed through unprotected remote clients.

Desktop Security Solution

In This Section

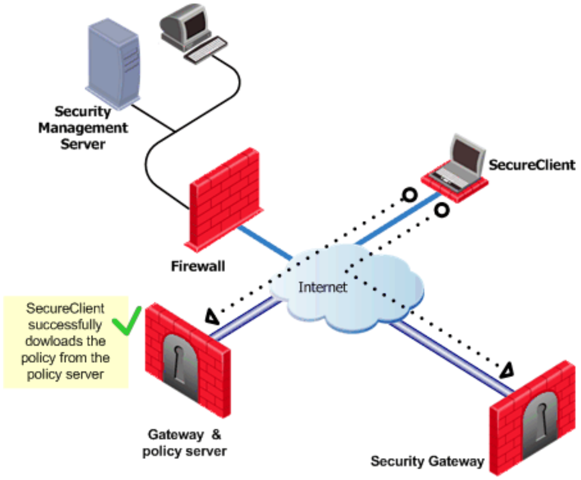
Introducing Desktop Security

Check Point VPN SecureClient protects remote clients by enforcing a Desktop Security Policy on the remote client. The administrator defines the Desktop Security Policy in the form of a Rule Base. Rules can be assigned either to specific user groups or to all users, enabling the definition of flexible policies.

The Desktop Security Policy is downloaded by the Security Management server to a Policy Server, a module installed on a security gateway, which serves as a repository for the Desktop Security Policy. SecureClient machines download their Desktop Security Policies from the Policy Server.

When a SecureClient connects to the organization's gateway to establish a VPN, it can connect to a Policy Server as well and retrieve its Desktop Security Policy and begin enforcing it. SecureClient can accept, encrypt or drop connections depending on their Source, Destination and Service.

Figure 21-1 Retrieving the Desktop Security Policy from a Policy Server


Security Management Server
SecureClient
Firewall
Internet
SecureClient successfully dowloads the policy from the policy server
Gateway & policy server
Security Gateway

The Desktop Security Policy

The Desktop Security Policy is divided into two parts:

- inbound rules — enforced on connections destined **to** the SecureClient machine
- outbound rules — enforced on connections originating **from** the SecureClient machine

A rule in a Desktop Security Policy specifies the following:

- source
- destination
- service
- action (accept, drop, encrypt)
- track (log, alert)

Connections to machine inside the organization (i.e. all the machines in the VPN domain of the gateway) are automatically encrypted, even if the rule allowing them to pass is an accept rule.

Implied Rules

In addition to the inbound and outbound rules explicitly defined by the administrator, implicit "cleanup" rules are automatically appended at the end of both the inbound and outbound policies:

- The outbound implicit rule allows all connections originating from the SecureClient machine, thus allowing connections which do not match any of the previous rules.
- The inbound implicit rule blocks all connections destined to the SecureClient machine which do not match any of the previous rules, on the assumption that what is not explicitly allowed is to be rejected.

User Granularity

You can define different rules for remote users based on location and user groups:

Location — For example, you can define a less restrictive policy for a user connecting from within the organization (e.g., a user with SecureClient installed on his laptop connecting from within the organization), and a more restrictive policy for the same user connecting from outside the organization (from a hotel room). This is done in the user's security Rule Base by configuring the location of the users for which the rule should be implemented.

User Groups — For example, you can define restrictive rules for ordinary users, but allow system administrators more access privileges.

In addition, you can define rules to be enforced for all remote users, by not specifying a specific user group, but rather all users.

Rules do not specify individual users but rather user groups. Because SecureClient does not know to which groups the currently logged-in user belongs, it must obtain this information from the Policy Server as follows: after SecureClient authenticates itself to the Policy Server, the Policy Server resolves the user groups to which the user belongs and sends this information to SecureClient. Once SecureClient knows to which groups the user belongs, it is able to enforce the rules defined for that user. Rules can also be applied to radius groups on the RADIUS server.

Default Policy

When SecureClient is started, and before it connects to the Policy Server, it enforces a "default policy," which consists of the rules defined for all users in the last policy downloaded from the Policy Server. This is because at this point, SecureClient does not know to which groups the user belongs. The default policy is enforced until the user downloads an updated policy (and the current user's groups information) from a Policy server.

If a SecureClient loses its connection to the Policy Server, it enforces the default policy until the connection is restored and a Policy is downloaded.

Policy Server

What is a Policy Server?

A Policy Server is a module installed on a security gateway, which serves as a repository for the Desktop Security Policy. SecureClient machines download their Desktop Security Policies from the Policy Server.

High Availability and Load Balancing

For load balancing and high availability, you can configure multiple Policy Servers. Load balancing can be especially important at times of high load, for example, when a large number of users log in at the beginning of the work day.

Connect Mode

- **High Availability between all Policy Servers, trying selected first** — SecureClient will always try the specified Policy Server first. If this server is unavailable, the SecureClient will randomly choose a different server from among the remaining servers.
- **High Availability only among selected Policy Servers** — SecureClient will randomly choose a Policy Server from the specified group. This option provides Load Balancing as well, since the load will be more equally distributed among the Policy Servers.

Policy Download

When is a Policy Downloaded?

When a user creates a site in SecureClient, a list of Policy Servers is downloaded to the client’s machine. If the user is using Connect mode (the default mode), a policy will be automatically downloaded from a Policy Server when the SecureClient machine connects to the site. The automatic policy download can also be disabled — this configuration is controlled in the user’s profile.

Policy Renewal

If a time-out is defined for a policy (default is 60 minutes), SecureClient will reconnect to the Policy Server to download a new policy when half specified time period has elapsed. If more than one Policy Server is defined (see "High Availability and Load Balancing" on page 494) SecureClient will try to reconnect to the Policy Server from which it last successfully downloaded a policy. If it cannot connect to that Policy Server, it will try the others.

If SecureClient cannot download a new policy from any Policy Server, it will try again after a fixed interval (default is 5 minutes). If SecureClient fails to download a new policy after the timeout expires, it will revert to the default policy.

Logs and Alerts

Logs and alert specified Desktop Security Policy can be viewed using the SecureClient **Diagnostics**. In addition, all alerts are saved and uploaded to the Security Management server. When the client connects to a Policy Server, it uploads the alerts to a spooling process, which passes the alerts to the Security Management server, where they can be viewed by the SmartView Tracker.

Desktop Security Considerations

In This Section

Planning the Desktop Security Policy

You should carefully plan your users policy, properly balancing considerations of security and convenience. The policy should allow the desktop users to work as freely as possible, but at the same time make it hard to attack the remote user's desktop. Here are some points to consider:

- You should not explicitly allow any service to be opened to the SecureClient (i.e. allow a service in the inbound policy), unless the user has a specific server running on that port. Even if you do allow connections to be opened to the client, be very careful about who is allowed to open the connection, and from where.
- A restrictive policy (e.g., allow only POP3, IMAP and HTTP and block all the rest) will make it more difficult for your users to work. If you allow only specific services in the outbound policy and block all the rest, every time you will find out that a certain service is needed by your users, you will have to change the outbound policy and make sure the clients poll the new policy. The best way to implement the outbound policy is to use rules only to block specific problematic services (such as Netbus) and allow the rest.
- Outbound connections to the encryption domain of the organization are always encrypted, even if the outbound rule for the service specifies “accept”.
- Keep in mind that the implied rules (see “Implied Rules” on page 492) may allow or block services which were not explicitly handled in previous rules. For example, if your client runs a server on his machine, you must create an explicit rule allowing the connection to the client's machine. If you do not, the connection will be blocked by the inbound implicit block rule.

Avoiding Double Authentication for Policy Server

When using Policy Server High Availability, it is possible that users will connect to the organization through one gateway and to a Policy Server which is installed on a different module. In this case they will be prompted twice for authentication — once for the gateway module and the other for the Policy Server. If a user usually connects to the organization through a specific gateway, and this gateway has a Policy Server module installed on it, this double authentication can be avoided by configuring the user's profile to use the **High Availability among all Policy Servers, trying selected first** option, and selecting the primary Policy Server as that one the gateway through which the user usually connects to the organization. This way, after the user authenticates to the gateway, he will automatically be authorized to download the security policy from the Policy Server installed on that gateway.

Configuring Desktop Security

In This Section

Server Side Configuration

1. Install the Policy Server add-on module from the Check Point installation CD. The Policy Server add-on should be installed only on machines that have security gateway modules installed on them.
2. Open the gateway object on which you have installed a Policy Server and select the **General Properties** tab. In the **Check Point Products** section select **SecureClient Policy Server**.
3. Go to the **Authentication** tab. In the **Policy Server > Users** section select a group of users that is allowed to retrieve policies from this Policy Server.
4. Repeat steps 2 and 3 for each additional Policy Server.
5. Go to **Policy > Global Properties** and select the **Remote Access** tab. In **Revert to default policy after**, select the time-out for desktop security policies (see "Policy Renewal" on page 495 for more information).
6. In the policy selection toolbar, select **Desktop Security**.
7. Configure the inbound rules. Using the **Rules>Add Rule** menu item, you can add rules to the policy.

   In inbound rules, the SecureClient (the desktop) is the destination, and you can specify the users to which the rule is to be applied.
8. Configure the outbound rules.

   In outbound rules, the SecureClient (the desktop) is the source, and you can specify the users to which the rule is to be applied.
9. Install the policy. Be sure to install both the Advanced Security policy on the gateways and the Desktop Security policy on your Policy Servers.

Client Side Configuration

Connect Mode

1. Double click your SecureClient icon at the bottom right side of you desktop and press **Properties**.
2. Choose **Logon to Policy Server** if you wish to logon to a Policy Server automatically after connecting to a site.
3. Select **Support Policy Server High Availability** if your site has several Policy Servers and you want SecureClient to attempt to load balance between them. If you choose to use this feature you must select one of the following:
   - **High Availability among all servers, trying selected first** — Select the primary Policy Server.
   - **High Availability only among selected servers —** Select the servers to which you wish to connect.

As an administrator, you can eliminate the user's need to configure these steps by creating a custom profile for them.

Chapter 22

Layer Two Tunneling Protocol (L2TP) Clients

In This Chapter

The Need for Supporting L2TP Clients

For some organizations there are clear benefits to be gained by using the Microsoft IPSec client for remote access to internal network, rather than the more feature rich and secure Check Point SecuRemote/SecureClient.

Reasons for using the Microsoft L2TP IPSec client include the fact that is an inherent part of the Windows 2000 and Windows XP operating systems, does not require an additional client to be installed, and is free.

Solution - Working with L2TP Clients

In This Section

Introduction to L2TP Clients

Check Point Security Gateways can create VPNs with a number of third party IPSec clients. This explanation focuses on the Microsoft IPSec/L2TP client.

You can access a private network through the Internet by using a virtual private network (VPN) connection with the Layer Two Tunneling Protocol (L2TP). L2TP is an industry-standard Internet tunneling protocol.

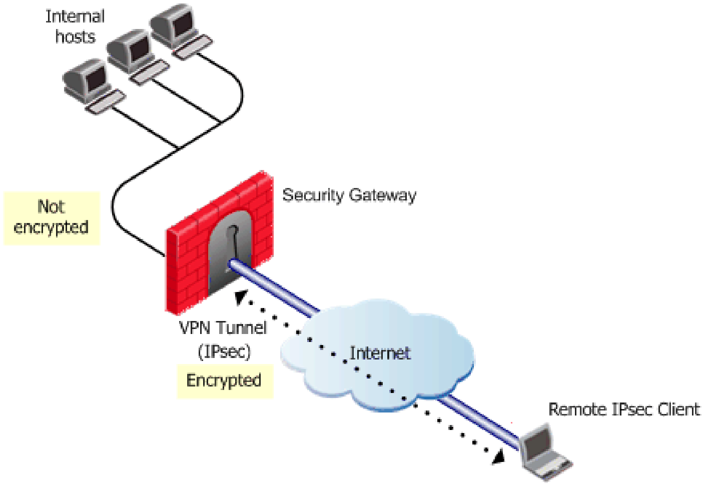
Check Point supports the Microsoft IPSec/L2TP client on Windows 2000 and Windows XP machines. The client is an integral part of the Windows operating system.

Creating a Remote Access environment for users with Microsoft IPSec/L2TP clients is based on the same principles as those used for setting up Remote Access for SecuRemote/SecureClients. It is highly recommended to read and understand Chapter 14, “Introduction to Remote Access VPN”” before attempting to configure Remote Access for Microsoft IPSec/L2TP clients.

Establishing a VPN between a Microsoft IPSec/L2TP Client and a Check Point Gateway

To allow the user at the Microsoft IPSec/L2TP client to access a network resource protected by a security gateway, a VPN tunnel is established between the Microsoft IPSec/L2TP client and the gateway, as shown in Figure 22-1.

Figure 22-1 IPSec Client to Check Point Gateway Connection



The process of the VPN establishment is transparent to the user, and works as follows:

1. A user at an Microsoft IPSec/L2TP client initiates a connection to a security gateway.
2. The Microsoft IPSec/L2TP client starts an IKE (Internet Key Exchange) negotiation with the peer gateway in order to initiate construction of an encrypted tunnel.
3. During IKE negotiation, the identities of the remote client machine and the security gateway are authenticated. This authentication is performed by means of certificates. Both sides send their certificates to each other as means of proving their identity. This ensures that a connection can be made only from the authenticated machine.
4. Both peers exchange encryption keys, and the IKE negotiation ends.
5. Encryption is now established between the client and the gateway. All connections between the client and the gateway are encrypted inside this VPN tunnel, using the IPSec standard.

6. The Client starts a short L2TP negotiation, at the end of which the client can pass to the gateway L2TP frames that are IPSec encrypted and encapsulated.
7. The security gateway now authenticates the user at the Microsoft IPSec/L2TP client. This authentication is in addition to the client machine authentication in step 3. This identification can happen via two methods.
   - A Certificate
   - An MD5 challenge, whereby the user is asked to enter a username and a password (pre-shared secret)
8. The security gateway allocates to the remote client an Office Mode IP address to make the client routable to the internal network. The address can be allocated from all of the Office Mode methods.
9. The Microsoft IPSec/L2TP client connects to the gateway, and can browse and connect to locations in the internal network.

Behavior of an L2TP Connection

When using an IPSec/L2TP client, it is not possible to connect to organization and to the outside world at the same time.

This is because when the client is connected to the gateway, all traffic that leaves the client is sent to the gateway, and is encrypted, whether or not it is intended to reach the protected network behind the gateway. The gateway then drops all encrypted traffic that is not destined for the encryption domain of the gateway.

security Gateway Requirements for IPSec/L2TP

In order to use Microsoft IPSec/L2TP clients, the security gateway must be set up for remote access. The setup is very similar to that required for remote access using SecuRemote/SecureClient, and involves creating a Remote Access community that includes the security gateway(s) and the user groups.

An additional requirement is to configure the security gateway to supply addresses to the clients by means of the Office Mode feature.

Authentication of Users and Client Machines

There are two methods used to authenticate an L2TP connection:

- Using Legacy Authentication
- Using certificates

Authentication Methods

L2TP clients can use any of the following Authentication schemes to establish a connection:

- Check Point password
- OS password
- RADIUS
- LDAP
- TACACS

Using a username and password verifies that a user is who they claim to be. All users must be part of the Remote Access community and be configured for Office Mode.

Certificates

During the process of establishing the L2TP connection, two sets of authentication are performed. First, the *client machine* and the *security gateway* authenticate each other's identity using certificates. Then, the *user* at the client machine and the *security gateway* authenticate each other using either certificates or a pre-shared secret.

The Microsoft IPSec/L2TP client keeps separate certificates for IKE authentication of the client machine, and for user authentication.

On the security gateway, if certificates are used for user authentication, then the security gateway can use either the same certificate or different certificates for user authentication and for the IKE authentication.

Certificates for both clients and users can be issued by the same CA or a different CA. The users and the client machines are defined separately as users in SmartDashboard.

Certificates can be issued by:

- The Internal Certificate Authority (ICA) on the Security Management server, *or*

- An OPSEC certified Certificate Authority.

The certificates must use the PKCS#12 format, which is a portable format for storing or transporting private keys and certificates. The certificates are transferred to and stored on the client machine.

Authenticating the Client Machine During IKE

The Microsoft IPSec/L2TP client machine needs a certificate to authenticate itself to the security gateway during IKE negotiation.

It is possible to have only one certificate for all client machines, but you will then not be able to identify the machine that the user logged on from. For example, SmartView Tracker would show “user=bob, machine=generic_laptop” rather than “user=bob, machine=bob_laptop”.

The computer account (we call it the machine account) must use PKI and must be in the RemoteAccess community. It is not affected by the authentication scheme in the Remote Access tab in the GUI. It may or may not be a good idea to use the same certificate (and "machine" user) for all clients. You can use an internal CA certificate with no problem for this user. It makes no difference if the authentication tab is defined or not.

The user account is more important, because that is the basis for rule matches and logs. This may use either MD5-challenge (passwords) or certificates. If you choose MD5-challenge, the certificate selection in the remote access tab is irrelevant. As for the user definition, it makes no difference how, if at all, the authentication tab is defined. The password is always the shared secret defined in the encryption tab. Note that this behaviour differs from that of Secure Client, where passwords in the authentication tab override shared secrets from the encryption tab.

The client machine administrator must install the certificate in the machine certificate store.

Authenticating the User

Connecting with Microsoft IPSec/L2TP clients requires that every user be authenticated. Users can be authenticated with:

- certificates, *or*
- using an MD5 challenge, whereby the user is asked to enter a username and a password (pre-shared secret). The user must be informed of the password "out-of-band"

The user certificate can be easily added to the user certificate store. If the user certificate is on a Smart Card, plugging it into the client machine will automatically place the certificate into the certificate store.

User Certificate Purposes

It is possible to make sure that PKI certificates are used only for a defined *purpose*. A certificate can have one or more purposes, such as “client authentication”, “server authentication”, “IPSec” and “email signing”. Purposes appear in the *Extended Key Usage extension* in the certificate.

The certificates used for IKE authentication do not need any purposes. For the user authentication, the Microsoft IPSec/L2TP client requires that

- The user certificate must have the “client authentication” purpose.
- The gateway certificate must have the “server authentication” purpose.

Most CAs (including the ICA) do not specify such purposes by default. This means that the CA that issues certificates for IPSec/L2TP clients must be configured to issue certificates with the appropriate purposes (in the Extended Key Usage extension).

It is possible to configure the ICA on the Security Management server so that the certificates it issues will have these purposes. For OPSEC certified CAs, it is possible to configure the Security Management server to create a certificate request that includes purposes (in the Extended Key Usage extension).

It is also possible to configure the Microsoft IPSec/L2TP clients so that they do not validate the gateway’s certificate during the L2TP negotiation. This is not a security problem because the client has already verified the gateway certificate during IKE negotiation.

Considerations for Choosing Microsoft IPSec/L2TP Clients

SecureClient is much more than a personal firewall. It is a complete desktop security solution that allows the administrator to define a full desktop security policy for the client. IPSec clients are more basic remote clients, and for some organizations may provide an adequate set of capabilities.

When using an IPSec/L2TP client, it is not possible to connect to organization and to the outside world at the same time. For some organizations, this may be an appropriate connection policy as it effectively dedicates the machine to being connected to the organization. SecuRemote/SecureClient on the other hand, makes it possible to be connected to the organization and to the Internet at the same time.

Configuring Remote Access for Microsoft IPSec/L2TP Clients

In This Section

General Configuration Procedure

Establishing a Remote Access VPN for Microsoft IPSec/L2TP clients requires configuration to be performed both on the security gateway and on the client machine. The configuration is the same as setting up Remote Access for SecuRemote/SecureClients, with a few additional steps. It is highly recommended to read and understand Chapter 14, “Introduction to Remote Access VPN”” before configuring Remote Access for Microsoft IPSec/L2TP clients.

The general procedure is as follows:

1. Using SmartDashboard, configure a Remote Access environment, including generating authentication credentials (normally certificates) for the users.
2. Generate certificates to authenticate the client machines.
3. Configure support for Office Mode and L2TP on the security gateway.
4. On the client machine, place the user certificate in the User Certificate Store, and the client machine certificate in the Machine Certificate Store.
5. On the client machine, set up the Microsoft IPSec/L2TP client connection profile.

Configuration details are described in the following sections.

Configuring a Remote Access Environment

1. Follow the instructions in "VPN for Remote Access Configuration" on page 283.

Defining the Client Machines and their Certificates

1. Define a user that corresponds to each client machine, or one user for all machines, and generate a certificate for each client machine user. The steps are the same as those required to define users and their certificate (see "VPN for Remote Access Configuration" on page 283).
2. Add users that correspond to the client machines to a user group, and add the user group to the Remote Access VPN community.

Configuring Office Mode and L2TP Support

1. Configure Office Mode. For detailed instructions, see "Configuring Office Mode" on page 314.
2. In the gateway object, **Remote Access** page, check **Support L2TP**.
3. Select the **Authentication Method** for the users:
   - To use certificates, choose **Smart Card or other Certificates (encryption enabled)**.
   - To use a username and a shared secret (password), choose **MD5-challenge**.
4. For **Use this certificate**, select the certificate that the gateway presents in order to authenticate itself to users. This certificate is used if certificates are the chosen **Authentication Method** for users, in step 3.

Preparing the Client Machines

1. In the Windows **Services** window of the client machine, make sure that the **IPSec Policy Agent** is running. It should preferably be set to Automatic.
2. Make sure that no other IPSec Client (such as SecuRemote/SecureClient) is installed on the machine.

Placing the Client Certificate in the Machine Certificate Store

1. Log in to the client machine with administrator permissions.
2. Run the Microsoft Management Console. Click **Start > Run**
3. Type: `MMC`, and press Enter.
4. Select **Console > Add/Remove Snap-In**.
5. In the **Standalone** tab, click **Add**.
6. In the **Add Standalone Snap-in** window, select **Certificates**.
7. In the **Certificates snap-in** window, select **Computer account**.
8. In the **Select Computer** window select the computer (whether local or not) where the new certificates have been saved.
9. Click **Finish** to complete the process and click **Close** to close the **Add/Remove Snap- in** window.
10. The MMC **Console** window is displayed, where a new certificates branch has been added to the Console root.
11. Right-click on the **Personal** entry of the **Certificates** branch and select **All Tasks > Import**. A Certificate Import Wizard is displayed.
12. In the Certificate Import Wizard, browse to the location of the certificate.
13. Enter the certificate file password.
14. In the **Certificate Store** window make sure that the certificate store is selected automatically based on the certificate type.
15. Select **Finish** to complete the Import operation.

Using the MMC, the certificate can be seen in the certificate store for the “Local Computer”.

Placing the User Certificate in the User Certificate Store

1. On the client machine, double-click on the user's certificate icon (the `.p12` file) in the location where it is saved. A Certificate Import Wizard is displayed
2. Enter the password.
3. In the **Certificate Store** window make sure that the certificate store is selected automatically based on the certificate type.
4. Select **Finish** to complete the Import operation.

Using the MMC, the certificate can be seen in the certificate store for the "current user".

Setting up the Microsoft IPSec/L2TP Client Connection Profile

Once the Client machine's certificate and the user's certificate have been properly distributed, set up the L2TP connection profile.

1. In the client machine, right-click on the **My Network Places** icon on the desktop and select **Properties**.
2. In the **Network and Dial-up Connections** window, select **Make New Connection**. The Network Connection Wizard is displayed.
3. In the **Network Connection Type** window: On Windows 2000 machines select **Connect to a private network through the Internet**. On Windows XP machines select **VPN or dial-up**, and in the next window select **VPN**.
4. In the **Destination Address** window, enter the IP address or the resolvable host name of the gateway.
5. In the **Connection Availability** window, make the new connection available **For all users** or **Only for myself**.
6. In the closing window, provide a name for the new connection, for example, *L2TP_connection*.
7. The **Connect** window for the new connection type is displayed.

To complete the L2TP connection configuration, proceed as follows. Note that the order is important:

8. In the **Connect** window, click **Properties**.

9. In the **Networking** tab, select the L2TP server.

10. In the **Security** tab, choose **Advanced > Settings**, and select **Use extensible Authentication protocols** or **Allow these protocols**.

    If you select **Use extensible Authentication protocols**: Choose either **MD5-challenge**, or **Smart Card or other Certificates (encryption enabled)**. Make the same choice as made on the gateway.

    If you select **Allow these protocols**: Choose **Unencrypted password (PAP)**.

    For more information, see "Configuring Office Mode and L2TP Support" on page 512.

11. Click **OK** to save the configured settings and to return to the **Connect** window.

12. In the **Connect** window, enter the user name and password or select a certificate.

Configuring User Certificate Purposes

To configure the CA to Issue Certificates with Purposes

1. If using the ICA, run the ICA Management Tool (for more details about the tool, see the chapter "The Internal Certificate Authority (ICA) and the ICA Management Tool" in the *Security Management Server Administration Guide*), and in the **Configure the CA** page:

    - Change the property **IKE Certificate Extended Key Usage** property to the value 1, to issue gateway certificates with the "server authentication" purpose.
    - Change the property **IKE Certificate Extended Key Usage** to the value 2 to issue user certificates with the "client authentication" purpose.

    If using an OPSEC certified CA to issue certificates, use the **DBedit** command line or the graphical Database Tool to change the value of the global property `cert_req_ext_key_usage` to `1`. This will cause the Security Management server to request a certificate that has purposes (Extended Key Usage extension) in the certificate.

2. Using SmartDashboard, issue a new certificate for the security gateway. (In the **VPN** page, in the **Certificate List** section click **Add**. A new **Certificate Properties** window opens.) Look at the certificate properties and check that the Extended Key Usage Extension appears in the certificate.

3. In the **Remote Access** page of the security gateway object, in the **L2TP Support** section, select the new certificate.

To Configure the Microsoft IPSec/L2TP Clients so they do not Check for the "Server Authentication" Purpose

The following procedure tells the Microsoft IPSec/L2TP Client not to require the "Server Authentication" purpose on the gateway certificate.

1. In the client machine, right-click on the **My Network Places** icon on the desktop and select **Properties**.
2. In the **Network and Dial-up Connections** window, double click the L2TP connection profile.
3. Click **Properties**, and select the **Security** tab.
4. Select **Advanced (custom settings)**, and click **Settings**.
5. In the **Advanced Security Settings** window, under **Logon security**, select **Use Extensible Authentication Protocol (EAP)**, and click **Properties**.
6. In the **Smart Card or other Certificate Properties** window, uncheck **Validate server certificate**, and click **OK**.

Note - The client validates all aspects of the gateway certificate, during IKE authentication, other than the "Server Authentication" purpose.

Making the L2TP Connection

7. Click on **Connect** to make the L2TP connection.
8. To view the IP address assigned to the connection, either view the **Details** tab in the connection **Status** window, or use the `ipconfig /all` command.

For More Information...

For more information about how to configure advanced capabilities for Microsoft IPSec/L2TP clients, see

- "Non-Private Client IP Addresses" on page 580.
- "Enabling IP Address per User" on page 310.
- "Back Connections (Server to Client)" on page 588.

The L2TP protocol is defined in RFC 2661. Encryption of L2TP using IPSec is described in RFC 3193. For information about the L2TP protocol and the Microsoft IPSec/L2TP client, see the Network and Dial Up Connections Help in Windows 2000 and XP.

Chapter **23**

Secure Configuration Verification

In This Chapter

The Need to Verify Remote Client’s Security Status

Network and Firewall administrators can easily control computers inside their organization. In a Microsoft domain based environment, this is done by controlling the user’s privileges through the network domain controller. The administrator can disable hazardous components such as Java and ActiveX controls in browsers, install Anti-Virus checkers and make sure they are running correctly.

In the case of remote users, the administrator’s options are limited, because remote users access the organization from outside the LAN (e.g., across the Internet), and are usually unable to connect to the domain. The administrator cannot control and verify their configuration through the domain controller.

For example, suppose the remote user has ActiveX enabled, and connects to a website containing a malicious ActiveX control which infects his or her computer. When the remote user connects to the organization’s LAN, the LAN becomes vulnerable as well.

Even a properly configured Desktop Security Policy, important as it is, does not afford protection against this type of attack, because the attack does not target a vulnerability in the access control to the user’s machine, but rather takes advantage of the vulnerable configuration of applications on the client.

The Secure Configuration Verification Solution

In This Section

Introducing Secure Configuration Verification

Secure Configuration Verification (SCV) enables the administrator to monitor the configuration of remote computers, to confirm that the configuration complies with the organization's Security Policy, and to block connectivity for machines that do not comply. SCV does not replace the Desktop Security Policy, but complements it. SCV strengthens enterprise security by ensuring SecureClient machines are configured in accordance with the enterprise Security Policy.

SCV is a platform for creating and using SCV checks. SCV checks include sets of conditions that define a securely configured client system, such as the user's browser configuration, the current version of the Anti-Virus software installed on the desktop computer, the proper operation of the personal firewall policy, *etc*. These security checks are performed at pre-defined intervals by SecureClient. Depending on the results of the SCV checks, the security gateway decides whether to allow or block connections from the client to the LAN.

Check Point's SCV solution comes with a number of predefined SCV checks for the operating system and user's browser, and it also allows OPSEC partners, such as Anti-Virus software manufacturers, to add SCV checks for their own products.

How does SCV work?

SCV works in six steps:

1. Installing SCV plugins on the client.
2. Configuring and SCV Policy on the Security Management server.
3. Downloading the SCV Policy to the Client.
4. Verifying the SCV Policy.
5. Runtime SCV checks.
6. Making the organizational Security Policy SCV aware.

Installing SCV Plugins on the Client

SCV checks are performed through special DLLs which check elements of the client's configuration and return the results of these checks. An SCV application registers its SCV DLLs in the system registry.

The first step in configuring SCV is for the administrator to install the applications that provide the SCV checks on the client. During installation, these applications register themselves as SCV plug-ins and write a hash value of their SCV DLLs to prevent tampering.

Configuring an SCV Policy on the Security Management server

An SCV Policy is a set of rules or conditions based on the checks that the SCV plug-ins provide. These conditions define the requested result for each SCV check, and on the basis of the results, the client is classified as securely configured or non-securely configured. For example, an administrator who wishes to disallow a file-sharing application would define a rule in the SCV Policy verifying that the file-sharing application process is not running.

Note - The SCV check described in this example is among the pre-defined SCV checks included with Security Management server (see "Check Point SCV Checks" on page 524). This check must be configured to test for the specific process.

If *all* the SCV tests return the required results, the client is considered to be securely configured. If even one of the SCV tests returns an unexpected result, the client is considered to be non-securely configured.

Downloading the SCV Policy to the Client

When SecureClient downloads its Desktop Policy from the Policy Server, it downloads its SCV Policy at the same time.

Verifying the SCV Policy

After downloading the SCV Policy, SecureClient confirms that the SCV DLL's specified in the SCV Policy have not been tampered with by calculating their hash values and comparing the results with the hash values specified for the DLLs when they were installed (see *Installing SCV Plugins on the Client*).

Runtime SCV Checks

At regular intervals (default is every 15 seconds), SecureClient performs the SCV checks specified in the SCV Policy by invoking the SCV DLLs, and compares the results to the SCV Policy. The SCV Policy can be configured to display a popup notification on non-securely configured clients and/or send a log to the Security Management server.

Making the Organizational Security Policy SCV-Aware

SecureClient is now able to determine whether the client is securely configured. Once all the organization's clients have been configured according to the previous steps, the administrator specifies the actions to be taken on the security gateway based on the client's SCV status. For example, the administrator can specify that non-securely configured clients cannot access some or all of the resources on the corporate LAN, protecting the organization from the dangers associated with the client's poor security configuration.

The administrator can choose whether to enforce SCV for remote clients. If SCV is enforced, only securely configured clients are allowed access under the rule. If SCV is not enforced, all clients are allowed access under the rule.

In simplified mode, this is configured globally. In traditional mode, this is configured individually for each rule. See "Server Side Configuration" on page 529 for more information.

When the client connects to a security gateway, an IKE negotiation takes place between SecureClient and the gateway. If the gateway's Security Policy requires an SCV check to be made, the gateway holds the connection while it checks if the client is securely configured (SCVed). If the gateway already knows the client's SCV status (i.e., the SCV status was checked in the last 5 minutes), then:

- If the client is securely configured, the gateway allows the connection.

- If the client is not securely configured, the gateway either drops the connection, or accepts and logs it (this behavior is configurable).

If the gateway does not know the client's SCV status, it initiates an SCV check by sending an ICMP unreachable error message containing an SCV query to the client. When a client gets this SCV query, it tries to determine its SCV status. In Connect mode, the client also connects to a Policy Server to download an updated SCV Policy. In parallel, when the client gets the SCV query, it starts sending SCV status replies to the gateway via UDP port 18233 every 20 seconds for 5 minutes. These replies are used as a keep-alive mechanism, in order to keep the user's connection alive in the gateway's state tables while the client is trying to determine its SCV status. The keep alive packets also allow the user to open subsequent connections in the 5 minute period in which they are sent without a need for further SCV queries. When the client determines its SCV status, it sends an SCV reply containing the status back to the gateway via UDP port 18233. When the gateway receives the SCV status of the user, it decides how to handle the user's connection.

SCV Checks

Check Point SCV Checks

A number of SCV checks are provided as part of the SecureClient installation, including:

- **SC_VER_SCV** — a version check that verifies that the SecureClient version is up to date, according to the administrator's specification.
- **Network Configuration Monitor** — verifies that:
  - the Desktop Policy is enforced by SecureClient on all network interface cards
  - non-IP protocols are not enabled on any interface
- **OS Monitor** — verifies the remote user's Operating System version, Service Pack, and Screen Saver configuration (activation time, password protection, *etc*.).
- **HotFix Monitor** — verifies that operating system security patches are installed, or not installed.
- **Group Monitor** — verifies whether the user had logged on the machine and that the user is a member of certain Domain User Groups specified by the administrator.

- **Process Monitor** — checks whether a specified process is running on the client machine (e.g. that a file sharing application is not running, or that Anti-Virus software is running). Process Monitor may also check whether a process is not running.
- **user_policy_scv** — checks the state of the desktop policy, i.e. whether the user is logged on to a policy server, and whether the desktop policy is recent.
- **Browser Monitor** — verifies the Internet Explorer version and specific IE configuration settings, such as various Java and ActiveX options.
- **Registry Monitor** — verifies that a certain key or value is present in the system registry. RegMonitor may check not only for the existence/exclusion of keys but also their content.
- **ScriptRun** — runs a specified executable on SecureClient machine and tests the return code of the executable (e.g. a script that checks whether a certain file is present and sets a return code accordingly). ScriptRun can run a script which performs additional configuration checks.
- **Anti-Virus Monitor** — detects whether an Anti-Virus program is running and checks its version. Supported Anti-Virus programs: Norton, Trend Office Scan, and McAfee.
- **SCVMonitor** — verifies the version of the SCV product, specifically the versions of the SCV DLLs installed on the client's machine.
- **HWMonitor** — verifies the CPU type, family, and model.

Third Party SCV Checks

SCV checks can be written by third party vendors using Check Point's OPSEC SCV SDK. After these applications are installed, the administrator can use these SCV checks in the SCV Policy.

Additional Script Elements

- **SCVpolicy** — selects SCV checks out of the ones defined in SCVNames (see: "SCVNames" on page 535) that will run on the user's desktop.
- **SCVGlobalParams** — is used to define general SCV parameters.

A network administrator can easily enable a set of specific SCV checks (e.g. only check that the user's SecureClient is enforcing a security policy) or as many SCV checks as required (e.g. all of the above SCV checks). The SCV checks are performed independently by the SCV Dynamic Link Libraries, and SecureClient checks their status through the SCV plugins every 15 seconds, and determines whether the user is securely configured or not. If one or more of the tests fails, the SecureClient is considered to be non-securely configured.

Note - To enforce a specific SCV check, set the parameters of the check in the SCVNames section, and include the name of the check in SCVPolicy.

Considerations regarding SCV

In This Section

Planning the SCV Policy

The file `$FWDIR/conf/local.scv` on the Security Management server contains a sample of a basic SCV policy for checks that are supplied with any SCV installation. You can review this file to help you decide which SCV tests to perform. If you need additional SCV checks for OPSEC products, such as Anti-Virus and Endpoint security SCV checks, visit: http://www.opsec.com.

User Privileges

To implement SCV effectively, it is suggested that you consider not to allow your remote users to have administrative privileges on their desktops. Giving the users administrative privileges can allow them to change system settings and cause SCV tests to fail. A desktop which fails an SCV check is a potential security threat to the organization.

For example, as an administrator you may want to configure the user's browser not to allow him to download Java applets from websites. A normal user will not be able to download these applets, but a user with administrative privileges can override the browser's configuration. A properly defined SCV policy can indicate that the browser's configuration had changed and trigger a proper action on the gateway side. However, if the user is allowed by the gateway to pass to the LAN - either by a wrong configuration of the SCV policy or lack of enforcement of the user's SCV status on the gateway side - then the user's desktop will become a potential security risk to the LAN.

The SCV policy itself is protected. Users can not change the SCV policy definition files they receive, even if they have administrative rights. The SCV policy files supplied to the client are signed before arriving to the client and checked against their signature by SecureClient. If the signatures do not match, the SCV check fails.

Using pre-NG Clients with SCV

Pre-NG clients do not include the SCV mechanism described above. Version 4.1 clients, for example, can perform only a limited set of SCV tests:

- check that one of several predefined Security Policies is installed on all interfaces
- check that IP forwarding is not enabled
- check that non TCP/IP protocols are not installed

Using SCV with these clients is limited to these tests only, and does not offer the complete SCV functionality described above. It is possible to configure the gateway to rely on these tests to determine the SCV status of the client, however this is somewhat misleading, since it doesn't offer an indication about the security level of the user's desktop in the parameters that were not covered by these basic tests.

In general, it is recommended to upgrade earlier clients to the latest versions, in order to benefit from the enhancements added to the SCV mechanism in the new versions.

Configuring SCV

In This Section

Server Side Configuration

1. First you need to configure several general parameters regarding SCV. Open your SmartDashboard and go to **Policy > Global Properties** and select the **Remote Access > Secure Configuration Verification (SCV)** tab. This tab has several options:

    - **Apply Secure Configurations on Simplified Mode -** specifies whether all remote access rules in the simplified policy mode should have the SCV flag turned on.
    - **Upon Verification failure** - specifies the action that should be performed when the client fails one or more SCV checks. The options are to Block the client's connection or to Accept it and send a log about the event.
    - **Basic configuration verification on client's machine** - specifies whether SecureClient should perform SCV checks to determine whether the policy is installed on all network interfaces cards on the client's desktop, and whether only TCP/IP protocols are installed on these interfaces.
    - **Configurations Violation Notification on client's machine** - specifies whether a log record should be saved on the Security Management server machine indicating that a remote user is not SCVed (this is a general indication, without a specification of a certain SCV check the user's desktop had failed).

2. To configure whether earlier clients, prior to NG, are considered SCVed, go to **Policy > Global Properties** and select the **Remote Access > Early Version Compatibility** tab. In this tab you can choose the **Required policy for all desktops** from the list of predefined policies. Select **Client is enforcing required policy** if you wish that earlier clients that do not enforce the selected policy will not be considered SCVed.

3. Close the **Global Properties** screen.

4. If you are using simplified mode (the mode that supports VPN communities), skip this step. If you are using traditional mode, edit your Security Policy Rule base and add SCV checks for your remote access rules (Client Encrypt or Client Auth rules). To enable SCV for a remote access rule, right click on the action tab of the rule and choose **Edit properties > Apply rule Only if Desktop Configuration is Verified**. Close the properties screen by pressing **OK**.
5. Edit the `local.scv` file in the `$FWDIR/conf` directory and configure the SCV policy. For more information, see "SCV Policy Syntax" on page 530 and "The local.scv Sets" on page 534.
6. Install the policy - in the policy install dialog box select the Advanced Security policy for the gateways and the Desktop Security policy for the Policy Servers.

Client Side Configuration

1. If you intend to use an OPSEC SCV application, install the application on the client and enable the application's integration with SCV (see the application's documentation for information on how to do this).
2. Start SecureClient and connect to the gateway to receive the SCV Policy. See: "Desktop Security" for more information.

SCV Policy Syntax

The SCV Policy is configured by the administrator in the text file `$FWDIR/conf/local.scv`. This file can be edited either manually by the administrator using a text editor or using a tool called SCVEditor, available at: http://www.opsec.com. The `local.scv` file is a policy file, containing sets, subsets and expressions.

Note - In general, you can use the pre-defined checks (in the SCVNames section of the `local.scv` file) as templates and list the modified checks in the SCVPolicy section, without writing new SCV subsets.

Sets and Sub-sets

Each set has a certain purpose which was predefined for it. For example, one set can be used to define certain parameters, another could specify certain actions that should take place in a certain event etc. Sets are differentiated by their names and hierarchy in a recursive manner. Each set can have a sub-set, and each sub-set can have a sub-set of its own and so on. Subsets can also contain logical expressions. Sets and sub-sets with more than one sub-sets/conditions are delimited by left and right parentheses **()**, and start with the set/sub-set name. Differentiation between sub-sets/expressions with the same hierarchy is done using the colon **:**. For example:

```
(SetName
   :SubSetName1 (
      :ExpressionName1_1 (5)
      :ExpressionName1_2 (false)
   )
   :SubSetName2 (
      :ExpressionName2_1 (true)
      :SubSetName2_1 (
         :ExpressionName2_1_1 (10)
      )
   )
)
```

In the example above the set named **SetName** has two subsets: **SubSetName1** and **SubSetName2**. **SubSetName1** has two conditions in it (**ExpressionName1_1** and **ExpressionName1_2**). **SubSetName2** has one condition (**ExpressionName2_1)** and one subset (**SubSetName2_1**) in it. **SubSetName2_1** has one condition as well **(ExpressionName2_1_1**).

Expressions

Expressions are evaluated by checking the value of the expression (which corresponds to an SCV check) and comparing it with the value defined for the expression (the value in the parentheses). For example, in the browser monitor SCV check provided with SecureClient, you can specify the following expression:

```
 :browser_major_version (5)
```

This expression checks whether the version of the Internet Explorer browser installed on the client is 5.x. If the (major) version is 5, this expression is evaluated as true, otherwise it is evaluated as false. The name of the expression (e.g. “browser_major_version”) is determined by the SCV application and is supplied by manufacturer.

If several expressions appear one after the other, they are logically ANDed, meaning that only if all expressions are evaluated as true, then the value of all of them taken together is true. Otherwise (if even one of the expressions is false), the value of all of them is false. For example:

```
:browser_major_version (5)
:browser_minor_version (0)
```

These expressions are ANDed. If the version of Internet Explorer is 5 AND the minor version is 0 (i.e. version 5.0), then the result is true, otherwise it is false. If the version of Internet Explorer is, for example, 4.0, then the first expression is false and the second one is true, and the result of both of them is false.

Sometimes, some expressions can influence the way in which others are evaluated. For example:

```
:browser_major_version (5)
:browser_minor_version (0)
:browser_version_operand (">=")
```

These expressions are ANDed, but the third expression influences the way that the first and second ones are evaluated. In the example above, if the version of Internet Explorer is greater than or equal to (“>=”) 5.0, then the result is true, otherwise it is false. If the version of Internet Explorer is, for example, 4.5, then the result is false, if the version is 5.1 or higher then the result is true.

Logical Sections

As mentioned earlier, subsequent expressions are automatically ANDed. However, sometimes it is necessary to perform a logical OR between expressions, instead of logical AND. This is done by using labels:

The **begin_or (orX)** label - this label starts a section containing several expressions. The end of this section is marked by a **end (orX)** label (**X** should be replaced with a number which differentiates between different sections OR sections). All of expressions inside this section are logically ORed, producing a single value for the section. For example:

```
:begin_or(or1)
:browser_major_version (5)
:browser_major_version (6)
:end(or1)
```

This section checks whether the version of Internet Explorer is 5 OR 6 - if it is then the result is true, otherwise it is false.

The **begin_and (andX)** label - this label is similar to the **begin_or (orX)** label, but the expressions inside are evaluated and logically ANDed. The end of this section is marked by a **end (andX)** or the **end (orX)** label. As mentioned earlier, simple subsequent expressions are automatically ANDed. The reason that this label exists is to allow nested ANDed sections inside ORed sections. For example, if an administrator considers old browsers as secure since they do not have a lot of potentially unsafe components, and new browsers as secure, since they contain all the latest security patches, he can define the following SCV rules:

```
:begin_or (or1)
:begin_and (and1)
:browser_major_version (5)
:browser_minor_version (0)
:browser_version_operand (">=")
:end (and1)
:begin_and (and2)
:browser_major_version (3)
:browser_minor_version (0)
:browser_version_operand ("<=")
:end (and2)
:end (or1)
```

In the example above, the first AND section checks whether the version of IE >= 5.0, the second AND section checks whether the version of IE is <=3.0 and the are ORed. The entire example is evaluated as true only if the version of IE is larger than (or equal to) 5.0 OR lower than (or equal to) 3.0.

Expressions and Labels with Special Meanings

There are several expressions and labels which have special meaning:

- **begin_admin (admin)** - this label starts a section defining several actions which are performed only if the client is considered as non-SCVed by previous expressions in the subset (i.e. if previous expressions in the subset have returned a value of false). The end of this section is marked by the **end (admin)** label.
- **send_log (type)** - This expression is used as part of the **begin_admin (admin) - end (admin)** section, and determines whether to send a log to the Security Management server (and the client's diagnostic tool) specifying that the client is not SCVed.

  The word **type** should be replace by the type of log to send, such as **log/alert.** Alert means sending a log to the Security Management server, while log means sending the log to the remote client's diagnostic tool.
- **mismatchmessage ("Message")** - This expression is used as part of the **begin_admin (admin) - end (admin)** section, and specifies that a popup message should be shown on the remote user's desktop, indicating the problem. The text in the inverted commas (**Message**) should be replaced by a meaningful text which should instruct the client about the possible sources of the problem and the action he should perform.

For example:

```
:browser_major_version (5)
:browser_minor_version (0)
:browser_version_operand (">=")
:begin_admin (admin)
:send_log (alert)
:mismatchmessage ("The version of your Internet Explorer browser
is old. For security reasons, users with old browsers are not
allowed to access the local area network of the organization.
Please upgrade your Internet Explorer to version 5.0 or higher. If
you require assistance in upgrading or additional information on
the subject, please contact your network administrator”)
:end (admin)
```

In this example, if the user’s IE browser’s version is lower than 5.0, an alert is sent to the Security Management server machine and a popup message is shown to the user with indication of the problem.

The local.scv Sets

The `local.scv` policy files contains one set called SCVObject. This set must always be present and contains all the subsets which deal with the SCV checks and parameters. Currently SCVObject has 3 subsets:

- **SCVNames** - This section is the main SCV policy definition section, in which all of the SCV checks and actions are defined. This is the definition part of the SCV policy, and doesn’t actually determine the SCV checks that will be performed. In this section sets of tests are defined. Later on, the administrator will choose from these sets those he wants to run on the user’s desktop.
- **SCVPolicy** - This section specifies the names of the SCV checks that should actually be performed on the client’s machine, from the SCV checks defined in **SCVNames**.
- **SCVGlobalParams** - This section contains some global SCV parameters.

SCVNames

In this section the administrator specifies the names and different checks for the SCV products. Here is a general definition of an SCV check subset of SCVNames:

```
: (SCVCheckName1
        :type (plugin)
        :parameters (
          :Expression1 (value)
          :Expression2 (value)
          :begin_admin (admin)
          :send_log (alert)
          :mismatchmessage ("Failure Message")
          :end (admin)
        )
     )
```

The test section begins with the name of the SCV check (SCVCheckName1). SCVCheckName1 defines the name of the set of tests. It is defined in the SCV application and should be provided by the SCV manufacturer. The **type (plugin)** expression specifies that the test is performed by an SCV DLL plugin. The **parameters** subset is where the SCV rules and actions are defined. The **type (plugin)** expression and the **parameters** subset should always be specified when defining a subset of SCV checks (such as SCVCheckName1).

SCVPolicy

This section defines the names of the SCV checks that should be enforced (the names are part of the SCV check names specified in SCVNames). This section’s general structure is:

```
:SCVPolicy (
     :(SCVCheckName1)
     :(SCVCheckName2)
  )
```

Note - there is a space between the colon (:) and the opening brace.

The Difference between SCVNames and SCVPolicy

- The SCVNames section defines the different parameters for the checks.
- The SCVPolicy section states which checks are enforced.

To enforce a specific SCV check:

- Set the check’s parameters in SCVNames.
- Include the name of the check in SCVPolicy.

SCVGlobalParams

This section includes global parameters for SCV.

```
:SCVGlobalParams (
   :enable_status_notifications (true)
   :status_notifications_timeout (10)
   :disconnect_when_not_verified (false)
   :block_connections_on_unverified (false)
   :scv_policy_timeout_hours (24)
   :enforce_ip_forwarding (true)
   :not_verified_script ("myscript.bat")
   :not_verified_script_run_show (true)
   :not_verified_script_run_admin (false)
   :not_verified_script_run_always (false)
   :allow_non_scv_clients (false)
 :block_scv_client_connections (false)
)
```

A Complete Example of a local.scv File

Following is a complete example of a `local.scv` file.

Note that in the following example the internal syntax of some of the SCV subsets differs from the syntax described earlier. SCV policy syntax has evolved in recent versions, while these SCV checks were written using the old syntax. For example, in the `sc_ver_scv` subset, the **begin_admin (admin) - end (admin)** section does not exist. In addition, the mismatchmessage expression which was in this section is replaced with MismatchMessage (using capital letters) expression. The syntax and operation of MismatchMessage is similar to the one specified for mismatchmessage, although it does not appear in a **begin_admin (admin) - end (admin)** section.

Another difference in the `sc_ver_scv` subset compared to the syntax explained above concerns the `EnforceBuild_XX_Operand` and `SecureClient_XX_BuildNumber` expressions. These expressions are not ANDed but rather evaluated automatically in accordance with the operating system the user has. For example, if the user has a Windows 2000 system, only the `EnforceBuild_2K_Operand` and `SecureClient_2K_BuildNumber` expressions are evaluated, and the expressions relating to different operating systems are not.

Some other minor changes from the described syntax appear in the `local.scv` policy file. You can review the changes in the default `local.scv` policy file. In general, you can use the pre-defined checks (in the SCVNames section) as templates and list the modified checks in the SCVPolicy section, without writing new SCV subsets.

Note - To enforce a specific SCV check, set the parameters of the check in the SCVNames section, and include the name of the check in SCVPolicy.

Sample

```
(SCVObject
  :SCVNames (
    : (user_policy_scv
      :type (plugin)
      :parameters (
      )
    )
    : (BrowserMonitor
      :type (plugin)
      :parameters (
        :browser_major_version (5)
        :browser_minor_version (0)
        :browser_version_operand (">=")
        :browser_version_mismatchmassage ("Please upgrade your Internet 
browser.")
        :intranet_download_signed_activex (disable)
        :intranet_run_activex (disable)
        :intranet_download_files (disable)
        :intranet_java_permissions (disable)
        :trusted_download_signed_activex (disable)
        :trusted_run_activex (disable)
        :trusted_download_files (disable)
        :trusted_java_permissions (disable)
        :internet_download_signed_activex (disable)
        :internet_run_activex (disable)
        :internet_download_files (disable)
        :internet_java_permissions (disable)
        :restricted_download_signed_activex (disable)
        :restricted_run_activex (disable)
        :restricted_download_files (disable)
        :restricted_java_permissions (disable)
        :send_log (alert)
      :internet_options_mismatch_message ("Your Internet browser settings do
not meet policy requirements\nPlease check the following settings:\n1. In
your browser, go to Tools -> Internet Options -> Security.\n2. For each Web
content zone, select custom level and disable the following items: DownLoad
signed ActiveX, Run ActiveX Controls, Download Files and Java Permissions.")
    )
   )
   : (OsMonitor
    :type (plugin)
    :parameters (
      :os_version_mismatchmessage ("Please upgrade your operating system.")
      :enforce_screen_saver_minutes_to_activate (3)
      :screen_saver_mismatchmessage ("Your screen saver settings do not meet
policy requirements\nPlease check the following settings:\n1. Right click on
your desktop and select properties.\n2. Select the Screen Saver tab.\n3.
Under Wait choose 3 minutes and check the Password Protection box.")
      :send_log (log)
      :major_os_version_number_9x (4)
      :minor_os_version_number_9x (10)
      :os_version_operand_9x (">=")
      :service_pack_major_version_number_9x (0)
      :service_pack_minor_version_number_9x (0)
      :service_pack_version_operand_9x (">=")
      :major_os_version_number_nt (4)
      :minor_os_version_number_nt (0)
      :os_version_operand_nt ("==")
      :service_pack_major_version_number_nt (5)
      :service_pack_minor_version_number_nt (0)
      :service_pack_version_operand_nt (">=")
      :major_os_version_number_2k (5)
      :minor_os_version_number_2k (0)
      :os_version_operand_2k ("==")
      :service_pack_major_version_number_2k (0)
      :service_pack_minor_version_number_2k (0)
      :service_pack_version_operand_2k (">=")
      :major_os_version_number_xp (5)
      :minor_os_version_number_xp (1)
      :os_version_operand_xp ("==")
      :service_pack_major_version_number_xp (0)
      :service_pack_minor_version_number_xp (0)
      :service_pack_version_operand_xp (">=")
    )
   )
   : (ProcessMonitor
    :type (plugin)
    :parameters (
      :begin_or (or1)
       :AntiVirus1.exe (true)
       :AntiVirus2.exe (true)
      :end (or1)
    :IntrusionMonitor.exe (true)
    :ShareMyFiles.exe (false)
    :begin_admin (admin)
     :send_log (alert)
     :mismatchmessage ("Please check that the following processes are 
running:\n1. AntiVirus1.exe or AntiVirus2.exe\n2. 
IntrusionMonitor.exe\n\nPlease check that the following process is not 
running\n1. ShareMyFiles.exe")
    :end (admin)
   )
  )
  : (groupmonitor
   :type (plugin)
   :parameters (
    :begin_or (or1)
     :begin_and (1)
      :"builtin\administrator" (false)
      :"BUILTIN\Users" (true)
     :end (1)
     :begin_and (2)
      :"builtin\administrator" (true)
      :"BUILTIN\Users" (false)
     :end (and2)
     :end (or1)
    :begin_admin (admin)
     :send_log (alert)
     :mismatchmessage ("You are using SecureClient with a non-authorized 
user.\nMake sure you are logged on as an authorized user.")
     :securely_configured_no_active_user (false)
    :end (admin)
   )
  )
  : (HotFixMonitor
   :type (plugin)
   :parameters (
    :147222 (true)
    :begin_admin (admin)
     :send_log (alert)
     :mismatchmessage ("Please install security patch Q147222.")
    :end (admin)
   )
  )
  : (AntiVirusMonitor
   :type (plugin)
   :parameters (
    :type ("Norton")
    :Signature (">=20020819")
    :begin_admin (admin)
     :send_log (alert)
```

```
          :mismatchmessage ("Please update your AntiVirus (use the LiveUpdate
option).")
        :end (admin)
      )
    )
    : (HWMonitor
      :type (plugin)
      :parameters (
        :cputype ("GenuineIntel")
        :cpumodel ("9")
        :cpufamily ("6")
        :begin_admin (admin)
          :send_log (alert)
          :mismatchmessage ("Your machine must have an\nIntel(R) Centrino(TM)
processor installed.")
        :end (admin)
      )
    )
    : (ScriptRun
      :type (plugin)
      :parameters (
        :exe ("VerifyScript.bat")
        :begin_admin (admin)
          :send_log (alert)
          :mismatchmessage ("Verification script has determined that your
configuration does not meet policy requirements.")
        :end (admin)
      )
    )
    : (RegMonitor
      :type (plugin)
      :parameters (
        :value
("Software\TrendMicro\PC-cillinNTCorp\CurrentVersion\Misc.\PatternVer>=414
")
        :begin_admin (admin)
          :send_log (alert)
          :mismatchmessage ("Please update your AntiVirus (use the LiveUpdate
option).")
        :end (admin)
      )
    )
    : (SCVMonitor
      :type (plugin)
      :parameters (
        :scv_version ("54014")
        :begin_admin (admin)
          :send_log (alert)
          :mismatchmessage ("Please upgrade your Secure Configuration
Verification products package.")
```

```
        :end (admin)
      )
    )
    : (sc_ver_scv
      :type (plugin)
      :parameters (
        :Default_SecureClientBuildNumber (52032)
        :Default_EnforceBuildOperand ("==")
        :MismatchMessage ("Please upgrade your SecureClient.")
        :EnforceBuild_9X_Operand (">=")
        :SecureClient_9X_BuildNumber (52030)
        :EnforceBuild_NT_Operand ("==")
        :SecureClient_NT_BuildNumber (52032)
        :EnforceBuild_2K_Operand (">=")
        :SecureClient_2K_BuildNumber (52032)
        :EnforceBuild_XP_Operand (">=")
        :SecureClient_XP_BuildNumber (52032)
      )
    )
  )
  :SCVPolicy (
    : (BrowserMonitor)
    : (HWMonitor)
    : (AntiVirusMonitor)
  )
  :SCVGlobalParams (
    :enable_status_notifications (false)
    :status_notifications_timeout (10)
    :disconnect_when_not_verified (false)
    :block_connections_on_unverified (false)
    :scv_policy_timeout_hours (24)
    :enforce_ip_forwarding (true)
    :not_verified_script ("")
    :not_verified_script_run_show (false)
    :not_verified_script_run_admin (false)
    :not_verified_script_run_always (false)
    :not_verified_script_run_always (false)
    :allow_non_scv_clients (false)
)
```

When using this file, it is important to maintain the same indentation/nesting format.

Common Attributes

Typically, an administrator might need to change only a few of the common parameters (SCV checks) contained in the SCV policy file.

SCV Checks

1. **Anti-Virus monitor**

   Parameters:

   - `Type ("av_type")`

     Type of Anti-Virus. For example, "Norton", "VirusScan", "OfficeScan", or "ZoneLabs".

   - `Signature(x)`

     Required Virus definition file signature. The signature's format depends on the AntiVirus type. For example, on Norton Antivirus the signature maybe be ">=20031020". (The format for Norton's AV signature is "yyyymmdd").

     For TrendMicro Officescan, the signature maybe "<650"

     For McAfee's VirusScan, use signature (">404291") for a signature greater than 4.0.4291

     For Zone Labs, use signature (">X.Y.Z") where X = Major Version, Y = Minor Version, and Z = Build Number of the `.dat` signature file.

AntiVirusMonitor does not support "begin_or" and the "begin_and" syntax. See: "Expressions and Labels with Special Meanings".

2. **BrowserMonitor**

   Parameters:

   - `browser_major_version (5)`

     Major version number of Internet Explorer. If this field does not exist in the `local.scv` file, or if this value is 0, the IE'S version will not be checked as part of the BrowserMonitor check.

   - `browser_minor_version (0)`

     Internet Explorer's minor version number.

   - `browser_version_operand (">=")`

     The operator used for checking the Internet Explorer's version number.

- `browser_version_mismatchmessage ("Please upgrade your Internet Browser.")`

  Message to be displayed in case of a non-verified configuration for the Internet Explorer's version.

- `intranet_download_signed_activex (enable)`

  The maximum permission level that IE should have for downloading signed ActiveX controls from within the local Intranet.

- `intranet_run_activex (enable)`

  The maximum permission level that IE should have for running signed ActiveX controls from within the local Intranet.

- `intranet_download_files (enable)`

  The maximum permission level that IE should have for downloading files from within the local Intranet.

- `intranet_java_permissions (low)`

  The maximum security level that IE Explorer should have for running java applets from within the local Intranet.

  (low) means a low security level.

- `trusted_download_signed_activex (enable)`

  The maximum permission level that IE should have for downloading signed ActiveX controls from trusted zones.

- `trusted_run_activex (enable)`

  The maximum permission level that IE should have for running signed ActiveX controls from trusted zones.

- `trusted_download_files (enable)`

  The maximum permission level that IE should have for downloading files from trusted zones.

- `trusted_java_permissions (medium)`

  The maximum security level that IE should have for running java applets from trusted zones.

- `internet_download_signed_activex (disable)`

  The maximum permission level that IE should have for downloading signed ActiveX controls from the Internet.

- `Internet_run_activex (disable)`

The maximum permission level that IE should have for running signed ActiveX controls from the Internet.

- `internet_download_files (disable)`

  The maximum permission level that IE should have for downloading files from the Internet.

- `internet_java_permissions (disable)`

  The maximum security level that IE should have for running java applets from the Internet.

- `restricted_download_signed_activex (disable)`

  The maximum permission level that IE should have for downloading signed ActiveX controls from restricted zones.

- `restricted_run_activex (disable)`

  The maximum permission level that IE should have for running signed ActiveX controls from restricted zones.

- `restricted_download_files (disable)`

  The maximum permission level that IE should have for downloading files from restricted zones.

- `restricted_java_permissions (disable)`

  The maximum security level that IE should have for running java applets from restricted zones.

- `send_log (type)`

  Determines whether to send a log to Security Management server for specifying that the client is not “SCVed.”

  This SCV check does not support the “`begin admin/end admin`” parameter section.

  The `(type)` section should be replaced by `(log)` or `(alert)`

- `internet_options_mismach_message (“Your Internet browser settings do not meet policy requirements”)`

  Mismatch message for the Internet Explorer settings.

BrowserMonitor can be configured to check only Internet Explorer’s version, or only the browser’s settings for a certain zone. For example, if none of the following parameters appear:

i. `restricted_download_signed_activex`

ii. `restricted_run_activex`

iii. `restricted_download_files`

iv. `restricted_java_permissions`

then BrowserMonitor will not check the restricted zones' security settings. In similar fashion, if the parameter "`browser_major_version`" does not appear or is equal to zero, then IE's version number is not checked.

BrowserMonitor does not support the "`begin_or`" and the "`begin_and`" syntax, and does not support the admin parameters. See also: "Expressions and Labels with Special Meanings".

For the script for checking Internet Explorer Service Pack, see "Script for Internet Explorer Service Pack" on page 552.

3. **Groupmonitor**

   Parameters

   - `"builtin\administrator" (false)`

     A name of a user group. The user has to belong to this group in order for the machine configuration to be verified.

   - `securely_configured_no_active_user (true)`

     Specifies whether the machine's configuration may be considered verified when no user is logged on. The default value is false.

4. **HotFixMonitor**

   Parameters

   - `HotFix_Number (true)`

     A number of a system HotFix to be checked. In order for the machine to be verified, the HotFix should be installed, for example: "`823980(true)`" verifies that Microsoft's RPC patch is installed on the operating system.

   - `HotFix_Name (true)`

     The full name of a system HotFix to be checked. In order for the machine to be verified, the HotFix should be installed, for example: "`KB823980(true)`" verifies that Microsoft's RPC patch is installed on the operating system.

Not all the mentioned fields for HotFixMonitor need to appear in the `local.scv` file. Some of them may not appear at all, or may appear more than once. These fields may also be ORed and ANDed. In this way, multiple HotFixes can be checked, and the results ORed or ANDed for extra flexibility.

5. **HWMonitor**

   Parameters

   - `cputype ("GenuineIntel")`

     The CPU type as described in the vendor ID string. The string has to be exactly 12 characters long. For example: `"GenuineIntel"`, or `"AuthenticAMD"`, or `"aaa bbb ccc "` where spaces count as a character.

   - `cpufamily(6)`

     The CPU family.

   - `cpumodel(9)`

     The CPU model.

HWMonitor does not support the "`begin_or`" and the "`begin_and`" syntax. See also: "Expressions and Labels with Special Meanings".

6. **OsMonitor**

   Parameters

   - `enforce_screen_saver_minutes_to_activate (3)`

     Time in minutes for the screen saver to activate. If the screen saver does not activate within this time period, then the client is not considered verified. In addition, the screen saver must be password protected.

   - `screen_saver_mismatchmessage ("Your screen saver settings do not meet policy requirements")`

     Mismatch message for the screen saver check. The screen saver will not be checked if the property "`enforce_screen_saver_minutes_to_activate`" does not appear, or if the time is set to zero.

   - `send_log (type)`

     Determines whether to send a log to Security Management server for specifying that the client is not "SCVed."

     This SCV check does not support the `"begin admin/end admin"` parameter section.

     The `(type)` section should be replaced by `(log)` or `(alert)`

   - `major_os_version_number_9x (4)`

     Specifies the major version required for 9x operating systems to be verified.

   - `minor_os_version_number_9x (10)`

Specifies the minor version required for 9x operating systems to be verified.

- `os_version_operand_9x ("">="")`

  Operator for checking the operating system's version on 9x.

- `service_pack_major_version_number_9x (0)`

  Specifies the major service pack's version required for 9x operating system's to be verified.

- `service_pack_minor_version_number_9x (0)`

  Specifies the minor service pack's version required for 9x operating systems to be verified.

- `service_pack_version_operand_9x ("">="")`

  Operator for checking the operating system's service pack on 9x.

- `major_os_version_number_nt (4)`

  Specifies the major version required for Windows NT operating systems to be verified.

- `minor_os_version_number_nt (10)`

  Specifies the minor version required for Windows NT operating systems to be verified.

- `os_version_operand_nt ("">="")`

  Operator for checking the operating system's version on Windows NT.

- `service_pack_major_version_number_nt (0)`

  Major service pack version required for Windows NT operating systems to be verified

- `service_pack_minor_version_number_nt (0)`

  Minor service pack version required for Windows NT operating systems to be verified

- `service_pack_version_operand_nt ("">="")`

  Operator for checking the operating system's service pack on Windows NT

- `major_os_version_number_2k (4)`

  Specifies the major version required for Windows 2000 operating systems to be verified.

- `minor_os_version_number_2k (10)`

Specifies the minor version required for Windows 2000 operating systems to be verified.

- `os_version_operand_2k ("">=")`

  Operator for checking the operating system's version on Windows 2000

- `service_pack_major_version_number_2k (0)`

  Specifies major service pack version required for Windows 2000 operating systems to be verified.

- `service_pack_minor_version_number_2k (0)`

  Specifies minor service pack version required for Windows 2000 operating systems to be verified.

- `service_pack_version_operand_2k (">=")`

  Operator for checking the operating system's service pack on Windows 2000

- `major_os_version_number_xp (4)`

  Specifies the major version required for Windows XP operating systems to be verified.

- `minor_os_version_number_xp (10)`

  Specifies the minor version required for Windows XP operating systems to be verified.

- `os_version_operand_xp (">=")`

  Operator for checking the operating system's service pack on Windows XP

- `service_pack_major_version_number_xp (0)`

  Specifies the major service pack version required for Windows XP operating systems to be verified.

- `service_pack_minor_version_number_xp (0)`

  Specifies the minor service pack version required for Windows XP operating systems to be verified.

- `service_pack_version_operand_xp (">=")`

  Operator for checking the operating system's service pack on Windows XP.

- `os_version_mismatches ("Please upgrade your operating system")`

Message to be displayed in case of a non-verified configuration for the operating system's version/service pack. The operating system's version and service pack will not be checked if none of the parameters appear in the scv file.

- :major_os_version_number_2003 (5)

  Specifies the major version required for Windows 2003 operating systems to be verified.

- :minor_os_version_number_2003 (2)

  Specifies the minor version required for Windows 2003 operating systems to be verified.

- :os_version_operand_2003 ("==")

  Operator for checking the operating system's service pack on Windows 2003

- :service_pack_major_version_number_2003 (0)

  Specifies the major service pack version required for Windows 2003 operating systems to be verified.

- :service_pack_minor_version_number_2003 (0)

  Specifies the minor service pack version required for Windows 2003 operating systems to be verified.

- :service_pack_version_operand_2003 (">=")

  Operator for checking the operating system's service pack on Windows 2003

OsMonitor can be configured to check only the screen saver's configuration, or only the operating system's version and service pack. For example, if none of the following parameters appear:

i. `major_os_version_number_xp`

ii. `minor_os_version_number_xp`

iii. `os_version_operand_xp`

iv. `service_pack_major_version_number_xp`

v. `service_pack_minor_version_number_xp`

vi. `service_pack_version_operand_xp`

then OsMonitor will not check the system's version and service pack on Windows XP platforms.

In similar fashion:

if the parameter "`enforce_screen_saver_minutes_to_activate`" does not appear, then the screen saver's configuration is not checked.

OSMonitor does not support the "`begin_or`" and the "`begin_and`" syntax. See also: "Expressions and Labels with Special Meanings".

7. **ProcessMonitor**

   Parameters

   - `ProcessName.exe (true)`

     A process the administrator would like to check. If the value is true, the process needs to be running for the machine to be verified. If the value is false, the process should *not* be running for the machine to be verified.

     ProcessMonitor can also be used to check for the existence/exclusion of more than one process. The fields may be ANDed or ORed for flexibility.

8. **RegMonitor**

   - PredefinedKeys (HIVE)

     Specify the registry hive from one of the following choices:

     - HKEY_CURRENT_USER
     - HKEY_LOCAL_MACHINE
     - HKEY_USERS

     If one of the hives is not specified, then HKEY_LOCAL_MACHINE is used.

     To configure a check for HKEY_CLASSES_ROOT, use HKEY_LOCAL_MACHINE\Software\Classes and HKEY_CURRENT_USER\Software\Classes.

   - `value (registry_value_path)`

     The path of a registry DWORD, under the hive specified by the predefined keys will be checked. The value should be an operator followed by a number, e.g. "`Software\TrendMicro\PC-cillinNTCorp\CurrentVersion\Misc.\PatternVer>=414`"

     The syntax for the value parameter is:

     ```
     :value ("pathOPval")
     ```

     For example:

     ```
     :value ("Software\...\PaternVer>=414")
     ```

- string (registry_string_path)

  The path of a registry string, under the hive specified by the predefined keys will be checked. The string's value is compared to the given value, in the way that DWORDs are compared.

- keyexist (registry_key_path)

  The path of a registry key to check if the key exists, under the hive specified by the predefined keys will be checked. The key must exist if the machine is to be verified.

- keynexist (registry_key_path)

  The path of a registry key to be checked for exclusion, under the hive specified by the predefined keys will be checked. For the machine to be verified, the key should not exist.

- allow_no_user (default: **true**)

  This parameter is valid only when a user is logged in to the machine.

  Since SC services and SCV checks run also when no user is logged on, a decision should be taken if the check passed or failed.

  If no user is logged on to the machine, and a running RegMonitor check is configured to monitor HKEY_CURRENT_USER, the behavior is according to the flag allow_no_user.

  If allow_no_user is true, the check will PASS.

  If allow_no_user is false, the check will FAIL.

  This attribute is not, by default, included in the local.scv file. If the attribute does not exist in the file, then the default setting used is also true.

  Configuring this attribute is done via local.scv. For example:

```
: (RegMonitor
         :type (plugin)
         :parameters (
              :keyexist ("HKEY_CURRENT_USER\Software\CheckPoint")
                 :allow_no_user (true)
                 :begin_admin (admin)
                         :send_log (alert)
                         :mismatchmessage ("mismatch message ")
                 :end (admin)
         )
 )
```

Not all the mentioned fields for RegMonitor need to appear in the `local.scv` file. Some of them may not appear at all, or may appear more than once. These fields may also be ORed and ANDed. In this way, multiple registry entries can be checked, and the results ORed or ANDed for extra flexibility.

Script for Internet Explorer Service Pack

RegMonitor can be configured to check the version and service pack of Internet Explorer. The script looks as follows:

```
: (RegMonitor
            :type (plugin)
            :parameters (
                        :begin_or (or1)
                                    :keynexist
("Software\Microsoft\Internet Explorer")
                            :string ("Software\Microsoft\Internet
Explorer\Version>=6")
                                    :begin_and (and1)
                                                :string
("Software\Microsoft\Internet Explorer\Version>=5.5")
:string ("Software\Microsoft\Windows\CurrentVersion\Internet
Settings\MinorVersion>=SP2")
                                                :string
("Software\Microsoft\Windows\CurrentVersion\Internet
Settings\MinorVersion<=SP9")
                                    :end_and (and1)
                                    :begin_and (and2)
                                                :string
("Software\Microsoft\Internet Explorer\Version>=5.5")
                                                :string
("Software\Microsoft\Windows\CurrentVersion\Internet
Settings\MinorVersion>=;SP2")
                                                :string
("Software\Microsoft\Windows\CurrentVersion\Internet
Settings\MinorVersion<=;SP9")
                                    :end_and (and2)
:end_or (or1)
                        :begin_admin (admin)
                                    :send_log (alert)
                                 :mismatchmessage ("Your IE must
be at least version 5.5 with SP2.")
                        :end (admin)
            )
)
```

9. **SCVMonitor**

   Parameters

   - `scv_version(“>=541000076”)`

Represents the SCV product's build number. This is the version of the DLLs in charge of the SCV checks. This number differs from the build number of SecureClient. SCV products can be upgraded, and maybe updated without updating SecureClient.

The string is an operator followed by the DLL's version number in the format "vvshhhbbb". For example, if you want the DLL version to be at least 54.1.0.220, the syntax should be: `scv_version ("">=541000220")`

SCVMonitor does not support the "`begin_or`" and the "`begin_and`" syntax. See also: "Expressions and Labels with Special Meanings".

10. **ScriptRun**

Parameters

- `exe ("VerifyScript.bat")`

  Runs an executable. Supply the name of the executable, and the full path to the executable.

- `run_as_admin ("no")`

  Determines whether the verification script is run with administrator privileges. The default is "no". The only other value is "yes".

- `run_timeout (10)`

  Time (in seconds) to wait for the executable to finish. If the executable does not finish within the set time, the process is considered as a failure, and the machine categorized as "not verified". The default value is zero, which is the same as "no timeout".

ScriptRun does not support the "`begin_or`" and the "`begin_and`" syntax. See also: "Expressions and Labels with Special Meanings".

11. **sc_ver_scv**

Parameters

- `Default_SecureClientBuildNumber (52032)`

  Build number for SecureClient. This build number is checked (with the specified operator) only if no specific build number is to be checked for a particular platform.

- `Default_EnforceBuildOperand ("==")`

  Operator for comparing the `local.scv`'s build number with the client build number.

- `MismatchMessage ("Please upgrade your SecureClient")`

Mismatch message to be displayed when the SecureClient's build does not match the local.scv's configuration.

- `EnforceBuild_9x_Operand ("˃=")`

  Operator for comparing the local.scv's build number with the client build number on Windows 9x platforms.

- `SecureClient_9x_BuildNumber (52030)`

  SecureClient build number for windows 9x platforms.

- `EnforceBuild_NT_Operand ("==")`

  Operator for comparing the local.scv's build number with the client build number on WindowsNT platforms.

- `SecureClient_NT_BuildNumber (52030)`

  SecureClient build number for WindowsNT platforms.

- `EnforceBuild_2K_Operand ("˃=")`

  Operator for comparing the local.scv's build number with the client build number on Window 2000 platforms.

- `SecureClient_2K_BuildNumer (52030)`

  SecureClient build number for Windows 2000 platforms.

- `EnforceBuild_XP_Operand ("˃=")`

  Operator for comparing the local.scv's build number with the client build number on Windows XP platforms.

- `SecureClient_XP_Buildnumber (52030)`

  SecureClient build number for Windows XP platforms.

sc_ver_scv does not support the "`begin_or`" and the "`begin_and`" syntax. See also: "Expressions and Labels with Special Meanings".

12. **user_policy_scv**

Parameters

- `logged_on_to_policy_server (true/false)`

  Specifies whether the user has to be logged on to a Policy Server to be considered SCVed.

- `policy_refresh_rate ("168")`

Time, in hours, for which the desktop policy remains valid. After 168 hours the desktop policy is not considered valid, and the user is no longer SCVed. If this parameter is not specified, the policy is not checked for freshness.

- `mismatchmessage ("Place a message here")`

  The message displayed when the `user_policy_scv` check fails.

- `dont_enforce_while_connecting`

  If this parameter is present, the user is considered SCVed while connecting to the gateway. The user is considered SCVed only for the duration of the connect process.

13. **SCVGlobalParams**

Parameters

For all boolean parameters (true or false), the values should not be enclosed in quotation marks.

- `enable_status_notifications (true/false)`

  If "true", SecureClient displays a balloon window when the Desktop is not SCVed. On windows 9x and NT, where balloons are not supported, popups appear.

- `status_notifications_timeout ()`

  The number of seconds the balloon window (see previous parameter) will be displayed.

- `disconnect_when_not_verified (true/false)`

  If "true", SecureClient will disconnect from the site when the Desktop is not SCVed.

- `block_connections_on_unverified (true/false)`

  If "true", SecureClient will drop all open connections when the Desktop is not SCVed.

Note - This parameter, if true, blocks all connections to the machine, not just those connections to and from the VPN site.

- `scv_policy_timeout_hours ()`

  The period (in hours) during which the SCV policy is considered valid since the last logon to the Policy Server. When this timeout is about to expire SecureClient will attempt to logon to the Policy Server to get a new SCV policy.

Possible values are between 1 and 504 hours(21 days). The default value is 168 hours (one week). If you set the value to 0, the SCV policy never expires (no time-out).

- `enforce_ip_forwarding (true/false)`

  If "true" the IP Forwarding between network interface cards on the user's desktop must be disabled for the user to be considered SCVed.

- `ip_forwarding_mismatchmessage ("Message string placed here")`

  The value is a string displayed when ip forwarding is enabled. For example: `ip_forwarding_mismatchmessage ("Please....etc")`

  This is relevant only if ip forwarding is part of the SCV checks, that is, if the parameter is defined as True.

- `not_verified_script ("script_name.bat")`

  The name of executable that will be run when the Desktop is not SCVed. The next three parameters provide more options related to the running of the executable.

- `not_verified_script_run_show (true/false)`

  If "true", the executable's progress will be displayed in an onscreen window.

- `not_verified_script_run_admin (true/false)`

  If "true", the executable will run with administrator privileges.

- `not_verified_script_run_always (true/false)`

  If "true", the executable will run every time the Desktop is not SCVed. If "false", it will run once per SecureClient session.

- `:allow_non_scv_clients (true/false)`

  If "true", the client will send a verified state to the enforcing gateway even if the OS does not support SCV.

Chapter 24

VPN Routing - Remote Access

In This Chapter

The Need for VPN Routing

There are a number of scenarios in which a gateway or remote access clients cannot connect directly to another gateway (or clients). Sometimes, a given gateway or client is incapable of supplying the required level of security. For example:

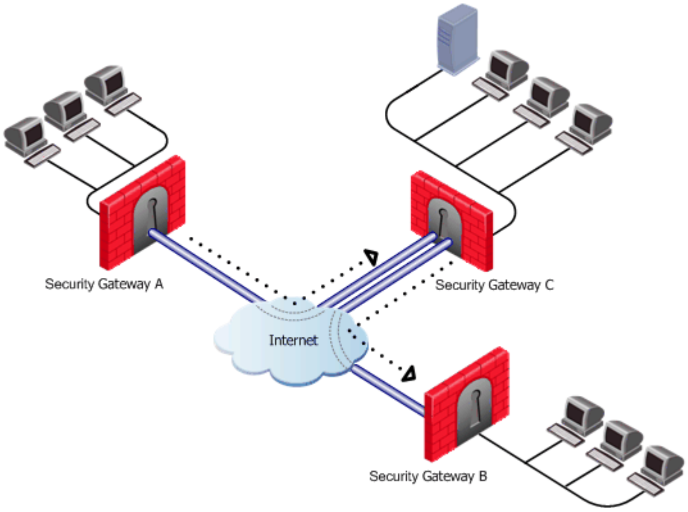
- Two gateways with *dynamically assigned IP addresses* (DAIP gateways). Hosts behind either gateway need to communicate; however, the changing nature of the IP addresses means the two DAIP gateways cannot open VPN tunnels. At the moment of tunnel creation, the exact IP address of the other is unknown.
- SecuRemote/SecureClient users wish to have a private conversation using *Voice-over-IP* (VoIP) software or utilize other client-to-client communication software such as Microsoft NetMeeting. Remote access clients cannot open connections directly with each other, only with configured gateways.

In all cases, a method is needed to enhance connectivity and security.

Check Point Solution for Greater Connectivity and Security

VPN routing provides a way of controlling how VPN traffic is directed. VPN routing can be implemented with gateway modules and remote access clients. Configuration for VPN routing is performed either directly through SmartDashboard (in simple cases) or by editing the VPN routing configuration files on the gateways (in more complex scenarios).

Figure 24-1 Simple VPN routing



In Figure 24-1, one of the host machines behind gateway A needs to connect with a host machine behind gateway B. For either technical or policy reasons, gateway A cannot open a VPN tunnel with gateway B. However, both gateways A and B can open VPN tunnels with gateway C, so the connection is routed through gateway C.

As well as providing enhanced connectivity and security, VPN routing can ease network management by hiding a complex network of gateways behind a single Hub.

Hub Mode (VPN Routing for Remote Clients)

VPN routing for remote access clients is enabled via Hub Mode. In Hub mode, all traffic is directed through a central Hub. The central Hub acts as a kind of router for the remote client. Once traffic from remote access clients is directed through a Hub, connectivity with other clients is possible as well as the ability to inspect the subsequent traffic for content.

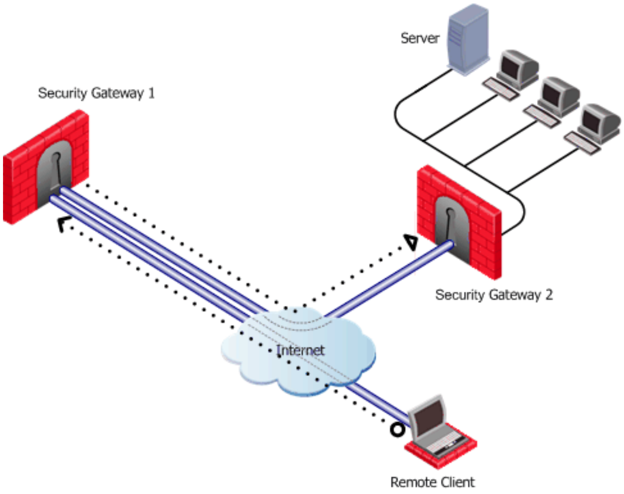
When using Hub mode, enable Office mode. If the remote client is using an IP address supplied by an ISP, this address might not be fully routable. When Office mode is used, rules can be created that relate directly to Office mode connections.

Note - Office mode is only supported in SecureClient, not SecuRemote.

Allowing SecureClient to Route all Traffic Through a Gateway

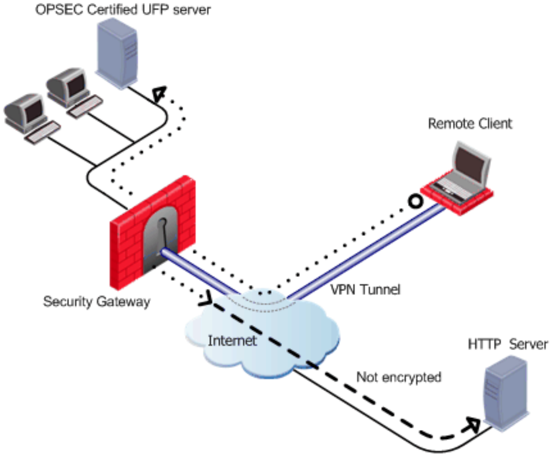
In Figure 24-2, the remote client needs to connect with a server behind security gateway 2. Company policy states that all connections to this server must be inspected for content. For whatever reason, security gateway 2cannot perform the required content inspection. When all the traffic is routed through security gateway 1, connections between the remote client and the server can be inspected.

Figure 24-2 Monitoring traffic to a Remote Client



Suppose the same remote client needs to access an HTTP server on the Internet. The same company policy regarding security still applies.

Figure 24-3 Remote client to Internet



The remote client's traffic is directed to the security gateway where it is directed to the UFP (URL Filtering Protocol) server to check the validity of the URL and packet content, since the gateway does not possess URL-checking functionality. The packets are then forwarded to the HTTP server on the Internet.

NATing the address of the remote client behind the security gateway prevents the HTTP server on the Internet from replying directly to the client. If the remote client's address is not NATed, the remote client will not accept the clear reply from the HTTP server.

Remote Client to Client Communication

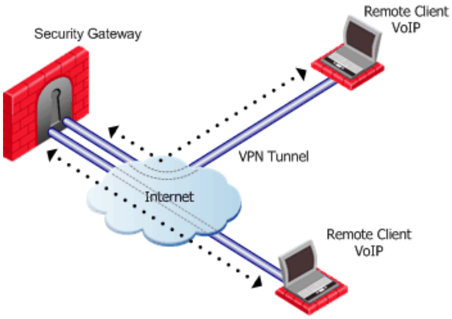
Remote client to client connectivity is achieved in two ways:

- By routing all the traffic through the gateway.
- Including the Office Mode range of addresses in the VPN domain of the gateway.

Routing all Traffic through the Gateway

Two remote users use VoIP software to hold a secure conversation. The traffic between them is directed through a central Hub, as in Figure 24-4:

Figure 24-4 Hub mode for remote access clients

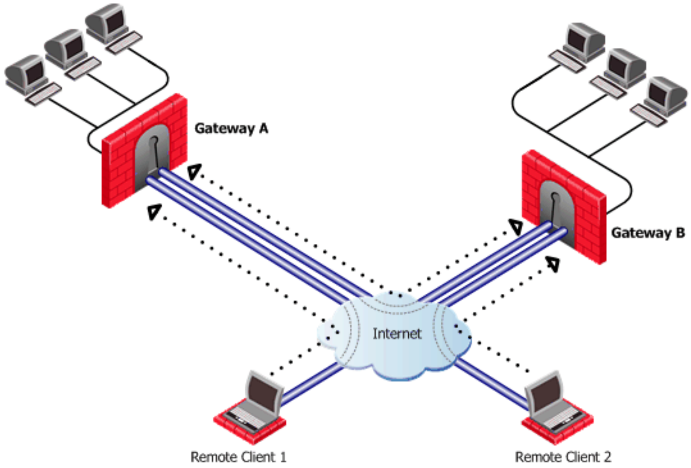


For this to work:

- **Allow SecureClient to route traffic through this Gateway** must be enabled on the gateway.
- The remote client must be configured with a profile that enables all traffic to be routed through the gateway.
- Remote clients are working in connect mode.

If the two remote clients are configured for Hub mode with different gateways, the routing takes place in three stages - each remote client to its designated gateway, then between the gateways:

Figure 24-5 Remote clients to different hubs



In Figure 24-5, remote client 1 is configured for Hub mode with gateway A. Remote client 2 is configured for Hub mode with gateway B. For the connection to be routed correctly:

- Office mode *must* be enabled.
- VPN configuration files on both gateways must include the Office Mode address range used by the other. In Figure 24-5, the VPN configuration file on gateway A directs all traffic aimed at an Office Mode IP address of gateway B towards gateway B. A connection leaves Remote Client1 and is sent to gateway A. From gateway A the connection is redirected to gateway B. Gateway B once more redirects the traffic towards Remote Client2. The reply from Remote Client2 follows the same path but in reverse.
- Office mode addresses used by both gateways must be non-overlapping.

Including the Office Mode Range of Addresses in the VPN Domain of the Gateway

Another way to achieve client to client communication is by defining an Office Mode range of addresses for remote clients, and including this range of addresses in the VPN domain of the gateway that acts as the Hub. Each remote client directs communication to the remote peer via the gateway; from the remote client's perspective, its peer belongs to the VPN domain of the gateway.

Note - Including the Office mode address range within the VPN domain of a gateway is only possible with *NG with Application Intelligence*.

Configuring VPN Routing for Remote Access VPN

Common VPN routing scenarios can be configured through a VPN star community, but not all VPN routing configuration is handled through SmartDashboard. VPN routing between gateways (star or mesh) can be also be configured by editing the configuration file `$FWDIR\conf\vpn_route.conf`.

VPN routing cannot be configured between gateways that do not belong to a VPN community.

Enabling Hub Mode for Remote Access clients

1. On the **Remote Access** page of the **Gateway properties** window, **Hub Mode configuration** section, select **Allow SecureClient to route all traffic through this gateway**.
2. On the Properties window of the **Remote Access** community, **Participating Gateways** page, set the gateway that functions as the "Hub".
3. On the **Participant User Groups** page, select the remote clients.
4. Create an appropriate access control rule in the Security Policy Rule Base. VPN routing traffic is handled in the Security Policy Rule Base as a single connection, matched to *one rule only*.
5. Configure the profile on the remote client to route all communication through the designated gateway.

Configuration of Client to Client Routing by Including the Office Mode Range of Addresses in the VPN Domain of the Gateway

To configure VPN routing for remote access clients via the VPN domain, add the Office mode range of addresses to the VPN domain of the gateway:

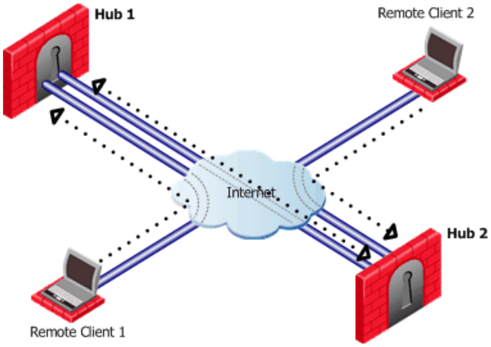
1. In SmartDashboard, create an address range object for the Office Mode addresses.
2. Create a group that contains both the VPN domain and Office mode range.
3. On the **General properties** window of the gateway object **> Topology page > VPN domain** section, select **Manually defined**.
4. Select the group that contains both the VPN domain of the gateway and the Office mode addresses.

The remote clients must connect to the site and perform a site update before they can communicate with each other.

Client to Client via Multiple Hubs Using Hub Mode

Figure 24-6 shows two remote clients each configured to work in Hub mode with a different gateway:

Figure 24-6 Remote clients with different Hubs



Remote Client 1 works in Hub mode with Hub 1. Remote Client 2 works in Hub mode with the Hub 2. In order for VPN routing to be performed correctly:

- Remote clients must be working in Office mode

- Office mode address range of each gateway must be included in the vpn_route.conf file installed on the other gateway.

Table 24-1

Destination	Next hop router interface	Install On
Hub1_OfficeMode_range	Hub1	Hub2
Hub2_OfficeMode_range	Hub2	Hub1

When Remote Client 1 communicates with Remote Client 2:

- The traffic first goes to the Hub 1, since Remote Client 1 is working in Hub mode with Hub 1.
- Hub 1 identifies Remote Client 2's IP address as belonging to the Office mode range of Hub 2.
- The `vpn_route.conf` file on Hub 1 identifies the next hop for this traffic as Hub 2.
- The traffic reaches the Hub 2; Hub 2 redirects the communication to Remote Client 2.

Chapter 25

Link Selection for Remote Access Clients

In This Chapter

Overview

Link Selection is a method used to determine which interface is used for incoming and outgoing VPN traffic as well as the best possible path. Using the *Link Selection* mechanisms, the administrator can focus individually on which IP addresses are used for VPN traffic from remote access clients on each gateway.

For more information on *Link Selection*, see *"Link Selection" on page 207*.

IP Selection by Remote Peer

There are several method that can determine how remote access clients resolve the IP address of the local gateway. Remote peers can connect to the local gateway using:

- **Always use this IP address:**
    - **Main address** - The VPN tunnel is created with the gateway main IP, specified in the **IP Address** field on the **General Properties** page of the gateway.
    - **Selected address from topology table** - The VPN tunnel is created with the gateway using a selected IP address chosen from the drop down menu that lists the IP addresses configured in the **Topology** page of the gateway.
    - **Statically NATed IP** - The VPN tunnel is created using a NATed IP address. This address is not required to be listed in the topology tab.
    - **Calculate IP based on network topology** - An IP address that is calculated by network topology. Using *Calculate IP Based on Network Topology*, the remote access client discovers the appropriate IP address of the gateway through the topology information defined in the gateway's network object. Since a remote access client can be in a different place each time it connects, a new calculation takes place each time the client connects. For this to succeed, all the gateway's interfaces must have their topology properly configured.
- **Use a probing method:**
    - **Using ongoing probing** - When a session is initiated, all possible destination IP addresses continuously receive RDP packets. The VPN tunnel uses the first IP to respond (or to a primary IP if a primary IP is configured and active), and stays with this IP until the IP stops responding. The RDP probing is activated when a remote access client connects and continues as a background process.

- **Using one time probing** - When a remote access client connects, all possible destination IP addresses receive an RDP session to test the route. The first IP to respond is chosen, and stays chosen until the next time the client reconnects.

IP addresses you do not wish to be probed (i.e., internal IP addresses) may be removed from the list of IP's to be probed - see "Resolving Addresses via Probing" on page 224.

For both the probing options (one-time and on-going) a *Primary Address* can be assigned.

Primary Address

When implementing one of the probing methods on a gateway that has a number of IP addresses for VPN, one of the IP addresses can be designated as a Primary Address. As a result, peers assign the primary address IP a higher priority even if other interfaces have a lower metric.

Enabling a primary address has no influence on the IP selected for outgoing VPN traffic. If the remote access client connects to a peer gateway that has a primary address defined, then the remote access client will connect to the primary address (if active) regardless of network speed (latency) or route metrics.

If the primary address fails and the connection fails over to a backup, the VPN tunnel will stay with the backup until the primary becomes available again.

Link Selection for Remote Access Scenarios

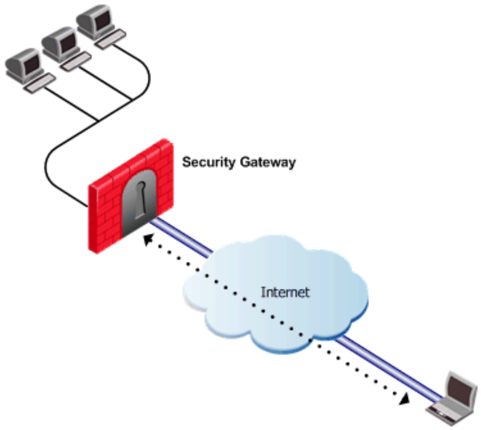
Link Selection may be used in different infrastructures. This section describes scenarios that benefit from the implementation of *Link Selection*.

In This Section

Gateway with a Single External IP Address

This is the simplest case. In Figure 25-1, the local gateway has a single external interface for VPN:

Figure 25-1 Single IP for Remote Access Client



Now consider configuration for the local gateway in terms of:

- How remote access clients select an IP on the local gateway for VPN traffic

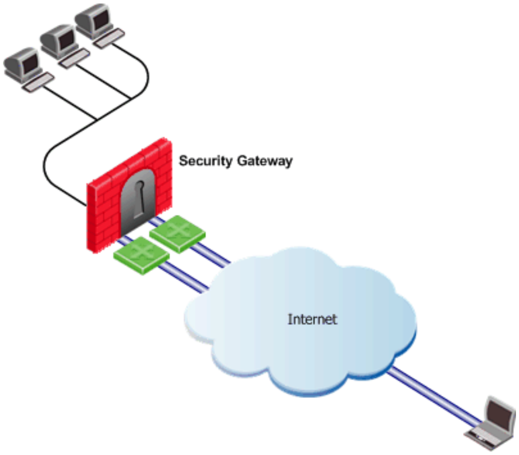
Since there is only one interface available for VPN:

- For determining how remote access clients discover the local gateways IP for VPN, select **Main address** or choose an IP address from the **Selected address from topology table** drop down menu.
- If the IP address is located behind a static NAT device, select **Statically NATed IP**.

Gateway with Multiple External IP Addresses

In this scenario, the local gateway has two external IP addresses available for remote access clients.

Figure 25-2 Connecting to Gateway with Multiple Interfaces

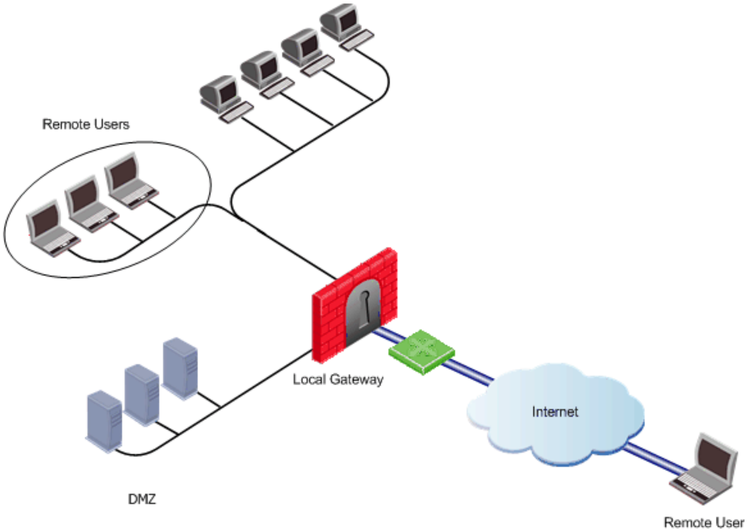


For determining how remote access clients discover the local gateway's IP address, use *ongoing probing*. The remote access client connects through the first IP address to respond (unless there is a primary IP address configured) and stays with this IP until it stops responding.

Calculate IP Based on Network Topology

Since a remote access client can be in a different place each time it connects, a new calculation takes place each time the client connects.

Figure 25-3 Accessing the internal network



In Figure 25-3, all remote users access the internal network by calculating which interface to use based on its current location. The remote user located outside the network will use an external interface and the remote users located within the network will use an internal interface.

Configuring Link Selection

Link selection is configured on each gateway in the **Topology > Link Selection** window. The settings apply to both gateway to gateway connections and remote access client to gateway connections.

The Link Selection configuration for remote users can be configured separately by using the following attributes.

These settings will override the settings configured on the Link Selection page.

Using Dbedit:

- Change the value `apply_resolving_mechanism_to_SR` to **false** on the gateway's object.
- `Use ip_resolution_mechanism` to select a link selection method. The valid values are:
    - `mainIpVpn`
    - `singleIpVpn`
    - `singleNATIpVPN`
    - `topologyCalc`
    - `oneTimeProb`
    - `ongoingProb`
- `single_VPN_IP_RA` – If a specific IP address from the gateway's topology or statically NATed IP was configured, the IP address should be set in this attribute.
- `interface_resolving_ha_primary_if` – The primary IP address used for one-time / ongoing probing.
- `use_interface_IP` – Used only for one-time / ongoing probing. Set to **true** if all IP addresses defined in topology tab should be probed. Set to **false** if the manual list of IP addresses should be probed.
- `available_VPN_IP_list` - Used only for one-time / ongoing probing. List of IP addresses that should be probed. (This list is used only if the value of `use_interface_IP` is **false**).

Configuring the Early Version Compatibility Resolving Mechanism

In SmartDashboard:

1. Click **Policy > Global Properties > Remote Access > Early Versions Compatibility.**
2. Select **Static calculation based on network topology** if remote access clients downloading topology from pre NGX gateways should use topology calculation to resolve IP addresses.
3. Select **Dynamic interface resolving mechanism** to convert to one of the methods in Table 25-1.

The method downloaded in topology to remote access clients by Pre NGX gateways will be “converted” as follows:

Table 25-1 Backward Compatibility

Gateway Method	Method used by Remote Access Client downloading topology from a Pre NGX Gateway
Selected address from topology table	Ongoing Probing
Statically NATed IP	Ongoing Probing
Use DNS resolving	Ongoing Probing
Calculate IP based on network topology	Main IP
Main IP	Main IP
Ongoing Probing	Ongoing Probing
One Time Probing	One Time Probing

Note - If the manual IP address list for probing is configured, Pre NGX (R60) remote access clients will probe all of the IP addresses and not just those listed in the IP address list. If Primary Address is configured, Pre NGX (R60) remote access clients will treat all IP addresses with the same priority (and will ignore the Primary Address).

Chapter **26**

Using Directional VPN for Remote Access

Enhancements to Remote Access Communities

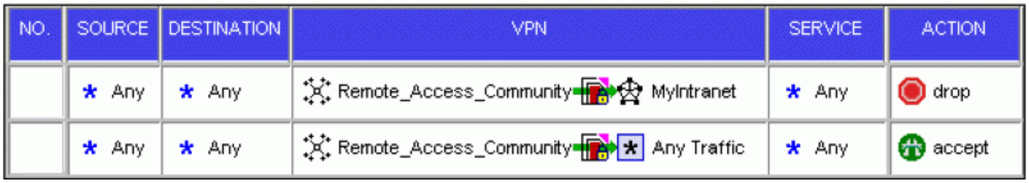
Remote Access communities now support:

- Directional VPN
- User groups in the destination column of a rule

Directional VPN in RA Communities

With Directional VPN configured for Remote Access communities, the option exists to reject connections to or from a particular network object. Consider the rules in Figure 26-1:

Figure 26-1 Directional VPN in Remote Access Communities



NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION
	* Any	* Any	Remote_Access_Community → MyIntranet	* Any	drop
	* Any	* Any	Remote_Access_Community → * Any Traffic	* Any	accept

Connections are not allowed between remote users and hosts within the “MyIntranet” VPN community. Every other connection originating in the Remote Access Community, whether inside or outside of the VPN communities, is allowed.

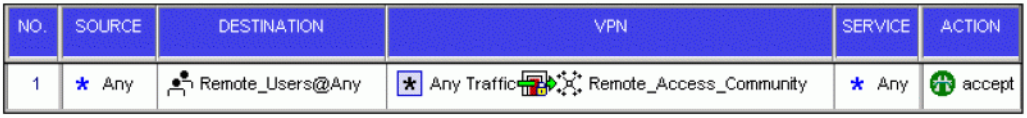
User Groups as the Destination in RA communities

User groups can be placed in the destination column of a rule. This makes:

- Configuring client to client connections easier
- Configuring "back connections" between a remote client and a gateway possible.

Figure 26-2 shows a directional rule for remote access communities which allows return "back" connections.

Figure 26-2 Directional rules in a Remote Access Community



NO.	SOURCE	DESTINATION	VPN	SERVICE	ACTION
1	* Any	Remote_Users@Any	* Any Traffic → Remote_Access_Community	* Any	accept

To include user groups in the destination column of a rule:

- The rule must be directional
- In the VPN column, the Remote Access community must be configured as the endpoint destination

Configuring Directional VPN with Remote Access Communities

To configure Directional VPN with Remote Access communities:

1. In **Global Properties > VPN** page **> Advanced >** Select **Enable VPN Directional Match in VPN Column**.
2. Right-click inside the VPN column of the appropriate rule, and select **Edit...** or **Add Direction** from the pop-up menu.

   The **VPN Match Conditions** window opens.
3. Click **Add...**

   The **Directional VPN Match Conditions** window opens.
4. From the drop-down box on the right, select the source of the connection.
5. From the drop-down box on the left, select the connection's destination.
6. Click **OK**.

Chapter 27

Remote Access Advanced Configuration

In This Chapter

Non-Private Client IP Addresses

Remote Access Connections

Suppose a SecuRemote/SecureClient user connects from behind a NAT device, using a non-private IP address that belongs to another organization. During the life of the connection, the gateway routes all traffic intended for that non-private IP address to the SecuRemote/SecureClient user, even traffic intended for the real owner of the IP address.

Solving Remote Access Issues

Set the `vpn_restrict_client_phase2_id in` the `Objects_5_0.C` file to the appropriate value, as follows:

Table 27-1 vpn_restrict_client_phase2_id

value	meaning
`om_only`	SecuRemote/SecureClient behind NAT devices can only connect using Office Mode.
`private_and_om`	SecuRemote/SecureClient can connect using either: • using Office Mode, *or* • when using private IP addresses (where the meaning of “private” is specified in the **NAT** page of the **Global Properties** window)
`none`	This setting (the default) does not address the problem described above.

Note - If the user is restricted to Office Mode or to the use of a private IP address, and attempts another type of connection, the connection will be dropped and a log will be sent to the SmartView Tracker.

Preventing a Client Inside the Encryption Domain from Encrypting

The Problem

If a SecuRemote/SecureClient located inside the VPN domain of one gateway opens a connection to a host inside the VPN domain of another gateway, the connection will be encrypted twice (once by the SecuRemote/SecureClient and again by the gateway) and decrypted only once (by the peer gateway).

The Solution

To prevent this from happening, configure the SecuRemote/SecureClient not to encrypt if both the SecuRemote/SecureClient and the host (the end-points of the connection) are in the VPN domains of gateways managed by the same Security Management server.

To do this, enable the `send_clear_traffic_between_encryption_domains` property in `objects_5_0.C`.

Note - If you enable this feature, ensure that a VPN is defined between the gateways. This feature is disabled when more than one site is defined in SecuRemote/SecureClient.

When the Client Has a Private Address

If the `send_clear_traffic_between_encryption_domains` property is enabled, a problem can arise when the gateway's VPN domain includes private addresses, where the meaning of "private" is specified in the **Non Unique IP Address Ranges** page of the **Global Properties** window.

If the SecuRemote/SecureClient connects from outside the VPN domain (for example, from a hotel) and is assigned (by the ISP or a NAT device) a private IP address which happens to be in the gateway's VPN domain, then SecuRemote/SecureClient will not encrypt when connecting to the VPN domain, and the connection will be dropped because it is in the clear.

You can configure SecuRemote/SecureClient to encrypt this traffic as follows:

- To encrypt traffic from private addresses, enable the `send_clear_except_for_non_unique` property in `objects_5_0.C`.
- To encrypt traffic from specific IP addresses, proceed as follows:

1. Define a group consisting of those addresses.
2. Enable the `send_clear_except_for_specific_addresses` property in `objects_5_0.C`.
3. Set `send_clear_except_for_address_group` to the name of the group defined in step 1.

Note - This feature is disabled when more than one site is defined in SecuRemote/SecureClient.

Working in Connect Mode While Not Connected

For users connected in SecuRemote/SecureClient Connect Mode, you can reduce the frequency of authentication by enabling the `allow_clear_traffic_while_disconnected` property in `objects_5_0.C`. SecuRemote/SecureClient will then not encrypt traffic to the peer encryption domain *when not connected*. This will prevent unnecessary authentication when connecting to unencrypted services in the peer encryption domain.

For example, if the site includes both private and public HTTP servers, there is no need to encrypt traffic to the public site. To prevent a user from unnecessarily authenticating only because she is an internal user, configure the following two rules in the Desktop Policy:

Table 27-2

Source	Destination	Service	Action
encryption domain	encryption domain	Any	Accept
Any	encryption domain	Any	Encrypt

.

Note - If you enable this feature, you must ensure that a VPN is defined between the gateways. This feature applies only to Connect Mode. This feature is disabled when more than one site is defined in SecuRemote/SecureClient.

Authentication Timeout and Password Caching

The Problem

Users consider multiple authentications during the course of a single session to be a nuisance. At the same time, these multiple authentications are an effective means of ensuring that the session has not been hijacked (for example, if the user steps away from the client for a period of time). The problem is finding the correct balance between convenience and security.

The Solution

Multiple authentication can be reduced by two means:

- Increasing the authentication timeout interval
- Caching the user's password

Authentication Timeout Interval

To specify the length of time between re-authentications, select **Policy> Global Properties - Remote Access** and in the **Authentication Timeout** section, enter a value in **Validation timeout**. Alternatively, check **Use default value**.

For Connect Mode, the countdown to the timeout begins from the time that the Client is connected.

Password Caching

When the timeout expires, the user will be asked to authenticate again. If password-caching is enabled, SecuRemote/SecureClient will supply the cached password automatically and the authentication will take place transparently to the user. In other words, the user will not be aware that re-authentication has taken place.

Password caching is possible only for multiple-use passwords. If the user's authentication scheme implement one-time passwords (for example, SecurID), then passwords cannot be cached, and the user will be asked to re-authenticate when the authentication time-out expires. For these schemes, this feature should *not* be implemented.

Password caching is specified in the SecureClient's **Authentication** window.

SecuRemote/SecureClient and Secure Domain Logon (SDL)

The Problem

When a SecuRemote/SecureClient user logs on to a domain controller, the user has not yet entered his or her SecuRemote/SecureClient credentials and so the connection to the domain controller is not encrypted.

The Solution

When the Secure Domain Logon (SDL) feature is enabled, then after the user enters the OS user name and password (but before the connection to the domain controller is started), **SecuRemote Client User Authentication** window is displayed. When the user enters the SecuRemote/SecureClient credentials, the connection to the domain controller takes place over an encrypted tunnel.

Enabling and Disabling Secure Domain Logon

To enable Secure Domain Logon (SDL), select **Enable Secure Domain Logon** from the SecuRemote/SecureClient **Passwords** menu.

Note the following:

- For Windows NT and Windows 2000:
  - SDL can only be enabled by the administrator, and only if the machine is configured as part of a domain.
  - You must reboot after enabling or disabling SDL.
- If you are using WINS (see "WINS (Connect Mode Only)" on page 585), configure WINS *before* enabling SDL.
- Do not change the machine domain configuration when Secure Domain Logon is enabled.

Domain Controller Name Resolution

If SecuRemote/SecureClient is configured in Connect Mode and Office Mode, SecuRemote/SecureClient automatically resolves the NT domain name using dynamic WINS.

Otherwise, SecuRemote/SecureClient resolves the NT domain name using either LMHOSTS or WINS.

LMHOSTS

The LMHOSTS name resolution service can be used in both LAN and dial-up configurations as follows:

Enter the relevant information (see Figure 27-1) the `$FWDIR/conf/dnsinfo.C` file on the security gateway, and install the policy.

Figure 27-1 Syntax

```
(
     :LMdata(
            :(
                    :ipaddr (<IP address>)
                    :name (<host name>)
                    :domain (<domain name>)
            )
            :(
                    :ipaddr (<IP address>)
                    :name (<host name>)
                    :domain (<domain name>)
            )
     )
)
```

When SecuRemote/SecureClient updates the topology, the name resolution data will be automatically transferred to the `dnsinfo` entry of the SecuRemote/SecureClient `userc.C` file and then to its `LMHOSTS` file.

WINS (Connect Mode Only)

The WINS name resolution service can be used in dial-up configurations only. It is not supported on Win 9x platforms.

To use the WINS, proceed as follows on the SecuRemote/SecureClient virtual adapter:

Warning - You must do this *before* enabling SDL (see "Enabling and Disabling Secure Domain Logon" on page 584).

1. Specify the primary and, optionally, the secondary WINS servers protected by the security gateway.
2. Reboot the SecuRemote/SecureClient machine.

Configuring SDL Timeout

Because SDL depends on the synchronization of concurrent processes, flexibility in defining timeouts is important.

The SDL Timeout feature of Secure Domain Logon allows you to define the period during which a user must enter his or her domain controller credentials. When the allocated time expires and no cached information is used (if applicable), the logon fails.

The timeout is controlled by the `sdl_netlogon_timeout (<value in seconds>)` parameter in the file `Objects_5_0.C`.

Note - This feature is not applicable if the Auto Local Logon option is enabled (Connect Mode Only).

Cached Information

When the SecuRemote/SecureClient machine successfully logs on to a domain controller, the user's profile is saved in cache. This cached information will be used if subsequent logons to the domain controller fail, for whatever reason.

To configure this option in the client registry, proceed as follows:

1. Go to `HKLM\Software\Microsoft\Windows NT\Current Version\Winlogon`.
2. Create a new key `CachedLogonCount` with the valid range of values from 0 to 50. The value of the key is the number of previous logon attempts that a server will cache.

   A value of 0 disables logon caching and any value above 50 will only cache 50 logon attempts.

Configuring Secure Domain Logon

1. Configure the SecuRemote Client to use LMHOSTS (all platforms) or WINS (all platforms except Win 9x).
2. For Win NT and Win 2000, configure the SDL timeout.
3. Define the site where the domain controller resides and download/update the topology.
4. If the client is not already a domain member, configure the machine as a domain member.
5. For Win NT and 2000:
   - Enable Auto Local Logon (optional)
   - Enable Secure Domain Logon
6. Reboot the computer and logon.

Using Secure Domain Logon

After you have rebooted the computer:

1. When the Windows NT **Logon** window is displayed, enter the operating system credentials.
2. Click **OK**.

   The SecuRemote **Logon** window is displayed.
3. Enter the SecuRemote credentials in the defined time (see "Configuring SDL Timeout" on page 586).

If you fail to logon and no cached information is used, wait one minute and try again.

If SDL is already configured on the client, the administrator can customize SecuRemote/SecureClient installation packages with SDL enabled by default, thereby relieving the user of the need to configure SDL manually. This can be done in two ways:

- Create a self-extracting client package using the SecureClient Packaging Tool (see "Packaging SecureClient") and select **Enable Secure Domain Logon (SDL)** in the **Operating System Logon** window *or*
- Edit the `product.ini` file in the SecuRemote installation package by setting the value of `EnableSDL` to `1`. See "Userc.C and Product.ini Configuration Files" for more information.

Back Connections (Server to Client)

Back connections (connections from the server to the client) are required by certain applications, such as Xterm. These connections do not have to be explicitly defined in the Rule Base. To achieve this, when a user logs on to a site, the username and IP address are stored in an authentication database for 15 minutes (this time frame is configurable). During this time, all back connections from server to client are allowed. After this time, back connections are sent in clear.

Sending Keep-Alive Packets to the Server

To enable the 15 minute interval, configure back connections so that Keep Alive transmissions are sent by the client to the server. This is especially necessary when a NAT device is used.

By sending Keep Alive packets, the IP Address maintained in the authentication database is constantly renewed. In the **Remote Access** page of the **Global Properties** window, check **Enable back connections (from gateway to client)** and specify a value for **Send Keep-Alive packet to the Gateway**.

Auto Topology Update (Connect Mode only)

You can configure SecuRemote Clients to automatically update a site's topology either when starting SecuRemote or just before the IKE key exchange in the **Remote Access** page of the **Global Properties** window.

In this window, the system administrator can:

- **Update topology every ... hours** — The site's topology will be updated before the next key exchange if the defined period has elapsed since the last topology update.

The following features become available if **Update topology every ... hours** is enabled:

- **Automatic update** — If enabled, the site will be updated after the key exchange (according to the value of **Update topology every ... hours**). This will allow to avoid prompting the user to update sites.
- **Upon VPN-1 SecuRemote/SecureClient startup** — If enabled, the user will be prompted to update the topology when the SecuRemote Client starts. If the user is not connected to the network when the SecuRemote Client starts, he or she can reject the prompt. In this case the topology will be automatically updated after the next key exchange with the site.

How to Work with non-Check Point Firewalls

If a SecuRemote/SecureClients is located behind a non-Check Point firewall, the following ports must be opened on the firewall to allow SecuRemote/SecureClient traffic to pass:

Table 27-3 ports to open for non-Check Point firewalls

port	explanation
UDP port 500	always, even if using IKE over TCP
TCP port 500	only if using IKE over TCP
IP protocol 50 ESP	unless always using UDP encapsulation
UDP port 2746	configurable; only if using UDP encapsulation
UDP port 259	only if using MEP, interface resolving or interface High Availability

Early SecuRemote/SecureClients Versions

Check Point's recommended upgrade sequence is Security Management server followed by Modules followed by SecuRemote/SecureClients. There will then be a period of time during which earlier version SecuRemote/SecureClients will be connecting to upgraded Modules. If any of these SecuRemote/SecureClients are Version 4.1, backwards compatibility must be enabled in the **Early Versions Compatibility** page (under **Remote Access**) of the **Global Properties** window.

Required policy for all desktops defines the type of policy that will be enforced on Version 4.1 Clients. From NG forward the security policy applied to SecuRemote/SecureClient is automatically the policy defined in the Desktop Security Rule Base. For Version 4.1 Clients, you must decide whether or not to enforce a policy, and if *yes*, whether or not to allow:

- all outgoing and all encrypted connections
- only outgoing connections
- only encrypted connections

If you select **Client is enforcing required policy**, an additional SCV check which verifies the Client's security policy is performed.

Resolving Internal Names with the SecuRemote DNS Server

The Problem

The SecuRemote/SecureClient must resolve the names of internal hosts (behind the security gateway) with non-unique IP addresses using an internal DNS server.

The Solution

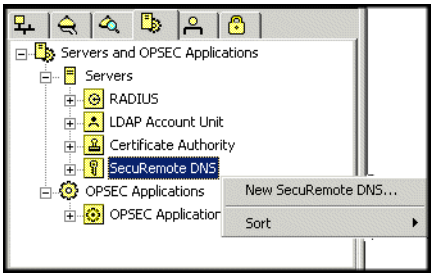
The simplest solution is to use Connect Mode and Office Mode. Otherwise, use the split DNS feature by defining a SecuRemote DNS Server.

The SecuRemote DNS Server is an object that represents an internal DNS server that can be used to resolve internal names with unregistered, (RFC 1981-style) IP addresses. It is best to encrypt the DNS resolution of these internal names. Not all DNS traffic should be encrypted, as this would mean that every DNS resolution would require authentication.

Configuring the SecuRemote DNS Server

1. Create a new SecuRemote DNS Server from the Objects Tree (Figure 27-2).

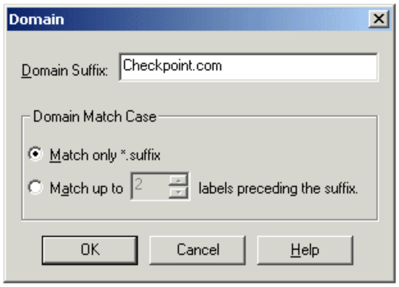
Figure 27-2 Create a SecuRemote DNS Server



2. In the **SecuRemote DNS Properties** window

   - **General** tab — Configure the general settings of the SecuRemote DNS Server as well as the host on which the SecuRemote DNS Server.
   - **Domains** tab — Add new domains or edit and remove existing domains.

Figure 27-3 Domain tab



Domain Suffix: Checkpoint.com

Domain Match Case

● Match only *.suffix

○ Match up to 2 labels preceding the suffix.

3. In the **Domain** tab (Figure 27-3), define the domain suffix and the matching rule. Names in the domain that correspond to the rule will be resolved by the SecuRemote DNS Server. All other names will be resolved by the SecuRemote client's default DNS server.

   - Specify the **Domain Suffix** for which the SecuRemote DNS Server will resolve the internal names (for example, checkpoint.com).
   - Select **Match only *.suffix** to specify that the maximum number of labels resolved will be 1.

     For example, if **Domain Suffix** is "checkpoint.com" and **Match only *.suffix** is selected (that is, the maximum prefix label count is in effect 1) then the SecuRemote DNS Server will be used to resolve "www.checkpoint.com" and "whatever.checkpoint.com" but not "www.internal.checkpoint.com."
   - Select **Match up to...labels preceding the suffix** to increase the number of labels to be matched.

     For example, if **Domain Suffix** is "checkpoint.com" and **Match up to...labels preceding the suffix** is selected and set to 3, then the SecuRemote DNS Server will be used to resolve "www.checkpoint.com" and "www.internal.checkpoint.com" but not "www.internal.inside.checkpoint.com".

Additional Considerations

Split DNS is disabled in the following cases:

- In Connect mode, while disconnected.

  To override, set `disable_split_dns_when_disconnected` in the SecuRemote/SecureClient `userc.C` file to `false`.
- In connect mode, while connected in Office Mode.

  To override, set `disable_split_dns_in_om` in the SecuRemote/SecureClient `userc.C` file to `false`.

Chapter 28

Multiple Entry Point for Remote Access VPNs

In This Chapter

The Need for Multiple Entry Point Gateways

The security gateway provides a single point of entry to the internal network. It is the gateway that makes the internal network “available” to remote machines. If the gateway fails, the internal network is no longer available. It therefore makes good sense to have *Multiple Entry Points* (MEP) to the same network.

The Check Point Solution for Multiple Entry Points

In a MEPed environment, more than one security gateway is both protecting and giving access to the same VPN domain. How a remote user selects a gateway in order to reach a destination IP address depends on how the MEPed gateways have been configured, which in turn depends on the requirements of the organization.

For more information, see "Multiple Entry Point VPNs" on page 229.

The Check Point solution for multiple entry points is based on a proprietary *Probing Protocol* (PP) that tests gateway availability. The MEPed gateways do not have to be in the same location; they can be widely-spaced, geographically.

Note - In a MEPed gateway environment, the only remote client supported is the Check Point SecuRemote/SecureClient.

SecureClient Connect Profiles and MEP

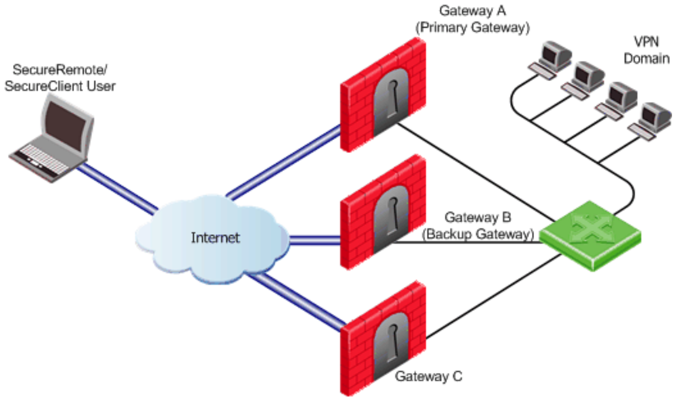
There are three methods used to choose which gateway will be used as the entry point for any given connection:

- **First to reply**. In a First to Reply MEP environment, SecureClient attempts to connect to the gateway configured in the profile. If the configured gateway does not reply, the first gateway to respond is chosen.
- **Primary/Backup.** With this method, SecureClient attempts to connect to the Primary gateway first. If the Primary gateway does not reply, SecureClient attempts to connect to the Backup gateway. If the Backup gateway does not reply, there are no further attempts to connect.
- **Random Selection**. In a Load Sharing MEP environment, SecureClient randomly selects a gateway and assigns the gateway priority. The remote peer stays with this chosen gateway for all subsequent connections to host machines within the VPN domain. Load distribution takes place on the level of "different clients", rather than the level of "endpoints in a connection". In addition, SecureClient ignores whatever gateway is configured as the "connect to gateway" in the profile.

Preferred Backup Gateway

Preferred Backup gateway allows remote hosts to choose which gateway in the MEP configuration will be the backup gateway. All other gateways in the MEP configuration will be ignored should the first two gateways become unavailable.

Figure 28-1 Preferred Backup Gateway



In this scenario:

- The VPN Domain is behind three gateways - A, B and C.
- Gateway A is the Primary gateway.
- Gateway B is the Backup gateway when gateway A is not available.
- Should gateway A and gateway B become unavailable, the remote host will not attempt to connect to gateway C.

Visitor Mode and MEP

Since the RDP gateway discovery mechanism used in a MEPed environment runs over UDP, this creates a special challenge for SecureClient in Visitor Mode, since all traffic is tunnelled over a regular TCP connection.

In a MEPed environment:

- The RDP probing protocol is not used; instead, a special Visitor Mode handshake is employed.
- When a MEP failover occurs, SecureClient disconnects and the user needs to reconnect to the site in the usual way.
- In a *Primary-Backup* configuration, the connection will failover to the backup gateway should the primary gateway become unavailable. Even if the Primary gateway is restored, the connection does not return to the primary gateway.
- All the gateways in the MEP:
    1. Must support visitor mode.
    2. The user must be working with a Visitor Mode enabled profile.

Routing Return Packets

To make sure return packets are routed correctly, the MEPed gateway makes use of IP pool NAT.

IP Pool NAT

IP pool NAT is a type of NAT in which source IP addresses from remote VPN domains are mapped to an IP address drawing from a pool of registered IP addresses. In order to maintain symmetric sessions using MEPed gateways, the MEPed gateway performs NAT using a range of IP addresses dedicated to that specific gateway and should be routed within the internal network to the originating gateway. When the returning packets reach the gateway, the gateway restores the original source IP address and forwards the packets to the source.

Note - When Office Mode is enabled, there is no need to configure IP Pool NAT since Office Mode dynamically assigns IP’s to remote hosts.

Disabling MEP

When MEP is disabled, MEP RDP probing and fail over will not be performed. As a result, remote hosts will connect to the gateway defined without considering the MEP configuration.

Configuring MEP

In This Section

To configure MEP, decide on the MEP selection method:

- First to Respond
- Primary/Backup
- Load Distribution

First to Respond

When more than one gateway leads to the same (overlapping) VPN domain, they are considered MEPed by the remote peer, and the first gateway to respond to the probing protocol is chosen. To configure *first to respond*, define that part of the network that is shared by all the gateways into a single group and assign that group as the VPN domain.

On the **Properties** window of each gateway network object, **Topology** page > **VPN Domain** section, select **Manually defined**, and define the *same* VPN domain for all gateways.

Primary-Backup

1. In the **Global Properties** window, **VPN > Advanced** page, select **Enable Backup Gateway**.
2. In the network objects tree, **Groups** section, create a group consisting of the gateways that act as backup gateways.
3. On the **VPN** page of the network object selected as the Primary gateway, select **Use Backup Gateways**, and select the group of backup gateways from the drop-down box. This gateway now functions as the primary gateway for a specific VPN domain.
4. Define the VPN for the backup gateway(s). Backup gateways do not always have a VPN domain of their own. They simply back-up the primary. If the backup gateway does not have a VPN domain of its own, the VPN domain should include only the backup gateway itself:

   a. On the **Properties** window of the backup network object, **Topology** page **> VPN Domain** section, select **Manually defined**.

   b. Select a group or network that contains only the backup gateway.

   If the backup *does* have a VPN domain:

   a. Verify that the IP address of the backup gateway is *not* included in the VPN domain of the primary.

   b. For each backup gateway, define a VPN domain that does *not* overlap with the VPN domain of any other backup gateway.

Note - There must be no overlap between the VPN domain of the primary gateway and the VPN domain of the backup gateway(s); that is, no IP address can belong to both.

5. Configure IP pool NAT to handle return packets. See: "Configuring Return Packets" on page 603.

Load Distribution

1. In the **Global Properties** window, **Remote Access > VPN Basic** page, **Load distribution** section, select **Enable load distribution for Multiple Entry Point configurations (Remote Access connections)**.
2. Define the same VPN domain for all gateways.

Checking this option also means that load distribution is dynamic, that is the remote client randomly selects a gateway.

Configuring Return Packets

Return packets are handled with IP pool NAT addresses belonging to the gateway.

Configuring IP pool NAT

In **Global Properties > NAT** page, select **Enable IP Pool NAT for SecuRemote/SecureClient and gateway to gateway connections**. Then:

1. For each gateway, create a network object that represents the IP pool NAT addresses for that gateway. The IP pool can be a network, group, or address range. For an address range, for example:
    - On the network objects tree, right-click **Network Objects** branch **> New > Address Range...** The **Address Range Properties** window opens.
    - On the **General** tab, enter the first IP and last IP of the address range.
    - Click **OK**. In the network objects tree, **Address Ranges** branch, the new address range appears.
2. On the gateway object where IP pool NAT translation is performed, **Gateway Properties** window, **NAT** page, **IP Pools (for Gateways)** section, select either (or both):
    - **Use IP Pool NAT for VPN client connections.**
    - **Use IP Pool NAT for gateway to gateway connections.**
    - In the **Allocate IP Addresses from** field, select the address range you created.
    - Decide after how many minutes unused addressees are returned to the IP pool.
    - Click **OK**.
3. Edit the routing table of each internal router, so that packets with an a IP address assigned from the NAT pool are routed to the appropriate gateway.

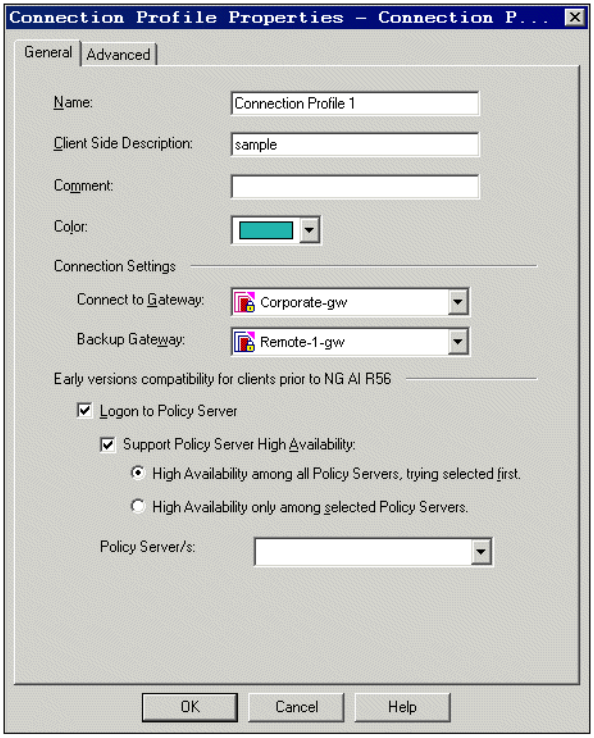
Configuring Preferred Backup Gateway

In SmartDashboard:

1. Click **Manage > Remote Access > Connection Profiles**.
2. Select existing profile and click **Edit** or click **New > Connection Profile**.

   The **Connection Profile Properties** window is displayed.

Figure 28-2 Connection Profile Properties Page



Connection Profile Properties - Connection P...

General | Advanced

Name: Connection Profile 1

Client Side Description: sample

Comment:

Color:

Connection Settings

Connect to Gateway: Corporate-gw

Backup Gateway: Remote-1-gw

Early versions compatibility for clients prior to NG AI R56

☑ Logon to Policy Server

☑ Support Policy Server High Availability:

● High Availability among all Policy Servers, trying selected first.

○ High Availability only among selected Policy Servers.

Policy Server/s:

3. In the **Connect to Gateway** and **Backup Gateway** fields, use the drop down menu to select the gateways that will function as the primary and backup gateways for this profile.
4. Click **OK**.

Disabling MEP

Disabling MEP is configured by setting the following Dbedit command to **true**:

- desktop_disable_mep

Chapter 29

Userc.C and Product.ini Configuration Files

In This Chapter

Introduction to Userc.C and Product.ini

The VPN administrator can use the Packaging Tool to produce customized SecuRemote/SecureClient packages for distribution to end-users. The Packaging Tool changes the behavior of SecuRemote/SecureClient by changing the values of the properties in the `Userc.C` and `Product.ini` files contained in the package.

However, not all of the properties in these files can be changed using the Packaging Tool. It is possible to changes the behavior of SecuRemote/SecureClient by manually editing the `Userc.C` and `Product.ini` files in the SecuRemote/SecureClient package, before distributing the package to end users.

The Userc.C File

Structure of Userc.C

The `Userc.C` configuration text file contains has three sections. *Global*, *Managers*, and *Gateways*.

- **Global**—Properties that are not specific to the site (managed by a single Security Management server) or to the peer gateway. It does not change on the client machine. To change the Global Properties section of the objects database, do *not* make any manual changes to the Global section of `userc.C`. Either edit the SmartDashboard Global Properties, or use the **DBedit** command line or the graphical Database Tool on the Security Management server.
- **Managers**—Properties that apply per Security Management server. Updated whenever the end user performs a Site Update.
- **Gateway**—Properties that are specific to a particular gateway. Updated whenever the end user performs a Site Update.

The section of the file where each parameter resides is indicated in the `Userc.C` file parameter tables (below), in the column labelled **Location in Userc.C.**

How Userc.C Is Automatically Updated

When the Security Policy is installed on the gateways, the objects database is also installed on the gateways. The part of the database that relates to remote clients is sent to the Topology Server on the gateway. When the clients perform a Site Update, they are actually downloading the Topology information from the Topology server, which updates the Managers and gateway sections of `userc.C` on the clients. The file is stored on the client machine in the `SecuRemote\database` directory. The parameters appear in the `options` section.

How to Manually Edit Userc.C

Do not make any manual changes to the Global section of `userc.C`.

Manually edit the Managers and Gateway sections of `userc.C` as follows:

1. Extract `userc.C` from the original SecuRemote/SecureClient `tgz` format installation package.
2. Edit the `userc.C` parameters, as needed.

> **Warning -** SecuRemote/SecureClient performs minimal syntax checking for the `userc.C` file. If a parameter is edited incorrectly, the file may become corrupted, and sites may need to be redefined.

3. Recreate the `tgz` file.

The Product.ini file

The `Product.ini` configuration text file contains mostly properties that relate to the package installation. The properties are fixed. The `Product.ini` file is read only upon installation of the SecuRemote/SecureClient.

To change `products.ini` use the Packaging Tool, or if necessary, edit the file manually as follows:

1. Extract `products.ini` from the original SecuRemote/SecureClient `tgz` format installation package.
2. Perform the required manual editing of `products.ini`.
3. Recreate the `tgz` file. This is the SecuRemote/SecureClient package for end-users.

Userc.C File Parameters

In This Section

SecureClient

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- `default_ps (n.n.n.n)` — Specifies the IP address of the default Policy Server. If this property exists, SecureClient will automatically log on to the Policy Server (with IP n.n.n.n) when it is launched, relieving the user of the need to manually log on to the Policy Server — *Global*.
- `manual_slan_control` (**true**, false) — Disabling this property will remove the **Disable Policy** menu items from the **Policy** menu — *Global*.
- `allow_clear_in_enc_domain` (true, **false**) — If enabled, unencrypted connections will be accepted by SecureClient NG over Encrypt desktop rules and by SecureClient 4.1 running **Encrypted Only** or **Outgoing and Encrypted** policy, as long as both the source and the destination IP addresses are in the encryption domain of a single security gateway — *Global*.
- `disable_stateful_dhcp` (true, **false**) — As long as this attribute is false, DHCP packets will be allowed by SecureClient regardless of the enforced Desktop Security policy. If you set this attribute to true, DHCP will be allowed only if the Desktop Security policy allows it explicitly. This requires SecureClient version 4.1 to run a policy of **Allow All** and SecureClient NG to have DHCP enabled through specific rules — *Global*.

- `block_conns_on_erase_passwords` (true, **false**) — If true, the **Close VPN** option will replace **Erase Password** in the SecureClient's **Passwords** menu and the button ■ will appear in the toolbar. Selecting **Close VPN** or clicking the above button will result in blocking all encrypted connections — *Managers*.
- `enable_automatic_policy_update` (true, **false**) — Specifies whether Automatic Policy Update is enabled or not — *Managers*.
- `silent_policy_update` (true, false) — If true, the client will not prompt the user to update the policy upon client startup, even if the time specified in automaic_policy_update_frequency has passed. The client will still attempt to update the policy after successful key exchange — *Managers*.
- `PS_HA` (**true**, false) — Use backup Policy Servers on logon failure — Managers.
- `PS_LB` (true, **false**) — If true will randomize policy server — list so not all clients will try to connect to the same policy server — *Managers*.
- `LB_default_PS` (true, **false**) – If true, when default_ps(x.x.x.x) is set it will go to a random Policy Server in the same site (found by examining topology) — *Managers*.
- `no_policy` (true, false) — Indicates disable policy state — *Global*.
- `policy_expire` (**60**) — Timeout of policy, in minutes. This property can also be controlled in SmartDashboard — *Managers*.
- `retry_frequency` (**30**) — If logged in to a Policy Server, but failed to re-logon after half the expiry time, this parameter (in seconds) specifies the amount of time before retrying logon. On each attempt all Policy Servers are tried — *Managers*.
- `automaic_policy_update_frequency` (**10080**) – Controls how frequently (in seconds) SecureClient should update policy files — *Managers*.
- `suppress_all_balloons` (true, **false**) - which controls all balloon messages. If the flag is set to true, no message balloons are displays. If false, all balloons are displayed. Note that the balloon's messages will still appear in the `.tde` files and will be logged in the Status Dialog's MessageViewer.
- `sdl_browse_cert` (true, **false**) - When set to false, the browse certificate in "change authentication" is disabled. When set to true, the browse dialog in SDL mode is restricted, you can only browse files, not create, change or lanuch applications.
- `disconnect_when_in_enc_domain` (**true**, false)- If the client is connected to a site, and an interface appears with an IP address located within one of the gateway's VPN domains, the client is disconnected. A message balloon explains why.

- open_full_diagnostic_tool (true, **false**) - When set to false, SC will open only log-view of diagnostic. When set to true, SC will open full diagnostic. In any case, the full diagnostic tool will open from the start menu.
- tt_failure_show_notification (true, **false**) - If fail_connect_on_tt_failure is false, (meaning that a connection will succeed even though tt failed) then a string notification of tt-failure will show in the connecton progress details because of this flag.
- simplified_client_route_all_traffic (true, **false**) - This attribute determines whether the Simplified Client performs connections using route-all-traffic or not.
- scv_allow_sr_clients (true, **false**)- If set to true, SecuRemote clients, which by default are not SCV verified, will send a verified state to the enforcing gateway.
- use_profile_ps_configuration (**true**, false) - Set to true to enable remote users to connect to one Gateway and logon to a Policy Server behind another gateway.
- force_route_all_in_profile (true, **false**) — If set to true, profiles created by the user will have the "route all traffic" option selected and grayed in the profile creation/edit dialog. - *Global*
- enable_mode_switching (**true**, fales) - If set to true, client has the option to switch between *Extended View* and *Compact View*.

Hot Spot Registration

- enabled (true, **false**) - Set to **true** to enable a user to perform Hotspot registration.
- log (true, **false**) - Set to **true** to send logs with the list of IP addresses and ports accessed during registration.
- connect_timeout (600) - Maximum number of seconds to complete registration.
- max_ip_count (5) - Maximum number of IP addresses allowed during registration.
- block_hotspot_after_connect (true, **false**) - If set to **true** upon successful connect, the recorded ports and addresses will not remain open.
- max_trials (0) - This value represents the maximum number of unsuccessful hotspot registration attempts that an end user may perform. Once this limit is reached, the user will not be allowed to attempt registration again. The counter

is reset upon reboot, or upon a successful VPN connect. In addition, if you modify the max_trials value, the modification will take affect only upon successful connect, or reboot.

If the max_trials value is set to 0, an unlimited number of trials is allowed.

- local subnets (true, **false**) - Restrict access to local subnets only.
- ports (80, 443, 8080) - Restrict access to specific ports.

Encryption

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- use_cert (true, **false**) – Specifies whether Use Certificate will be checked in the **IKE Authentication** window — *Global*.
- use_entelligence (**true**, false) – Specifies whether SecuRemote should attempt to use the Entrust Entelligence toolkit, if installed — *Global*.
- entrust_inifile — Full path to a non-default entrust.ini file, to be used by SecuRemote/SecureClient when working with entrust certificates — *Global*.
- certfile — Name of the last certificate used — *Global*.
- gettopo_port (**264**) — Which port to use for topology update — *Global*.
- pwd_erase_on_time_change (true, **false**) – Performs **Erase Passwords** when the user changes the system clock — *Global*.
- force_udp_encapsulation (true, false) – Indicates whether UDP encapsulation is used (transparent, and active profile in connect mode). Also used in Connect Mode to create the default profile — *Global*.
- support_tcp_ike (true, false) – Indicates whether TCP over IKE is used (transparent, and active profile in connect mode). Also used in Connect Mode to create the default profile — *Global*.
- support_tcp_ike (true/false/**use_site_default**) – Determine whether or not to attempt IKE over TCP — *Gateway*.
- support_ip_assignment (true, **false**) – Indicates whether Office Mode is used (transparent, and active profile in connect mode). Also used in connect mode to create the default profile — *Global*.

- ChangeUDPsport (**true**, false) – If the value of both flags ChangeUDPsport and force_udp_encapsulation is true, a random source port is used for IKE packets, and another random source port is used for UDP encapsulation packets — *Global*.
- uencapport (**2746**) — Specifies the port to be used on the UDP encapsulated packets when using UDP encapsulation — *Gateway*.
- ChangeIKEPort (**true**, false) – If true, do not bind to port 500. Instead, use router port and use address translation to make it seem as if the connection originated from port 500. This parameter allows other client applications (such as Nokia and Microsoft) to use that port. Note if the port is taken, another port will be used — *Global*.
- send_clear_traffic_between_encryption_domains (true, **false**) – if true and the source and the destination are behind encryption domains (not same domains), packets will be sent clear. This feature is enabled only if a single site is defined — *Managers*.
- send_clear_except_for_non_unique (**true**, false) – If true, send_clear_traffic_between_encryption_domains will not function for IP addresses which are defined as NAT private addresses.
- send_clear_except_for_specific_addresses (true, **false**)– If true, send_clear_traffic_between_encryption_domains will not function for IP addresses which are defined in send_clear_except_for_address_group — *Managers*.
- send_clear_except_for_address_group – Address group specification for send_clear_except_for_specific_addresses — *Managers*.
- dns_encrypt (**true**, false) — Overwrites the encrypting attribute received in the topology in the dnsinfo section. May be relevant for workarounds prior to version NG FP3 — Global.
- disable_split_dns_when_in_om (**true**, false) — Disable split DNS when in Office Mode — *Global*.
- disable_split_dns_when_disconnected (**true**, false) — Disable split DNS when disconnected — *Global*.
- disconnect_on_IKE_SA_expiry (true, **false**) – In connect mode, if the IKE timeout expires and this property is **true**, disconnect instead of erasing the passwords — *Global*.
- renew_users_ica_cert (**true**, false) – Specifies whether users be able to renew their certificates (explicitly or implicitly) — *Managers*.

- `renew_users_ica_cert_days_before (1-1000)` **60 —** How many days before expiration to start and perform an implicit renewal — *Managers*.
- `upgrade_fp1_and_below_users_ica_cert` (**true**, false) – Whether or not to implicitly renew certificates that were issued before NG FP2 — *Managers*.
- `ike_negotiation_timeout` (**36**) – Determines the maximum time in seconds that the IKE engine will wait for a response from the peer before timing out. This is the maximum interval between successive packets, and not the maximum negotiation lifetime — *Managers.*
- `phase2_proposal` (**large**, small) — Determines the size of the proposal sent by the client in Quick Mode, packet 1. This property is for backwards compatibility. NG FP3 and higher clients use phase2_proposal_size — Managers.
- `phase2_proposal_size` (**large**, small) — Determines the size of the proposal sent by NG FP3 or higher clients in Quick Mode, packet 1. If the value is missing the value of phase2_proposal is taken instead. NG FP3 clients will try a large proposal after a small proposal attempt fails — Managers.
- `vpn_peer_ls` (**true**, false) — In a MEP fully overlapping encryption domain configuration, if this property is TRUE, a gateway will be chosen randomly between the MEP gateways and will be given priority — Managers.
- `ike_support_dos_protection` (**true**, false) — Determines whether the client is willing to respond to a DoS protection request, by restarting Main Mode using a stateless protection. Equivalent to the SmartDashboard Global Property: Support IKE DoS Protection from unidentified Source — Managers.
- `sr_don't_check_crl` (**true**, false) — Do not check the CRL of the certificate — Managers.
- `crl_start_grace` (610200) — SecuRemote/SecureClient may accept CRLs that are not yet valid — Managers.
- `crl_end_grace` (1209600) — SecuRemote/SecureClient may accept CRLs that have recently expired — Managers.
- `site_default_tcp_ike` (**true**, false) — Determines the site default for attempting IKE over TCP. Each gateway has a property: “supports_tcp_ike” (true, false or use_site_default). If the value is set to 'use_site_default' then the management property site_default_tcp_ike is used by the client to determine whether to attempt IKE over TCP or not — Managers.

- `suppress_ike_keepalive` (**true**, false) — If the IPsec keepalive is turned on, and the value of the property “suppress_ike_keepalive” is false, empty UDP packets will be sent to the gateway (destination port 500). The UDP keepalive packets are sent only if there is an IKE SA with the peer and if UDP encapsulation was chosen — Managers.
- `default_phase1_dhgrp` — This field indicates which DH group to use for IKE phase 1 before the client has a topology. If the flag does not exist, group 2 will be used — Global.
- `to_expire` (**true**, false) — Whether or not to have a timeout for the phase2 IKE authentication. This property can also be controlled in SmartDashboard — Managers.
- `expire` (120) — Timeout of IKE phase2. This property can also be controlled in SmartDashboard — Managers.
- `ICA_ip_address` — The IP address of the Internal CA — Global.
- `allow_capi` (**true**, false) — Allow the disabling of CAPI storage to Internal CA registration — Global.
- `allow_p12` (**true**, false) — Allow the disabling of p12 file storage to Internal CA registration — Global.
- `trust_whole_certificate_chain` (**true**, false) — This attribute improve connectivity where there is a Certificate hierarchy, and the CA trusted by the gateway is a subordinate CA (not necessarily a direct subordinate) of the client trusted CA. Without this flag, both the gateway and the client must trust exactly the same CA — Global.
- `is_subnet_support` (**true**, false) — If turned on, IPsec SA will be valid for a subnet, otherwise it will be valid for a specific address — Gateway.
- `ISAKMP_hybrid_support` (**true**, false) — If turned on, when the authentication pop up appears, the user will have the option to choose between Hybrid mode and certificates as an authentication mode. (Otherwise the user will have the option to choose between certificates and pre-shared secret) — Gateway.
- `resolve_multiple_interfaces` (**true**, false) — If 'resolve_interface_ranges' (static interface resolving) is disabled or failed, and this property is turned on, then dynamic interface resolving will be done when addressing this gateway. In this case the interfaces of the gateway will be probed once — Gateway.
- `interface_resolving_ha` (**true**, false) — If dynamic interface resolving is used (see resolve_multiple_interfaces) and this property is turned on- the interfaces of the gateway will be probed per connection to see if they are live — Gateway.

- `isakmp.ipcomp_support` (**true**, false) — If the peer gateway is a least NG and the client is SecureClient (and not SecuRemote) then: — If the client is in “send small proposal” mode and this property is turned on then IP compression will be proposed. (If the client is in “send large proposal” mode then IP compression will be offered regardless of the value of this property) — Gateway.
- `supports_tcp_ike` (use_site_default) — If IKE over TCP is configured on the client AND either this property is 'true' or it's 'use_site_default' and site_default_tcp_ike is 'true', then IKE phase 1 will be done over TCP — Gateway.
- `supportSRIkeMM` (**true**, false) — When the authentication method is PKI, if this property is false, Main mode is not supported — Gateway.

Multiple Entry Point

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See “Structure of Userc.C” on page 608. Do not manually edit Global properties.

`resolver_ttl` (10) — Specifies how many seconds SecuRemote will wait before deciding that a gateway is down — Global.

`active_resolver` (true, false) — Specifies whether SecuRemote should periodically check the gateway status. Active gateway resolving may cause the dial-up connection to try to connect to an ISP. Turning this property off will avoid problems associated with this behavior — Global.

`resolver_session_interval` (30) — Specifies for how many seconds the gateway status (up or down) remains valid — Global, Managers.

Encrypted Back Connections

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- `keep_alive` (true, false) — Specifies whether the security gateway will maintain session key information for the Client, to allow encrypted back connections at any time. This property can also be controlled in SmartDashboard — Global, Managers.
- `keep_alive_interval` (20) — When keep_alive is true, SecuRemote will ping the security gateway every n seconds, where n is the number specified by the keep_alive_interval property. This property can also be controlled in SmartDashboard — Global.

Topology

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- `topology_over_IKE` (**true**, false) — Specifies whether New Site in SecuRemote will use IKE to authenticate the user. If this property is set to true, IKE will be used, either using Hybrid Authentication (i.e., any authentication method chosen in the Authentication tab of the user properties) or using certificates. If this property is set to False, SSL will be used (as in version 4.1), and users will need IKE pre-shared secret or certificate configured to define a new site — Global, Managers.
- `encrypt_db` (**true**, false) — Specifies whether the topology information in userc.C is maintained in encrypted format — Global.
- `silent_topo_update` (**true**, false) — Used for backwards compatibility, when working with servers that do not pass the property per site. This property can also be controlled in SmartDashboard — Global, Managers.
- `silent_update_on_connect` (**true**, false) — Tries to perform an update with the gateway to which a connection is being attempted, before connecting (applies to Nokia clients) — Global.
- `update_topo_at_start` (**true**, false) — If the timeout expires, update the topology upon start up of the SecuRemote/SecureClient GUI application — Global, Managers.

NT Domain Support

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- `no_clear_tables` (**true**, false) — Setting this property to true will enable the opening of new encrypted connections with the Encryption Domain after SecuRemote/SecureClient has been closed by logoff or shutdown, as long as encryption keys have been exchanged, and are still valid. This may be necessary when using a Roaming Profile with NT domains, since the PC tries to save the user's profile on the Domain Controller during logoff and shutdown, after SecuRemote/SecureClient has been closed by Windows. This feature should be used in conjunction with "keep_alive" (see "Encrypted Back Connections" on page 618), to ensure that valid encryption keys exist at all times — Global.
- `connect_domain_logon` (**true**, false) — Global. Setting this attribute to true enables clients using Connect Mode to log on to a Domain Controller via SDL. The user should do the following in order to logon to the Domain Controller:

  1. Log on to the local Windows machine.
  2. Connect to the organization.
  3. Logoff and log back on (within five minutes after logoff) to the protected Domain Controller, using the encrypted connection.

  Note -
  1. Enabling this setting will keep the client Connected to the organization for five minutes after the user logs off Windows.

  2. This feature was introduced before SDL in connect mode in was supported in NG FP2 HF2. In versions where SDL is supported, this property is used only for domain roaming profile support.

- `sdl_main_timeout` (60000) — In connect mode this property specifies the amount of time to wait for user to successfully connect or cancel the connect dialog — Global.

Miscellaneous

Note - Bold indicates the default value. *Global*, *Managers*, or *Gateway* indicates the location in Userc.C. See "Structure of Userc.C" on page 608. Do not manually edit Global properties.

- `enable_kill` (**true**, false) — Specifies whether the user can Stop SecuRemote/SecureClient. If this option is set to false, Stop VPN-1 SecuRemote or Stop VPN-1 SecureClient does not appear in the File menu or when right-clicking on the system tray icon — Global.
- `use_ext_auth_msg` (**true**, false) — Specifies whether SecuRemote/SecureClient will show custom messages in the authentication window upon success or failure. The messages should be placed in a file named AuthMsg.txt located in the SecuRemote directory (typically in Program Files\CheckPoint). See the AuthMsg.txt file in the SecuRemote package for more details — Global.
- `use_ext_logo_bitmap` (**true**, false) — Specifies whether SecuRemote/SecureClient will show a custom bitmap in the authentication window. The file should be named logo.bmp and should be placed in the SecuRemote directory (usually located under Program Files\CheckPoint) — Global.
- `guilibs` — Used to specify SAA DLL, and is documented in this context — Global.
- `pwd_type` (now, later) — Used internally to indicates now or later auth dialog state. Do not modify — Global.
- `connect_mode_erase_pwd_after_update` (true, false) — Erase password after a site update in Connect Mode. Used with silent_update_on_connect — Global.
- `disable_mode_transition` (**true**, false) — Do not enable user to switch between modes via GUI or command line — Global.
- `connect_api_support` (**true**, false) — Indicates SecuRemote/SecureClient mode.Set to true in order to work with the Connect API — Global.
- `connect_mode` — Indicates SecuRemote/SecureClient mode. True for connect mode — Global.
- `allow_clear_traffic_while_disconnected` (**true**, false) — Topology is not loaded when disconnected, ensuring that there are no popups on the LAN when disconnected — Global.
- `stop_connect_when_silent_update_fails` (**true**, false) — If trying to connect in silent_update_on_connect mode, and the topology update fails, the connection will fail — Global.

- `go_online_days_before_expiry` (0) — The number of days before Entrust automatic key rollover (certificate renewal). Zero equals never — Global.
- `go_online_always` (**true**, false) — When true, will attempt the LDAP (entrust.ini) protocol after successful IKE negotiation — Global.
- `implicit_disconnect_threshold` (900) — When losing connectivity on physical adapter, SecuRemote/SecureClient keeps the connected state for the amount of time (in seconds) specified by implicit_disconnect_threshold. If the time elapses, or if connectivity resumes with a different IP address, SecuRemote/SecureClient disconnects. This is useful in network environments with frequent network disconnection, such as wireless — Global.
- `active_test` — Active tests configuration — Global.
- `log_all_blocked_connections` — Used internally to indicates the mode, and reflects the state of the GUI checkbox. Do not modify — Global.
- `cache_password` — Used internally to save the state of the checkbox called "Remember password, as per site settings". Do not modify — Global.
- `dns_xlate` (**true**, false) — Turn off the split DNS feature. May be needed in versions prior to NG FP3. In later versions, split DNS is not used by default when in Office Mode — Global.
- `FTP_NL_enforce` (0, 1, 2) — Indicates the strictness of the FTP inspection (0 -no check, 1- default check: Multiple newline characters allowed, 2-strict check: no multiple newline characters allowed — Global.
- `show_disabled_profiles` (**true**, false) — In connect mode, if the IKE timeout expires and this property is TRUE, disconnect instead of erasing the passwords — Global.
- `post_connect_script` — Specify full path for a script that SecuRemote/SecureClient will run after a connection has been established (Connect Mode only) — Managers.
- `post_connect_script_show_window` (**true**, false) — Specifies whether or not the post-connect script will run in a hidden window — Managers.
- `list_style` — How the site icons are presented in the main frame window — Global.
- `mac_xlate` (**true**, false) — Needs to be set to true to support Split DNS where traffic to the "real" DNS server may not be routed the same way as traffic to the "split" DNS server. The most common scenario is "real" DNS server on the same subnet as the client. Split DNS modifies the IP destination of the packet, but not the MAC destination. With mac_xlate set to true, the MAC destination address is set to the address of the default gateway — Global.

- `mac_xlate_interval` — How frequently a check is made for the default gateway's MAC address (see mac_xlate) — Global.
- `sda_implicit` (**true**, false) — The working mode of the Software Distribution Agent (SDA). True = implicit, false = explicit — Global, Managers.
- `sda_imlicit_frequency` — The frequency (in minutes) with which the Software Distribution Agent (SDA) connects to ASD server to check for updates — Global, Managers.
- `sr_build_number`, `sr_sw_url_path`, `sr_sw_url_path_9x`, `sr_build_number_9x`, `sr_sw_url_path_nt`,`sr_build_number_nt`, `sr_sw_url_path_w2k`,`sr_build_number_w2k` — On the Security Management server machine, the names are desktop_sw_version, desktop_build_number, etc. These attributes help SecureClient decide if it needs to upgrade itself — Managers.
- `install_id_nt`,`install_id_9x`,`install_id_w2k` — Installation IDs — Managers.

Product.ini Parameters

Table 29-1 Parameters for Product.ini

Parameter (bold indicates the default)	Meaning
`OverwriteConfiguration=`**0**`/1`	Sets the value for **Update** or **Overwrite** choice during upgrade. The default value (0) means **Update** is chosen.
`ShowUpdateOverwrite=0/`**1**	Show the **Update** or **Overwrite** window to the user during installation. If the window is not shown to the user, the value placed in `OverwriteConfiguration` will be used.
`PathAskUser=0/`**1**	Show the **Choose Installation Destination** window to the user during installation. If the window is not shown to the user, the default value chosen by InstallShield will be used (usually this will be `C:\Program Files\CheckPoint\SecuRemote`).
`DesktopSecurityAskUser=0/`**1**	Show the **Desktop Security** window to the user during installation. If the window is not shown to the user, the value placed in `DesktopSecurityDefault` will be used.
`DesktopSecurityDefault=0/`**1**	Sets the value for Desktop Security installation. A value of 1 means that SecureClient will be installed, while a value of 0 means that SecuRemote will be installed.
`InstallDialupOnly=`**0**`/1`	Sets the value for binding to All Adapters or to Dialup Adapters only. A value of 0 means that the installation will bind to All Adapters.
`ShowNetworkBindings=0/`**1**	Show the Adapter Bindings window to the user during installation. If the window is not shown to the user, the value placed in `InstallDialupOnly` will be used.
`ShowReadmeFile=0/`**1**	Show the **Readme** window to the user - this window asks the user whether he/she would like to view the readme file before finishing the installation. A value of 0 means that the window will not be shown to the user, and the `readme` file will not be read during installation.
`ShowBackgroundImage=0/`**1**	Determine whether the background image will be displayed during installation.
`ShowSetupInfoDialogs=0/`**1**	Determine whether informative InstallShield dialogs (which require no user interaction) will be displayed.
`DisableCancelInstall=`**0**`/1`	An option to disable the Cancel operation from the installation dialogs.

Table 29-1 Parameters for Product.ini

Parameter (bold indicates the default)	Meaning
`ShowRestart=0/`**1**	Determine whether Do you want to restart dialog will be shown.
`RestartAfterInstall=0/`**1**	0 - Do no restart after installation, 1- Restart after installation.
`ShowRebootWarning=0/`**1**	Suppress the message "The installation will complete after reboot".
`IncludeBrandingFiles=`**0**`/1`	Determines whether the files `authmsg.txt` and `logo.bmp` (used for customizing the **Authentication** dialog) will be copied during installation. See the `userc.C` options section for more details on `use_ext_auth_msg` and `use_ext_logo_bitmap`.
`EnableSDL=`**0**`/1`	Sets the value of **Secure Domain Logon (SDL)** during installation. If the value is 1, SDL will be enabled during installation.
`SdlNetlogonTimeout (Seconds/`**0**`)`	Set timeout for the operating system Net Logon, if 0 do not change the current value.
`Support3rdPartyGina=`**0**`/1`	SecuRemote Client NG allows using third party GINA DLLs for authentication. If this property is not selected, the Windows `GINA DLL` will be used by default. Enabling this property may conflict with SDL operation if a third party `GINA DLL` is used.
`EnablePolicyView=0/`**1**	Enable the Policy View in the SecureClient Diagnostics application.
`EnableLogView=0/`**1**	Enable the Log View in the SecureClient Diagnostics application.
`EnableDiagnosticsView=0/`**1**	Enable the Diagnostics View in the SecureClient Diagnostics application.
`ShowKernelInstallation=0/`**1**	Determines whether or not the driver installation dialog is displayed.

Table 29-1 Parameters for Product.ini

Parameter (bold indicates the default)	**Meaning**
`OverwriteEntINI=`**0**`/1`	Determines whether existing entrust.ini files will be overwritten by the `entrust.ini` files in the installation. A value of 1 indicates that the existing `entrust.ini` file will be overwritten.
`DefaultPath` (**Full path**)	Default: C:\Program Files\CheckPoint\SecuRemote.
`ConnectMode=`**0**/1	Set default client mode: 0 - transparent, 1- connect mode.

Chapter 30

SSL Network Extender

In This Document:

Introduction to the SSL Network Extender

Whenever users access the organization from remote locations, it is essential that not only the usual requirements of secure connectivity be met but also the special demands of remote clients. These requirements include:

- Connectivity: The remote client must be able to access the organization from various locations, even if behind a NATing device, Proxy or Firewall. The range of applications available must include web applications, mail, file shares, and other more specialized applications required to meet corporate needs.
- Secure connectivity: Guaranteed by the combination of authentication, confidentiality and data integrity for every connection.
- Usability: Installation must be easy. No configuration should be required as a result of network modification. The given solution should be seamless for the connecting user.

To resolve these issues, a secure connectivity framework is needed to ensure that remote access to the corporate network is securely enabled.

The SSL (Secure Socket Layer) Network Extender is a simple-to-implement remote access solution. A thin client is installed on the user's machine. (The SSL Network Extender client has a much smaller size than other clients.) It is connected to an SSL enabled web server that is part of the Enforcement Module. By default, the SSL enabled web server is disabled. It is activated by using the SmartDashboard, thus enabling full secure IP connectivity over SSL. The SSL Network Extender requires a server side configuration only, unlike other remote access clients. Once the end user has connected to a server, the thin client is downloaded as an ActiveX component, installed, and then used to connect to the corporate network using the SSL protocol.

It is much easier to deploy a new version of the SSL Network Extender client than it is to deploy a new version of other conventional clients.

How the SSL Network Extender Works

The SSL Network Extender solution comprises a thin client installed on the user’s Desktop/Laptop and an SSL enabled web server component, integrated into the Security gateway.

To enable connectivity for clients using the SSL Network Extender - a security gateway must be configured to support SecuRemote/SecureClient, in addition to a minor configuration specific to SSL Network Extender.

The SSL Network Extender may be installed on the user's machine by downloading it from a gateway, R55 HFA10 (or higher).

Commonly Used Concepts

This section briefly describes commonly used concepts that you will encounter when dealing with the SSL Network Extender. It is strongly recommended that you review the “Remote Access VPN” section of this book before reading this guide.

In This Section:

Remote Access VPN

Refers to remote users accessing the network with client software such as SecuRemote/SecureClient, SSL clients, or third party IPSec clients. The security gateway provides a *Remote Access Service* to the remote clients.

Remote Access Community

A Remote Access Community, a Check Point concept, is a type of VPN community created specifically for users that usually work from remote locations, outside of the corporate LAN.

Office Mode

Office Mode is a Check Point remote access VPN solution feature. It enables a security gateway to assign a remote client an IP address. This IP address is used only internally for secure encapsulated communication with the home network, and therefore is not visible in the public network. The assignment takes place once the user connects and authenticates. The assignment lease is renewed as long as the user is connected. The address may be taken either from a general IP address pool, or from an IP address pool specified per user group, using a configuration file.

Visitor Mode

Visitor Mode is a Check Point remote access VPN solution feature. It enables tunneling of *all* client-to-gateway communication through a regular TCP connection on port **443**. Visitor mode is designed as a solution for firewalls and Proxy servers that are configured to block IPsec connectivity.

Endpoint Security on Demand

Endpoint Security on demand (ESOD) may be used to scan endpoint computers for potentially harmful software before allowing them to access the internal application. When end users access the SSL Network Extender for the first time, they are prompted to download an ActiveX component that scans the end user machine for Malware. The scan results are presented both to the gateway and to the end user. SSL Network Extender access is granted/denied to the end user based on the compliance options set by the administrator.

ESOD Policy per User Group

Since there are many different kinds of threats to your network's security, different users may require different configurations in order to guard against the increasing number and variety of threats. The ability to configure a variety of ESOD policies enables the administrator to customize the software screening process between different user groups.

Screened Software Types

ESOD can screen for the Malware software types listed in the following table:

Table 30-1 Screened Software Types

Software Type	Description
Worms	Programs that replicate over a computer network for the purpose of disrupting network communications or damaging software or data.
Trojan horses	Malicious programs that masquerade as harmless applications.
Hacker tools	Tools that facilitate a hacker's access to a computer and/or the extraction of data from that computer.

Table 30-1 Screened Software Types

Software Type	Description
Keystroke loggers	Programs that record user input activity (that is, mouse or keyboard use) with or without the user’s consent. Some keystroke loggers transmit the recorded information to third parties.
Adware	Programs that display advertisements, or records information about Web use habits and store it or forward it to marketers or advertisers without the user’s authorization or knowledge.
Browser plug-ins	Programs that change settings in the user's browser or adds functionality to the browser. Some browser plug-ins change the default search page to a pay-per-search site, change the user's home page, or transmit the browser history to a third party.
Dialers	Programs that change the user’s dialup connection settings so that instead of connecting to a local Internet Service Provider, the user connects to a different network, usually a toll number or international phone number.
3rd party cookies	Cookies that are used to deliver information about the user’s Internet activity to marketers.
Other undesirable software	Any unsolicited software that secretly performs undesirable actions on a user's computer and does not fit any of the above descriptions.

Special Considerations for the SSL Network Extender

This section lists SSL Network Extender special considerations, i.e. pre-requisites, features and limitations:

In This Section:

Pre-Requisites

The SSL Network Extender pre-requisites are listed below:

Client-side Pre-Requisites

The SSL Network Extender client-side pre-requisites are listed below:

- Remote client must be running the following:
  - Windows 2000 Pro
  - Windows XP Home Edition and Pro
  - Windows Vista
  - Linux RHEL 3.0
  - Linux Suse 9 and up
  - Red Hat Linux 7.3
  - Mac OSX Tiger
- Remote client must use the following. Each must allow ActiveX or Java Applet.
  - Internet Explorer version 5.0 or higher
  - FireFox
  - Safari
- First time client installation, uninstall and upgrade requires administrator privileges on the client computer.

Server-Side Pre-Requisites

The SSL Network Extender server-side pre-requisites are listed below:

- The SSL Network Extender is a server side component, which is part of a specific Enforcement Module, with which the SSL Network Extender is associated. It may be enabled on the gateway, already configured to serve as a Remote Access SecureClient gateway.
- The specific security gateway must be configured as a member of the Remote Access Community, and configured to work with Visitor Mode. This will not interfere with SecureClient functionality, but will allow SecureClient users to utilize Visitor Mode.
- The same access rules are configured for both SecureClient and SSL Network Extender users.
- If you want to use Endpoint Security on Demand, you should install the ESOD server or the ESOD configuration tool. Customers can download the ESOD server from http://www.checkpoint.com/products/clientless/index.html along with its documentation.

Features

The SSL Network Extender features are listed below:

- Easy installation and deployment.
- Intuitive and easy interface for configuration and use.
- The SSL Network Extender mechanism is based on Visitor Mode and Office Mode.
- Automatic proxy detection is implemented.
- Small size client: Download size of SSL Network Extender package < 400K; after installation, size of SSL Network Extender on disk is approximately 650K.
- All security gateway authentication schemes are supported: Authentication can be performed using a certificate, Check Point password or external user databases, such as SecurID, LDAP, RADIUS and so forth.
- At the end of the session, no information about the user or gateway remains on the client machine.
- Extensive logging capability, on the gateway, identical to that in VPN-1 SecuRemote/SecureClient.
- High Availability Clusters and Failover are supported.

- SSL Network Extender Upgrade is supported.
- The SSL Network Extender supports the RC4 encryption method.
- Users can authenticate using certificates issued by any trusted CA that is defined as such by the system administrator in SmartDashboard.
- SSL Network Extender is now supported on IPSO.
- Endpoint Security on Demand prevents threats posed by Malware types, such as Worms, Trojan horses, Hacker's tools, Key loggers, Browser plug-ins, Adwares, Third party cookies, and so forth.
- SSL Network Extender can be configured to work in Hub Mode. VPN routing for remote access clients is enabled via Hub Mode. In Hub mode, all traffic is directed through a central Hub.

Configuring the SSL Network Extender

The following sections describe how to configure the server. Load Sharing Cluster Support, customizing the Web GUI, upgrading the SSL Network Extender client and Installation for Users without Administrator privileges are also discussed.

In This Section:

Configuring the Server

Before configuring the server, verify that you have a valid license for the SSL Network Extender.

Use `cpconfig` to verify that you have a valid license for the SSL Network Extender. Check Point software is activated with a License Key. You can obtain this License Key by registering the Certificate Key that appears on the back of the software media pack, in the Check Point Support Center:

http://support.checkpoint.com

Server-Side Configuration

The SSL Network Extender requires only server side configuration

In This Section:

Configuring the Gateway as a Member of the Remote Access Community

1. Open SmartDashboard, select the gateway object on the Network Object tab of the Objects Tree. The **General Properties** window is displayed.
2. Verify that **VPN** is selected and click **OK**.
3. Select **VPN** in the objects tree on the left hand side.
4. Verify that the module participates in the Remote Access Community. If not, add the module to the Remote Access Community.
5. In the **Topology Tab** of the **Gateway Properties** page, configure the VPN Domain for SSL Network Extender, in the same way that you configure it for SecureClient

Note - You can use the VPN Domain to configure SSL Network Extender to work in Hub Mode. All traffic is then directed through a central Hub. You can also use the "Set domain for Remote Access Community ..." button on the same tab to create different encryption domain for Remote Access clients that connect to the gateway (see "Configuring Selective Routing" on page 341).

6. Configure Visitor Mode, as described in the "Resolving Connectivity Issues" chapter. Configuring Visitor Mode doesn't interfere with regular SecureClient users' functionality. It merely allows SecureClient users to enable Visitor Mode. (For a description of Visitor Mode, refer to "Visitor Mode" on page 631.)

Note - The SSL Network Extender uses TCP 443 (SSL) to establish a secure connection with VPN SecurePlatform and the Nokia platform use TCP 443 (SSL) for remote administration purposes. Another port may be assigned to the SSL Network Extender, however, this is not recommended, as most proxies do not allow ports other than 80 and 443. Instead, it is strongly recommended that you assign the SecurePlatform, or Nokia platform web user interface to a port other than 443.

7. If you are working with SecurePlatform, you may perform the following actions:

- You can change the webui port, by running the following command:

  `webui enable <port number>` (for example, `webui enable 444`)

- You can disable the webui completely, by running the following command:

  `webui disable`

8. To change a Voyager port on Nokia platform, run:

   `voyager -e x -S <port number>` (x represents the encryption level.)

   For more information, run: `voyager -h`

9. Select **Remote Access > Office Mode**.

10. Configure Office Mode, as described in the "Office Mode" chapter. (For a description of Office Mode, refer to "Office Mode" on page 630.)

Note - Office Mode support is mandatory on the gateway side.

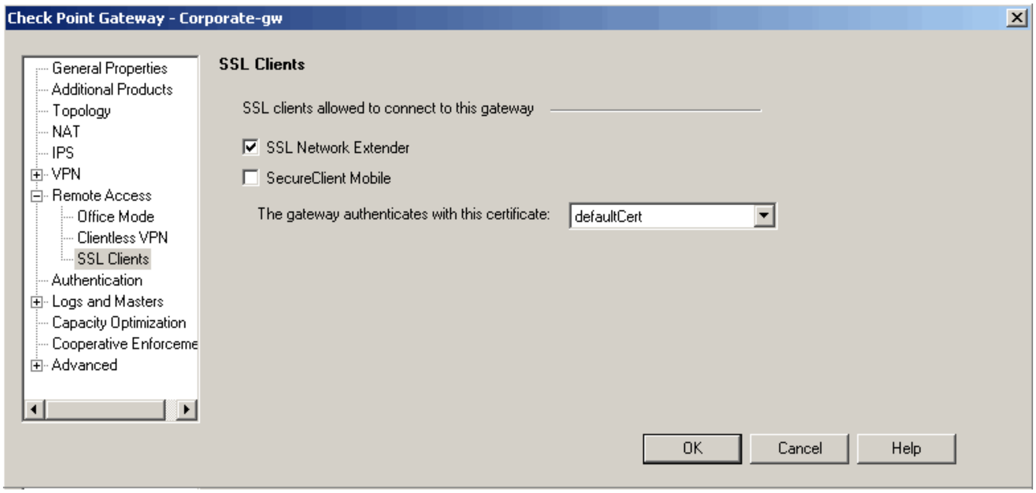
11. Configure Users and Authentication.

Configuring the Gateway to Support the SSL Network Extender

To configure the SSL Network Extender:

Note - You must configure each gateway that will be using the SSL Network Extender.

1. Select **Remote Access > SSL Network Extender**. The **SSL Network Extender** window is displayed.



SSL Clients

SSL clients allowed to connect to this gateway

☑ SSL Network Extender

□ SecureClient Mobile

The gateway authenticates with this certificate: defaultCert

2. Activate **Support SSL Network Extender**.

3. Select the server side certificate with which the gateway will authenticate from the drop-down list.

4. Click **OK**.

Configuring the SSL Network Extender

1. Select **Policy > Global Properties > Remote Access > SSL Network Extender**. The **SSL Network Extender Global Properties** window is displayed.
2. Select the user authentication method, employed by the SSL Network Extender, from the drop-down list. The options are:
   - **Certificate:** The system will authenticate the user *only* via a certificate. Enrollment is not allowed.
   - **Certificate with enrollment:** The system will authenticate the user *only* via a certificate. Enrollment is allowed. If the user does not have a certificate, he/she can enroll using a registration key, received previously from the system administrator.
   - **Legacy:** (Default) The system authenticates the user via his/her **Username** and **Password**.
   - **Mixed:** The system attempts to authenticate the user via a certificate. If the user does not have a valid certificate, the system attempts to authenticate the user via his/her **Username** and **Password**.

Management of Internal CA Certificates

If the administrator has configured **Certificate with Enrollment** as the user authentication scheme, the user can create a certificate for his/her use, by using a registration key, provided by the system administrator.

To create a user certificate for enrollment:

1. Follow the procedure described in “The Internal Certificate Authority (ICA) and the ICA Management Tool” in the *Security Management Server Administration Guide.*

Note - In this version, enrollment to an External CA is not supported.

2. Browse to the ICA Management Tool site, `https://<mngmt IP>:18265`, and select **Create Certificates**.

3. Enter the user’s name, and click **Initiate** to receive a Registration Key, and send it to the user.

   When the user attempts to connect to the SSL Network Extender, without having a certificate, the **Enrollment** window is displayed, and he/she can create a certificate for his/her use by entering the Registration Key, received from the system administrator.

   For a description of the user login experience, refer to “Downloading and Connecting the Client”.

Note - The system administrator can direct the user to the URL, `http://<IP>/registration.html,` to allow the user to receive a Registration Key and create a certificate, even if they do not wish to use the SSL Network Extender, at this time.

3. You can determine whether the SSL Network Extender will be upgraded automatically, or not. Select the client upgrade mode from the drop-down list. The options are:

   - **Do not upgrade:** Users of older versions will not be prompted to upgrade.
   - **Ask user:** (Default) Ask user whether or not to upgrade, when the user connects.
   - **Force upgrade:** Every user, whether users of older versions or new users will download and install the newest SSL Network Extender version.

Note - The Force Upgrade option should only be used in cases where the system administrator is sure that all the users have administrator privileges. Otherwise, the user will not be able to connect to and use the SSL Network Extender.

   For a description of the user upgrade experience, refer to “Downloading and Connecting the Client”.

4. You can determine whether the SSL Network Extender client will support the RC4 encryption method, as well as 3DES. (RC4 is a faster encryption method.) Select the supported encryption method from the drop-down list. The options are:

   - **3DES only:** (Default) The SSL Network Extender client supports 3DES, only.
   - **3DES or RC4:** The SSL Network Extender client supports the RC4 encryption method, as well as 3DES.

5. You can determine whether the SSL Network Extender will be uninstalled automatically, when the user disconnects. Select the desired option from the drop-down list. The options are:

- **Keep installed:** (Default) Do not uninstall. If the user wishes to uninstall the SSL Network Extender, he/she can do so manually.
- **Ask user whether to uninstall:** Ask user whether or not to uninstall, when the user disconnects.
- **Force uninstall:** Always uninstall automatically, when the user disconnects.

For a description of the user disconnect experience, refer to "Uninstall on Disconnect".

Note - The Uninstall on Disconnect feature will not ask the user whether or not to uninstall, and will not uninstall the SSL Network Extender, if a user has entered a suspend/hibernate state, while he/she was connected.

6. You can determine whether Endpoint Security on Demand will be activated, or not. When ESOD is activated, users attempting to connect to the SSL Network Extender will be required to successfully undergo an ESOD scan before being allowed to access the SSL Network Extender. Select the desired option from the drop-down list. The options are:
    - None
    - Endpoint Security on Demand

Fetching the xml Configuration File

After installing the ESOD server and configuring it, you must fetch the xml config file from the ESOD server by performing the following steps:

1. Open a browser on any machine.
2. Browse to `http://<site ip>/<site name or virtual directory>/sre/report.asp` and save the displayed XML file to disk, using **Save As**.
3. Copy the XML file to `$FWDIR/conf/extender/request.xml` on the gateway.

Upgrading ESOD

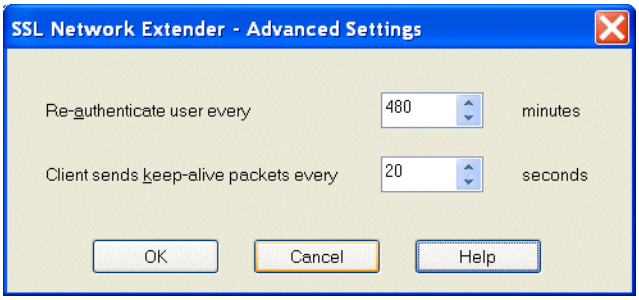
Note - At present, the Dynamic ESOD Update feature is not supported.

You can manually upgrade ESOD as follows:

1. Replace the `ICSScanner.cab` file, under `$FWDIR/conf/extender`, with the new package.
2. Edit the file `ics.html`, under `$FWDIR/conf/extender, as follows:`
   i. Search for `#Version=` and replace the current value with the new version.
   ii. Save.

7. Click **Advanced**. The **SSL Network Extender Advanced Settings** window is displayed.

Figure 30-1 SSL Network Extender Advanced Settings window



SSL Network Extender - Advanced Settings

Re-authenticate user every 480 minutes

Client sends keep-alive packets every 20 seconds

8. Configure the Session Timeout period. Once authenticated, remote users are assigned an SSL Network Extender *session*. The session provides the context in which the SSL Network Extender processes all subsequent requests until the user logs out, or the session ends due to a time-out.

Note - The default value is 8 hours. The minimum is 10 minutes, and the maximum is 24 hours.

Five minutes before the specified session time (timeout) has elapsed, the user may be prompted for his/her credentials, depending upon authentication settings, and once the credentials are accepted, the timeout interval is initialized. If the user has not provided credentials before the timeout has elapsed, the user is disconnected from the server and will need to reconnect the client manually.

9. Configure the keep-alive packets transmission frequency. The keep-alive packets inform NAT devices or HTTP proxies, via which the user is connected, that the user connection is still active.
10. Click **OK**. The **SSL Network Extender Global Properties** window is displayed.
11. Click **OK**.

Configuring ESOD Policies

On the Security Management server:

Note - Make sure that Endpoint Security on Demand is enabled in the **Global Properties > Remote Access > SSL Network Extender** page.

1. Navigate to the `$FWDIR/lib` directory.
2. Backup the `vpn_table.def` file.
3. Change the file name `vpn_table_HFA.def` to `vpn_table.def`.

On the security gateway:

1. Using the ESOD server, or ESOD configuration Tool (which can be downloaded from the Check Point download center), create xml policy files for each group and place them in `$FWDIR/conf/extender`.
2. You can create a default policy file, named request.xml. This is only optional, and will be used when no group is given.

3. In the $FWDIR/conf folder, create a file called ics.group. This should be a text file, in which, each row lists a group name and its policy xml file.

   Example of ics.group file:

   ```
   Group1  group1.xml

   Group2  group2.xml

   Group3  defGroup.xml

   Group4  defGroup.xml
   ```

   Important notes about the ics.group file:

   - The group name must be the same as its name in SmartDashboard.
   - Several groups can register to the same xml file.
   - Each group must appear only once in the ics.group file.
   - Only groups that are listed in the ics.group file will use their specific xml files. Groups that are not listed in the ics.group file will try to use the default policy, located in the request.xml file. If the request.xml file does not exist, an error will be returned.
   - The default xml file, request.xml, cannot appear in the ics.group file.

4. After creating the ics.group file (or after any change has been made), install policy.
5. Run cpstop and then cpstart on the security gateway.
6. Each user should be assigned the specific URL that matches his group. The URL should be in the format: https://hostIP/<groupName>_ics.html

   For example, all users belonging to "group1" will surf to the assigned URL: https://10.10.10.10/group1_ics.html.

For troubleshooting tips, see "Troubleshooting" on page 672.

Load Sharing Cluster Support

The SSL Network Extender provides Load Sharing Cluster Support.

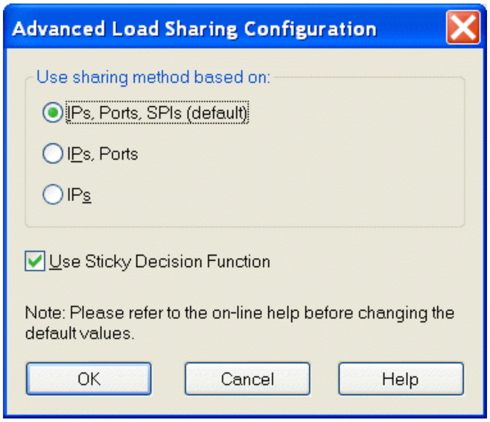
To provide Load Sharing Cluster Support:

1. Double-click the **Gateway Cluster Object** on the **Network Object** tab of the Objects Tree. The **Gateway Cluster Properties** window is displayed.

Note - A Load Sharing Cluster must have been created before you can configure use of sticky decision function.

2. Select **Cluster XL.** The **Cluster XL** tab is displayed.
3. Click **Advanced**. The **Advanced Load Sharing Configuration** window is displayed.

Figure 30-2 Advanced Load Sharing Configuration window



Advanced Load Sharing Configuration

Use sharing method based on:

● IPs, Ports, SPIs (default)

○ IPs, Ports

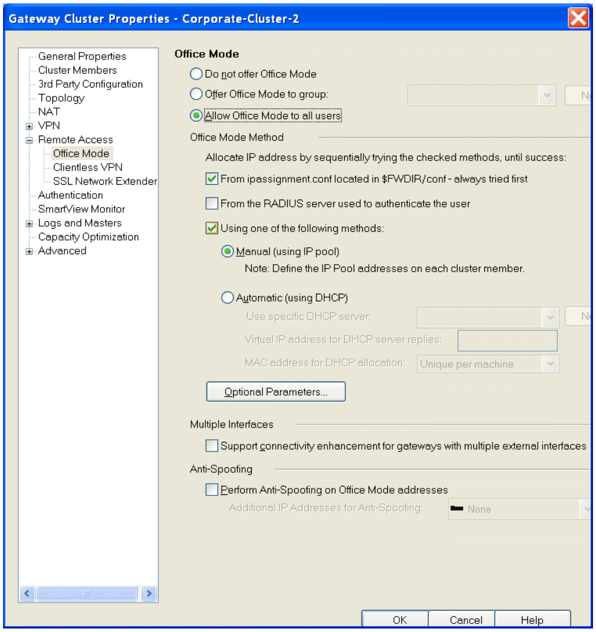
○ IPs

☑ Use Sticky Decision Function

Note: Please refer to the on-line help before changing the default values.

4. Select **Use Sticky Decision Function**. When the client connects to the cluster, all its traffic will pass through a single gateway. If that member gateway fails, the client will reconnect transparently to another cluster member and resume its session.
5. Select **Gateway Cluster Object > Remote Access > Office Mode.** When defining Office Mode, for use with Load Sharing Clusters, only the **Manual (using IP pool)** method is supported.

Figure 30-3 Advanced Load Sharing Configuration window



Office Mode

○ Do not offer Office Mode

○ Offer Office Mode to group:

● Allow Office Mode to all users

Office Mode Method

Allocate IP address by sequentially trying the checked methods, until success:

☑ From ipassignment.conf located in $FWDIR/conf - always tried first

☐ From the RADIUS server used to authenticate the user

☑ Using one of the following methods:

● Manual (using IP pool)

Note: Define the IP Pool addresses on each cluster member.

○ Automatic (using DHCP)

Use specific DHCP server:

Virtual IP address for DHCP server replies:

MAC address for DHCP allocation: Unique per machine

Optional Parameters...

Multiple Interfaces

☐ Support connectivity enhancement for gateways with multiple external interfaces

Anti-Spoofing

☐ Perform Anti-Spoofing on Office Mode addresses

Additional IP Addresses for Anti-Spoofing: None

Customizing the SSL Network Extender Portal

You can modify the SSL Network Extender Portal by changing skins and languages.

Configuring the Skins Option

To configure the Skins Option:

The `skin` directory is located under `$FWDIR/conf/extender` on the SSL Network Extender gateways.

There are two subdirectories. They are:

- `chkp`: contains skins that Check Point provides by default. At upgrade, this subdirectory may be overwritten.
- `custom`: contains skins defined by the customer. If `custom` does not exist yet, create it. At upgrade, this subdirectory is not overwritten. New skins are added in this subdirectory.

Disabling a Skin

1. Enter the specific skin subdirectory, under `custom`, that is to be disabled and create a file named `disable`. This file may be empty.
2. If the specific skin does not exist under `custom`, create it and then create a file within it named `disable`.
3. Install Policy. The next time that the user connects to the SSL Network Extender portal, this skin will not be available to him/her.

Example

```
cd $FWDIR/conf/extender/skin/custom
mkdir skin1
touch disable
```

Install Policy.

Creating a Skin

1. Enter the `custom` subdirectory.
2. Create a folder with the desired skin name.

Note - Verify that this name is not already used in `chkp`. If it is, the new skin definition will override the existing skin definition (as long as the new skin definition exists). Once you have deleted the new skin definition, the `chkp` skin definition will once again be used.

Each skin folder must contain the following five style sheets:

- `help_data.css`: The main OLH page uses this style sheet.
- `help.css`: The inner frame on the OLH page uses this style sheet.
- `index.css`: The ESOD pages, and the main SSL Network Extender portal page use this style sheet.
- `style.css`: All login pages use this style sheet.
- `style_main.css`: The main SSL Network Extender Connection page, Proxy Authentication page and Certificate Registration page use this style sheet.

Note - It is recommended that you copy the aforementioned files from another `chkp` skin, and then modify them as desired.

3. Install Policy after creating the new skin.

Example

Add your company logo to the main SSL Network Extender portal page.

```
cd $FWDIR/conf/extender/skin/custom

mkdir <skin_name>

cd <skin_name>

copy ../../chkp/skin2/* .
```

Place logo image file in this directory

Edit `index.css`.

Goto `.company_logo` and replace the existing URL reference with a reference to the new logo image file.

Save.

Install Policy.

Note - No spaces are allowed in the `<skin_name>`

Configuring the Languages Option

To configure the Languages Option:

The `languages` directory is located under `$FWDIR/conf/extender` on the SSL Network Extender gateways.

There may be two subdirectories. They are:

- `chkp`: contains languages that Check Point provides by default. At upgrade, this subdirectory may be overwritten.
- `custom`: contains languages defined by the customer. If `custom` does not exist yet, create it. At upgrade, this subdirectory is not overwritten. New languages are added in this subdirectory.

Disabling a Language

1. Enter the specific language subdirectory, under `custom`, that is to be disabled (if it exists) and create a file named `disable`. This file may be empty.
2. If the specific language does not exist under `custom`, create it and then create a file within it named `disable`.
3. Install Policy. The next time that the user connects to the SSL Network Extender portal, this language will not be available to him/her.

Adding a Language

1. Enter the `custom` subdirectory.
2. Create a folder with the desired language name.

   Note - Verify that this name is not already used in `chkp`. If it is, the new language definition will override the existing language definition (as long as the new language definition exists). Once you have deleted the new language definition, the `chkp` language definition will once again be used.

3. Copy the `messages.js` file of an existing `chkp` language to this folder.
4. Edit the `messages.js` file and translate the text bracketed by quotation marks.
5. Save.
6. Install Policy after adding the new language.

Example

```
cd $FWDIR/conf/extender/language

mkdir custom

cd custom

mkdir <language_name>

cd <language_name>

copy ../../chkp/english/messages.js
```

Edit the `messages.js` file and translate the text bracketed by quotation marks.

Save.

In `custom/english/messages.js`, add a line as follows:

```
<language_name>=”translation of language_name”;
```

Install Policy.

Note - No spaces are allowed in the <language_name>

Modifying a Language

1. Enter the custom subdirectory.
2. Create a folder with a language name that matches the chkp language folder to be modified.
3. Create an empty messages.js file, and insert only those messages that you want to modify, in the following format:

   ```
   <variable_name>=”<desired text>”;
   ```

Note - For reference, refer to the messages.js file, located in chkp/<language>.

Installation for Users without Administrator Privileges

The SSL Network Extender usually requires Administrator privileges to install the ActiveX component. To allow users that do not have Administrator privileges to use the SSL Network Extender, the Administrator can use his/her remote corporate installation tools (such as, Microsoft SMS) to publish the installation of the SSL Network Extender, as an MSI package, in configuring the SSL Network Extender.

To prepare the SSL Network Extender MSI package:

1. Move the extender.cab file, located in $FWDIR/conf/extender, to a Windows machine and open the file using WinZip.
2. Extract the cpextender.msi, and use as an MSI package, for remote installation.

On Windows Vista, Mac and Linux, it is possible to install SSL Network Extender for users that are not administrators, if the user knows the admin password. In this case, perform a regular SSL Network Extender installation and supply the administrator password when asked.

SSL Network Extender User Experience

In This Section:

This section describes the user experience, including downloading and connecting the SSL Network Extender client, importing a client certificate, and uninstall on disconnect.

Configuring Microsoft Internet Explorer

Check Point SSL Network Extender uses ActiveX controls and cookies to connect to applications via the Internet. These enabling technologies require specific browser configuration to ensure that the applications are installed and work properly on your computer. The Trusted Sites Configuration approach includes the SSL Network Extender Portal as one of your Trusted Sites. This approach is highly recommended, as it does not lessen your security. Please follow the directions below to configure your browser.

Trusted Sites Configuration

1. In Internet Explorer, select **Tools > Internet Options > Security**.
2. Select **Trusted sites**.
3. Click **Sites**.
4. Enter the URL of the SSL Network Extender Portal and click **Add**.
5. Click **OK** twice.

About ActiveX Controls

ActiveX controls are software modules, based on Microsoft's Component Object Model (COM) architecture. They add functionality to software applications by seamlessly incorporating pre-made modules with the basic software package.

On the Internet, ActiveX controls can be linked to Web pages and downloaded by an ActiveX-compliant browser. ActiveX controls turn Web pages into software pages that perform like any other program.

The SSL Network Extender can use ActiveX control in its applications. To use ActiveX you must download the specific ActiveX components required for each application. Once these components are loaded, you do not need to download them again unless upgrades or updates become available. If you do not want to use an ActiveX component you may work with a Java Applet.

Note - You must have Administrator rights to install or uninstall software on Windows XP Professional, as well as on the Windows 2000 operating systems.

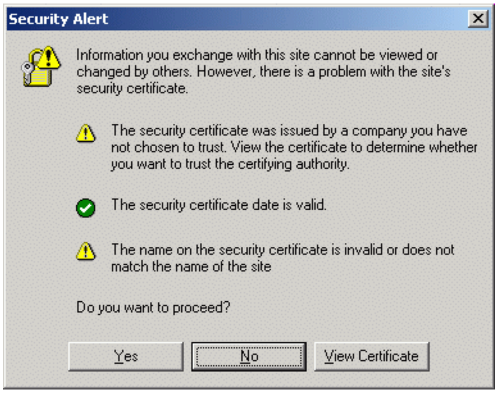
Downloading and Connecting the Client

The following section discusses how to download and connect the SSL Network Extender.

To download the Client

1. Using Internet Explorer, browse to the SSL Network Extender portal of the gateway at https://<GW name or IP>. The following Security Alert window may be displayed.

Figure 30-4 Security Alert Window



Security Alert

Information you exchange with this site cannot be viewed or changed by others. However, there is a problem with the site's security certificate.

⚠ The security certificate was issued by a company you have not chosen to trust. View the certificate to determine whether you want to trust the certifying authority.

✓ The security certificate date is valid.

⚠ The name on the security certificate is invalid or does not match the name of the site

Do you want to proceed?

The site’s security certificate has been issued by an authority that you have not designated as a trusted CA. Before you connect to this server, you must trust the CA that signed the server certificate. (The system administrator can define which CAs may be trusted by the user.) You can view in the certificate in order to decide if you wish to proceed.

Note - The administrator can direct the user to the URL, `http://< mngmt IP>:18264`, to install this CA certificate, thereby establishing trust, and avoiding future displays of this message.

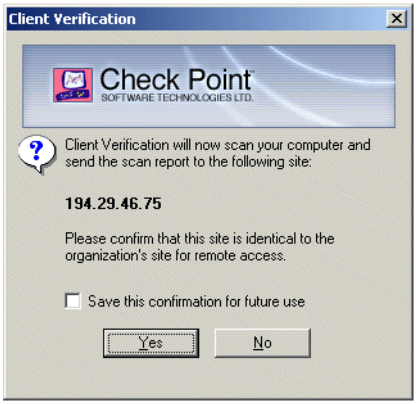
2. Click **Yes**.

   If Endpoint Security on Demand is enabled, the **ESOD web page** is displayed.

   If this is the first time that the user is scanned with ESOD, the user should install the ESOD ActiveX object.

   If this is the first time that ESOD is used, the following **Server Confirmation** window appears. The user is asked to confirm that the listed ESOD server is identical to the organization’s site for remote access.

Figure 30-5 Server Confirmation window



Client Verification will now scan your computer and send the scan report to the following site:

194.29.46.75

Please confirm that this site is identical to the organization's site for remote access.

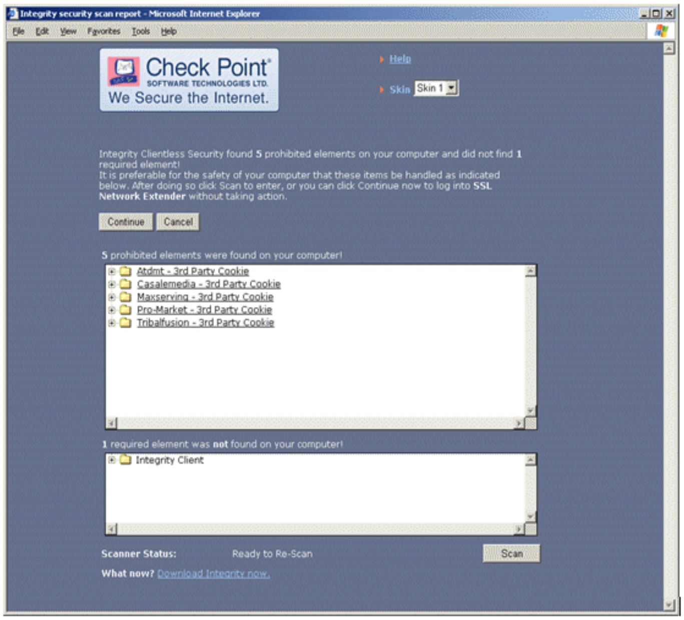
□ Save this confirmation for future use

3. Click one of the following:

   - If the user clicks **No**, an error message is displayed and the user is denied access.
   - If the user clicks **Yes** in Figure 30-5, the ESOD client continues the software scan. Moreover, if the **Save this confirmation for future use** check box is selected, the **Server Confirmation** window will not appear the next time the user attempts to login.

     Once the user has confirmed the ESOD server, an automatic software scan takes place on the client's machine. Upon completion, the scan results and directions on how to proceed are displayed as shown in Figure 30-6.

Figure 30-6 Scan Results



Integrity security scan report - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Check Point SOFTWARE TECHNOLOGIES LTD. We Secure the Internet.
Help
Skin
Continue
Cancel
Scan

Skin: Skin 1

Integrity Clientless Security found **5** prohibited elements on your computer and did not find **1** required element!
It is preferable for the safety of your computer that these items be handled as indicated below. After doing so click Scan to enter, or you can click Continue now to log into **SSL Network Extender** without taking action.

5 prohibited elements were found on your computer!

- Atdmt - 3rd Party Cookie
- Casalemedia - 3rd Party Cookie
- Maxserving - 3rd Party Cookie
- Pro-Market - 3rd Party Cookie
- Tribalfusion - 3rd Party Cookie

1 required element was **not** found on your computer!

- Integrity Client

Scanner Status: Ready to Re-Scan

What now? Download Integrity now.

ESOD not only prevents users with potentially harmful software from accessing your network, but also requires that they conform to the corporate antivirus and firewall policies, as well. A user is defined as having successfully passed the ESOD scan only if he/she successfully undergoes scans for *Malware, Anti-Virus,* and *Firewall*. Each malware is displayed as a link, which, if selected, redirects you to a data sheet describing the detected malware. The data sheet includes the name and a short description of the detected malware, what it does, and the recommended removal method/s.

The options available to the user are configured by the administrator on the ESOD server. The options are listed in the following table:

Table 30-2 Scan Options

Scan Option	Description
Scan Again	Allows a user to rescan for malware. This option is used in order to get refreshed scan results, after manually removing an undesired software item.
Cancel	Prevents the user from proceeding with the portal login, and closes the current browser window.
Continue	Causes the ESOD for Connectra client to disregard the scan results and proceed with the log on process.

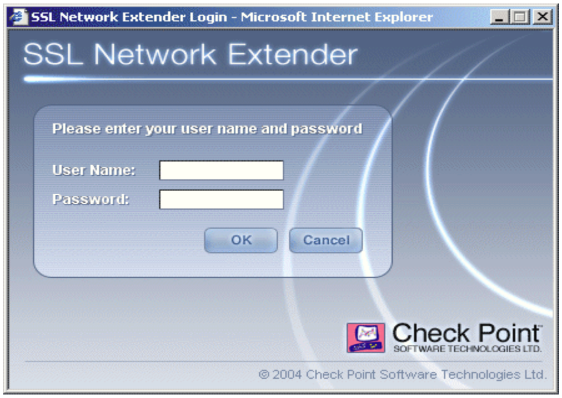
4. You can select a different language from the **Language** drop-down list (see Figure 30-6). If you change languages, while connected to the SSL Network Extender portal, you will be informed that if you continue the process you will be disconnected, and must reconnect.

5. You can select a different skin from the **Skin** drop-down list (see Figure 30-6). You can change skins, while connected to the SSL Network Extender portal.

6. Click **Continue** (see Figure 30-6).

   - If the configured authentication scheme is **User Password Only**, the following **SSL Network Extender Login** window is displayed.

Figure 30-7 SSL Network Extender Login Window



SSL Network Extender Login - Microsoft Internet Explorer
SSL Network Extender
Please enter your user name and password
User Name:
Password:
OK
Cancel
Check Point
SOFTWARE TECHNOLOGIES LTD.
© 2004 Check Point Software Technologies Ltd.

   Enter the **User Name** and **Password** and click **OK**.

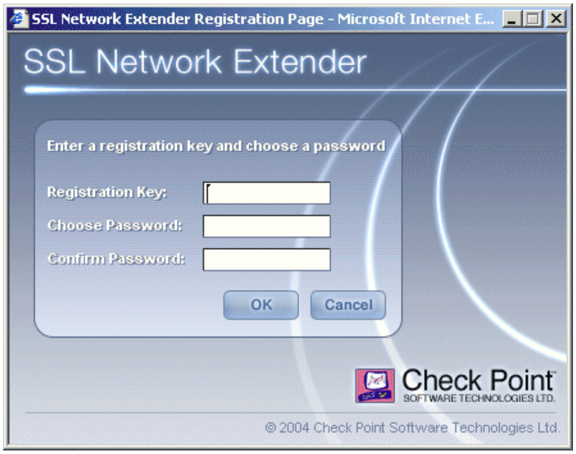
Note - If user authentication has been configured to be performed via a 3rd party authentication mechanism, such as SecurID or LDAP, the Administrator may require the user to change his/her PIN, or Password. In such a case, an additional Change Credentials window is displayed, before the user is allowed to access the SSL Network Extender.

   - If the configured authentication scheme is **Certificate without Enrollment**, and the user already has a certificate. If the user does not already have a certificate, access is denied.
   - If the configured authentication scheme is **Certificate with Enrollment**, and the user does not already have a certificate, the **Enrollment** window is displayed (see Figure 30-8):

     Enter the **Registration Key** and select PKCS#12 Password.

     Click **Ok**. The PKCS#12 file is downloaded.

Figure 30-8 Enrollment window


SSL Network Extender Registration Page - Microsoft Internet E...
SSL Network Extender
OK
Cancel
Check Point
SOFTWARE TECHNOLOGIES LTD.

Enter a registration key and choose a password

Registration Key:

Choose Password:

Confirm Password:

© 2004 Check Point Software Technologies Ltd.

At this point the user should open the file and utilize the Microsoft Certificate Import wizard as follows.

Note - It is strongly recommended that the user set the property **Do not save encrypted pages to disk** on the **Advanced** tab of the **Internet Properties** of Internet Explorer. This will prevent the certificate from being cached on disk.

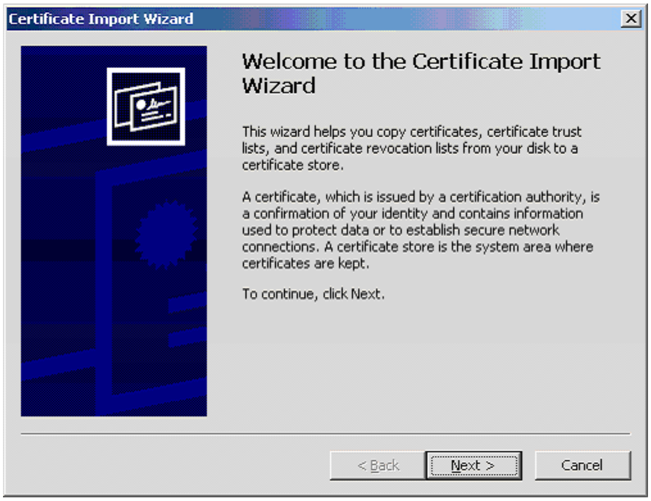
Importing a Client Certificate with the Microsoft Certificate Import Wizard to Internet Explorer

Importing a client certificate to Internet Explorer is acceptable for allowing access to either a home PC with broadband access, or a corporate laptop with a dial-up connection. The client certificate will be automatically used by the browser, when connecting to an SSL Network Extender gateway.

To import a client certificate:

a. Open the downloaded PKCS#12 file. The following **Certificate Import Wizard** window appears:

Figure 30-9 Certificate Import Wizard window



Certificate Import Wizard

Welcome to the Certificate Import Wizard

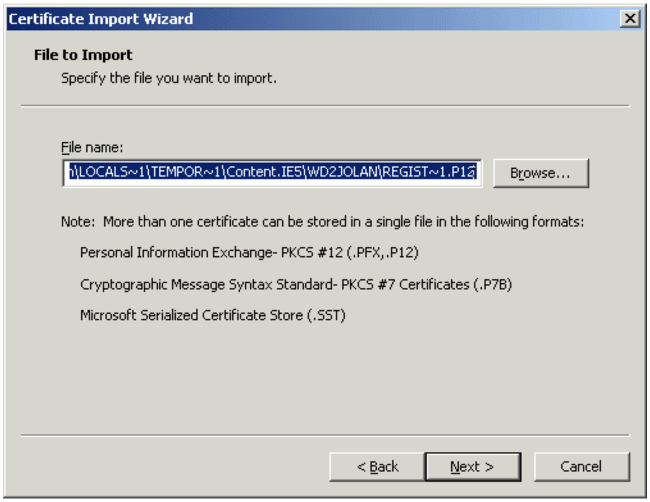
This wizard helps you copy certificates, certificate trust lists, and certificate revocation lists from your disk to a certificate store.

A certificate, which is issued by a certification authority, is a confirmation of your identity and contains information used to protect data or to establish secure network connections. A certificate store is the system area where certificates are kept.

To continue, click Next.

b. Click **Next**. The following **File to Import** window appears:

Figure 30-10 File to Import window



Certificate Import Wizard

File to Import

Specify the file you want to import.

File name:

n\LOCALS~1\TEMPOR~1\Content.IE5\WD2JOLAN\REGIST~1.P12

Note: More than one certificate can be stored in a single file in the following formats:

Personal Information Exchange- PKCS #12 (.PFX,.P12)

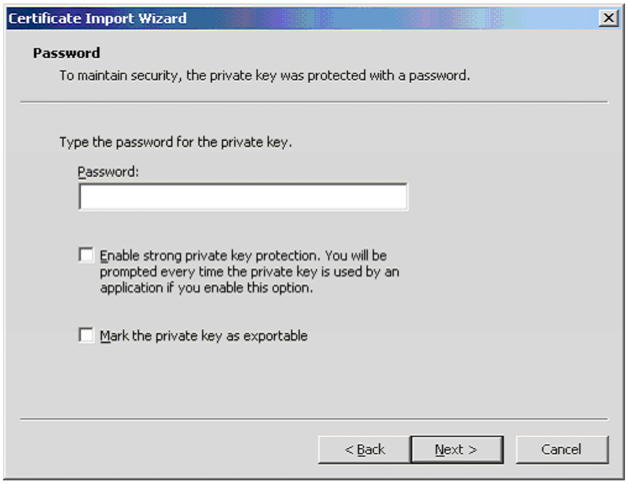
Cryptographic Message Syntax Standard- PKCS #7 Certificates (.P7B)

Microsoft Serialized Certificate Store (.SST)

The P12 file name is displayed.

c. Click **Next**. The following **Password** window appears:

Figure 30-11 Password window



Certificate Import Wizard

Password

To maintain security, the private key was protected with a password.

Type the password for the private key.

Password:

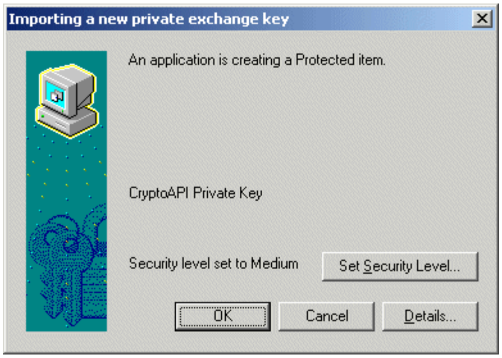
□ Enable strong private key protection. You will be prompted every time the private key is used by an application if you enable this option.

□ Mark the private key as exportable

It is strongly recommended that the user enable **Strong Private Key Protection**. The user will then be prompted for consent/credentials, as configured, each time authentication is required. Otherwise, authentication will be fully transparent for the user.

d. Enter your password, click **Next** twice. If the user enabled **Strong Private Key Protection,** the following **Importing a New Private Exchange Key** window appears:

Figure 30-12 Importing a New Private Exchange Key window



Importing a new private exchange key

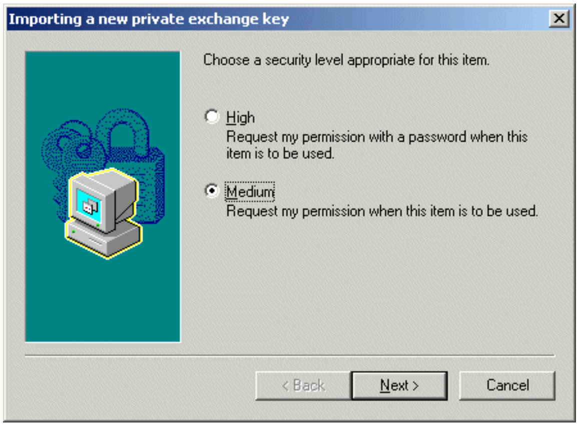
An application is creating a Protected item.

CryptoAPI Private Key

Security level set to Medium

- If you click **OK**, the Security Level is assigned the default value **Medium**, and the user will be asked to consent each time the certificate is required for authentication.
- If you click **Set Security Level**, the following **Set Security Level** window appears. Select either **High** or **Medium** and click **Next**.

Figure 30-13 Set Security Level window



Importing a new private exchange key

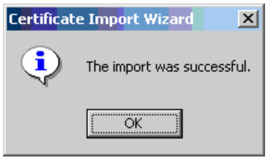
Choose a security level appropriate for this item.

○ High
Request my permission with a password when this item is to be used.

● Medium
Request my permission when this item is to be used.

e. Click **Finish**. The following **Import Successful** window appears:

Figure 30-14 Import Successful window



Certificate Import Wizard

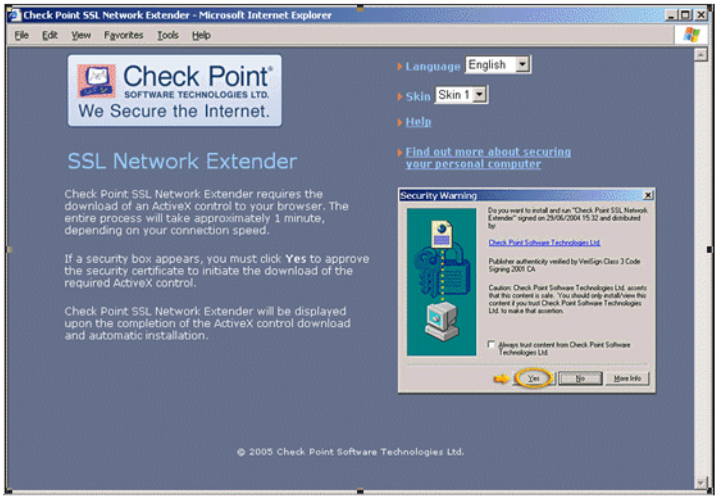
The import was successful.

f. Click **OK**.

g. Close and reopen your browser. You can now use the certificate that has now been imported for logging in.

7. If you are connecting to the SSL gateway for the first time, a VeriSign certificate message appears, requesting the user’s consent to continue installation.

Figure 30-15 VeriSign Certificate Message



Check Point SSL Network Extender - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Check Point SOFTWARE TECHNOLOGIES LTD.
We Secure the Internet.
Language English
Skin Skin 1
Help
Find out more about securing your personal computer
Security Warning
Yes
No
More Info

SSL Network Extender

Check Point SSL Network Extender requires the download of an ActiveX control to your browser. The entire process will take approximately 1 minute, depending on your connection speed.

If a security box appears, you must click **Yes** to approve the security certificate to initiate the download of the required ActiveX control.

Check Point SSL Network Extender will be displayed upon the completion of the ActiveX control download and automatic installation.

Do you want to install and run "Check Point SSL Network Extender" signed on 29/06/2004 15:32 and distributed by:

Check Point Software Technologies Ltd

Publisher authenticity verified by VeriSign Class 3 Code Signing 2001 CA

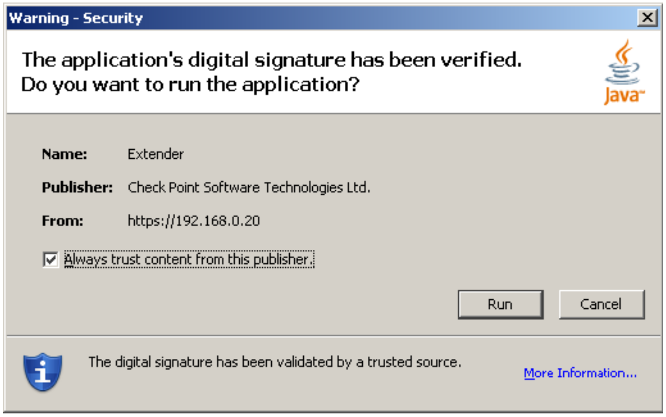
Caution: Check Point Software Technologies Ltd. asserts that this content is safe. You should only install/view this content if you trust Check Point Software Technologies Ltd. to make that assertion.

□ Always trust content from Check Point Software Technologies Ltd

© 2005 Check Point Software Technologies Ltd.

- If you connect using Java Applet, the following Java security message will appear. Click **Yes**.

Figure 30-16 Java Security Message



The application's digital signature has been verified.
Do you want to run the application?

Name: Extender

Publisher: Check Point Software Technologies Ltd.

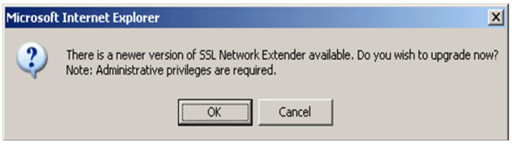
From: https://192.168.0.20

☑ Always trust content from this publisher.

The digital signature has been validated by a trusted source.

- If the system administrator configured the upgrade option, the following Upgrade Confirmation window is displayed:

Figure 30-17 Upgrade Confirmation window



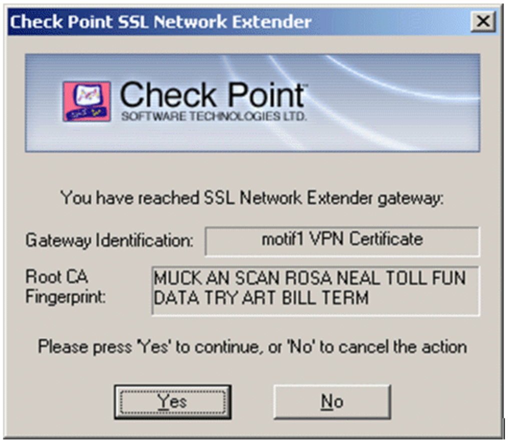
There is a newer version of SSL Network Extender available. Do you wish to upgrade now?
Note: Administrative privileges are required.

If you click **OK**, you must reauthenticate and a new SSL Network Extender version is installed.

If you click **Cancel**, the SSL Network Extender connects normally. (The **Upgrade Confirmation** window will not be displayed again for a week.) The **SSL Network Extender** window appears. A **Click here to upgrade** link is displayed in this window, enabling the user to upgrade even at this point. If you click on the **Click here to upgrade** link, you must reauthenticate before the upgrade can proceed.

8. At first connection, the user is notified that the client will be associated with a specific gateway. Click **Yes**.

Figure 30-18 Client associated with specific gateway



Check Point SSL Network Extender

You have reached SSL Network Extender gateway:

Gateway Identification: motif1 VPN Certificate

Root CA Fingerprint: MUCK AN SCAN ROSA NEAL TOLL FUN DATA TRY ART BILL TERM

Please press 'Yes' to continue, or 'No' to cancel the action

The server certificate of the gateway is authenticated. If the system Administrator has sent the user a *fingerprint*, it is strongly recommended that the user verify that the root CA fingerprint is identical to the fingerprint, sent to him/her.

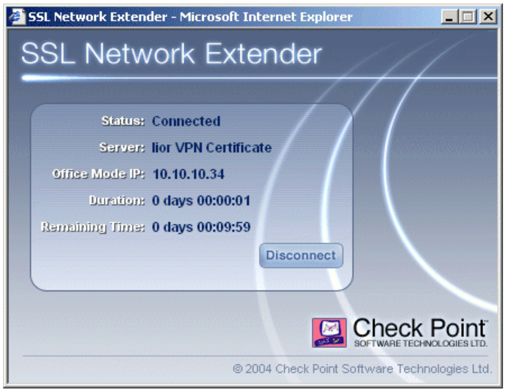
The system Administrator can view and send the fingerprint of all the trusted root CAs, via the **Certificate Authority Properties** window in SmartDashboard.

9. If the user is using a proxy server that requires authentication, the **Proxy Authentication** pop-up is displayed. The user must enter his/her proxy username and password, and click **OK**.

10. If you are connected with Windows Vista, a **Windows Firewall** message will appear. Click **Unblock**.

You may work with the client as long as the **SSL Network Extender Connection** window, shown below, remains open, or minimized (to the System tray).

Figure 30-19 Client connected



Status: Connected

Server: lior VPN Certificate

Office Mode IP: 10.10.10.34

Duration: 0 days 00:00:01

Remaining Time: 0 days 00:09:59

SSL Network Extender - Microsoft Internet Explorer
SSL Network Extender
Disconnect
Check Point
SOFTWARE TECHNOLOGIES LTD.
© 2004 Check Point Software Technologies Ltd.

Once the SSL Network Extender is initially installed, a new Windows service named Check Point SSL Network Extender and a new virtual network adapter are added. This new network adapter can be seen by typing `ipconfig /all` from the Command line.

Note - The settings of the adapter and the service must not be changed. IP assignment, renewal and release will be done automatically.

Both the virtual network adapter and the Check Point SSL Network Extender service are removed during the product uninstall.

Note - The Check Point SSL Network Extender service is dependent on both the virtual network adapter and the DHCP client service. Therefore, the DHCP client service must not be disabled on the user's computer.

There is no need to reboot the client machine after the installation, upgrade, or uninstall of the product.

11. When you finish working, click **Disconnect** to terminate the session, or when the window is minimized, right-click the icon and click **Disconnect**. The window closes.

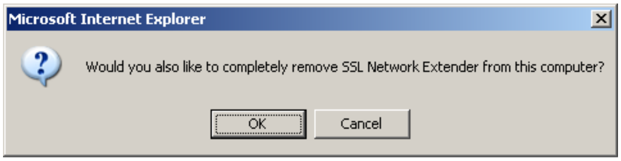
Uninstall on Disconnect

If the administrator has configured **Uninstall on Disconnect** to ask the user whether or not to uninstall, the user can configure **Uninstall on Disconnect** as follows.

To set Uninstall on Disconnect:

1. Click **Disconnect.** The **Uninstall on Disconnect** window is displayed, as shown in the following figure.

Figure 30-20 Uninstall on Disconnect

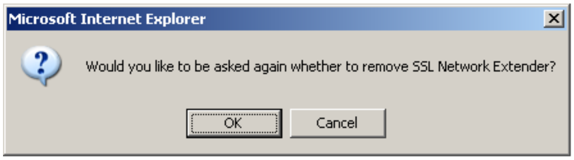


Microsoft Internet Explorer

Would you also like to completely remove SSL Network Extender from this computer?

2. Click **Ok** to Uninstall.

   If you select **Cancel** in Figure 30-20 the SSL Network Extender will not be uninstalled and the following message appears:



Microsoft Internet Explorer

Would you like to be asked again whether to remove SSL Network Extender?

   If you click **OK**, the **Uninstall on Disconnect** window will be displayed the next time the user connects to the SSL Network Extender.

Using SSL Network Extender on Linux / Mac Operating Systems

There are two methods to access Network Applications using Linux.:

- Java
- Command Line

Java

1. When connecting for the first time, the SSL Network Extender installation archive package is downloaded.

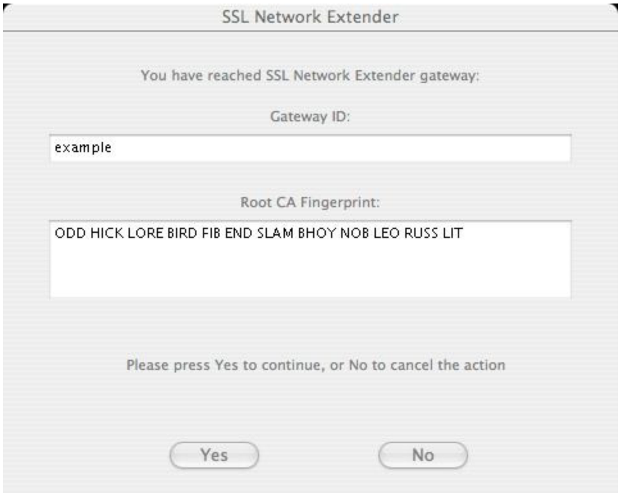
   This process is similar to the Windows Java installation.

2. If the user does not have root permissions, the user is prompted to enter a root password in order to install the package. Enter the password and press **Enter**.

   After the installation is finished, the applet will try to connect.

   If it is the first time, the following window is displayed:

Figure 30-21 Fingerprint Verification



SSL Network Extender

You have reached SSL Network Extender gateway:

Gateway ID:

example

Root CA Fingerprint:

ODD HICK LORE BIRD FIB END SLAM BHOY NOB LEO RUSS LIT

Please press Yes to continue, or No to cancel the action

   If the system Administrator has sent the user a fingerprint, it is strongly recommended that the user verify that the server certificate fingerprint is identical to the **Root CA Fingerprint** seen in the window.

3. Click **Yes** to confirm.

Command Line

To download the SSL Network Extender installation archive package:

1. In the **Network Applications Settings** window, click on **click here** in the sentence **For Linux command line SSL Network Extender installation click here.** The Shell archive package is downloaded to the users home directory.

   Before running the installation script, make sure execute permissions are available on the file. Use the command `chmod + x snx_install.sh` to add execution permissions.

2. Download the SSL Network Extender manual installation.

   The following links will appear:

   - Download MSI installation package for Windows

- Download command line SSL Network Extender for Linux
- Download command line SSL Network Extender for Macintosh

3. Select the appropriate operating system.

   The Shell archive package is downloaded to the user's home directory.

4. To execute the installation script run `snx_install.sh`.

   If the user does not have root permissions, the user is prompted to enter a root password in order to install the package. Enter the password and press **Enter**.

5. To connect after installation perform the following:

```
---------- Connect using the SNX command

Server_1:/ snx -s <server name> -u <user name>

Check Point's Linux SNX

build 5416000XX

---------- Enter Password

Please enter your password:

SNX authentication:

Please confirm the connection to gateway: <server name>

Root CA fingerprint: MOOD TREK ALP EEL FILM MESH RUBY BELA MACE

TEND DRY PUT

---------- Accept Fingerprint if it is valid

Do you accept? [y]es/[N]o: y

SNX - connected

Session parameters:

===================

Office Mode IP : 9.1.3.9

DNS Server : 19.18.17.16

DNS Suffix : domain.com
```

   Timeout : 10 minutes

6. To connect after installation perform the following:

```
---------- Disconnect by running the following

Server_1:/ snx -d

SNX - Disconnecting... done.
```

SSL Network Extender Command Attributes

Table 30-3 SSL Network Extender Command Attributes

Attributes	Description
`snx -f <configuration file>`	Run SSL Network Extender using parameters defined in a configuration file other than the default name or location.
`snx -d`	Disconnect from Connectra
`snx -s <server>`	Specify server IP or hostname
`snx -u <username>`	Specify a valid user
`snx -c <certificate file>`	Specify which certificate is used to authenticate.
`snx -l <CA directory>`	Define the directory where CA's certificates are stored.
`snx -p <port>`	Change the HTTPS port. (default port is TCP 443).
`snx -g`	Enable debugging. snx.elg log file is created.
`snx -e <cipher>`	Force a specific encryption algorithm. Valid values - RC4 and 3DES.

Configuration File Attributes

It is possible to predefine SSL Network Extender attributes by using a configuration file (`.snxrc`) located in the users home directory. When the SSL Network Extender command SSL Network Extender is executed, the attributed stored in the file are used by the SSL Network Extender command. To run a file with a different name execute the command `snx -f <filenmae>`.

Table 30-4 Configuration File Attributes

Attributes	Description
`server`	Specify server IP or hostname
`sslport`	Change the HTTPS port. (default port is TCP 443).
`username`	Specify a valid user
`certificate`	Specify which certificate is used to authenticate
`calist`	Define the directory where CA's certificates are stored.
`reauth`	Enable reauthentication. Valid values - {yes, no}
`debug`	Enable debugging. snx.elg log file is created. Valid values - {yes, no}. To activate debugging when running java, create a .snxrc file with the line debug yes in the home directory.
`cipher`	Force a specific encryption algorithm. Valid values: RC4 and 3DES
`proxy_name`	Define a Proxy hostname
`proxy_port`	Define a proxy port
`proxy_user`	Define a proxy user
`proxy_pass`	Define a password for proxy authentication

Note - Proxy information can only be configured in the configuration file and not directly from the command line.

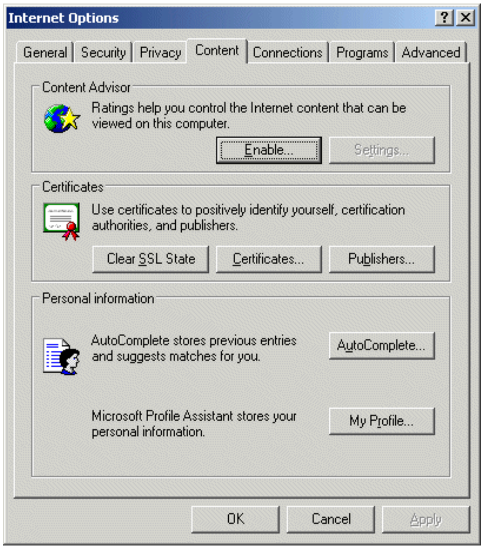
Removing an Imported Certificate

If you imported a certificate to the browser, it will remain in storage until you manually remove it. It is strongly recommended that you remove the certificate from a browser that is not yours.

To remove the imported certificate:

1. In the **Internet Options** window, shown in the following figure, access the **Content** tab.

Figure 30-22 Internet Options window



Ratings help you control the Internet content that can be viewed on this computer.

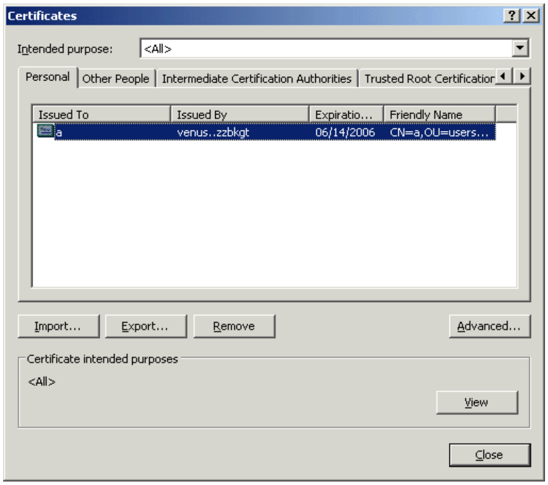
Use certificates to positively identify yourself, certification authorities, and publishers.

AutoComplete stores previous entries and suggests matches for you.

Microsoft Profile Assistant stores your personal information.

2. Click **Certificates.** The **Certificates** window is displayed:

Figure 30-23 Certificates window



Intended purpose: <All>

Issued To	Issued By	Expiratio...	Friendly Name
a	venus..zzbkgt	06/14/2006	CN=a,OU=users...

Certificate intended purposes

<All>

3. Select the certificate to be removed, and click **Remove**.

Troubleshooting

Tips on how to resolve issues that you may encounter.

SSL Network Extender Issues

All user's packets destined directly to the external SSL Network Extender gateway will not be encrypted by the SSL Network Extender.

If there is a need to explicitly connect to the gateway through the SSL tunnel, connect to the internal interface, which is part of the encryption domain.

The SSL Network Extender gateway allows users to authenticate themselves via certificates. Therefore, when connecting to the SSL Network Extender gateway, the following message may appear: "The Web site you want to view requests identification. Select the certificate to use when connecting."

In order not to display this message to the users, two solutions are proposed:

1. On the client computer, access the Internet Explorer. Under **Tools > Options > Security** tab, select **Local intranet > Sites**. You can now add the SSL Network Extender gateway to the Local intranet zone, where the Client Authentication pop-up will not appear. Click **Advanced**, and add the gateway's external IP or DNS name to the existing list.
2. On the client computer, access the Internet Explorer. Under **Tools > Options > Security** tab, select **Internet Zone > Custom Level**. In the **Miscellaneous** section, select **Enable** for the item **Don't prompt for client certificate selection when no certificates or only one certificate exists**. Click **OK**. Click **Yes** on the Confirmation window. Click **OK** again.

Note - This solution will change the behavior of the Internet Explorer for all Internet sites, so if better granularity is required, refer to the previous solution.

If the client computer has SecuRemote/SecureClient software installed, and is configured to work in 'transparent mode', and its encryption domain contains SSL Network Extender gateway, or otherwise overlaps with the SSL Network Extender encryption domain, the SSL Network Extender will not function properly.

To resolve this, disable the overlapping site in SecuRemote/SecureClient.

If the client computer has SecuRemote/SecureClient software installed, and is configured to work in ‘connect mode’, and its encryption domain contains SSL Network Extender gateway, or otherwise overlaps with the SSL Network Extender encryption domain, the SSL Network Extender will not function properly.

To resolve this, verify that the flag `allow_clear_traffic_while_disconnected` is **True** (which is the default value).

SSL Network Extender connections can not pass SCV rules. SecureClient users must be differentiated from SSL Network Extender users in order to allow the SecureClient connections to pass the SCV rules.

One way to do this is to use the SCV capabilities in the rulebase. In **Traditional Mode** you can configure two types of rules, by selecting the Apply Rule Only if Desktop Configuration Options are verified. The selected (SCV) rules will pass only SecureClient connections, while the rules that were not selected will pass SecureClient and SSL Network Extender connections. When using **Simplified Mode**, the Administrator may specify services that will be excluded from SCV checking. Both SecureClient and SSL Network Extender clients attempting to access such services will be allowed access, even when not SCV verified. SCV will not be enforced on specified services for both types of clients.

ESOD Issues

User did not pass the scan (a 'Continue' button is not displayed).

The user probably did not match the policy requirements.

- If using "ESOD per User Group" feature – Verify that the user is using the correct policy.
- According to the policy, Explain the user how to remove the elements that are blocking him.

User cannot access the given URL for his specific group.

- Make sure that the group listed in the URL is listed in the ics.group file, with the correct xml file.
- Make sure that the xml file that is assigned to the group exists in `$FWDIR/conf/extender`.
- Make sure Install Policy has been made since the ics.group file has changes.

User has passed the ESOD scan, but gets a "Wrong ESOD Scan" error when trying to connect.

This means that the user has passed the scan intended for a group that he does not belong to.

- Verify that the user is using the correct URL.
- Look at the SmartView Tracker. The log should state which xml file the user used for the scan.
- Make sure that this file is the same as the user's group file. If not, direct the user to the correct URL.

Chapter 31

Resolving Connectivity Issues

In This Chapter

The Need for Connectivity Resolution Features

While there are a few connectivity issues regarding VPN between gateways, remote access clients present a special challenge. Remote clients are, by their nature, mobile. During the morning they may be located within the network of a partner company, the following evening connected to a hotel LAN or behind some type of enforcement or NATing device. Under these conditions, a number of connectivity issues can arise:

- Issues involving NAT devices that do not support fragmentation.
- Issues involving service/port filtering on the enforcement device

Check Point Solution for Connectivity Issues

Check Point resolves NAT related connectivity issues with a number of features:

- IKE over TCP
- Small IKE phase II proposals
- UDP encapsulation
- IPSec Path Maximum Transmission Unit (IPSec PMTU)

Check Point resolves port filtering issues with *Visitor Mode* (formally: *TCP Tunneling*).

Other Connectivity Issues

Other connectivity issues can arise, for example when a remote client receives an IP address that matches an IP on the internal network. Routing issues of this sort are resolved using Office mode. For more information see: "Office Mode".

Other issues, such as Domain Name Resolution involving DNS servers found on an internal network protected by a gateway, are resolved with *Split DNS*. For more information on Split DNS see: "Remote Access Advanced Configuration".

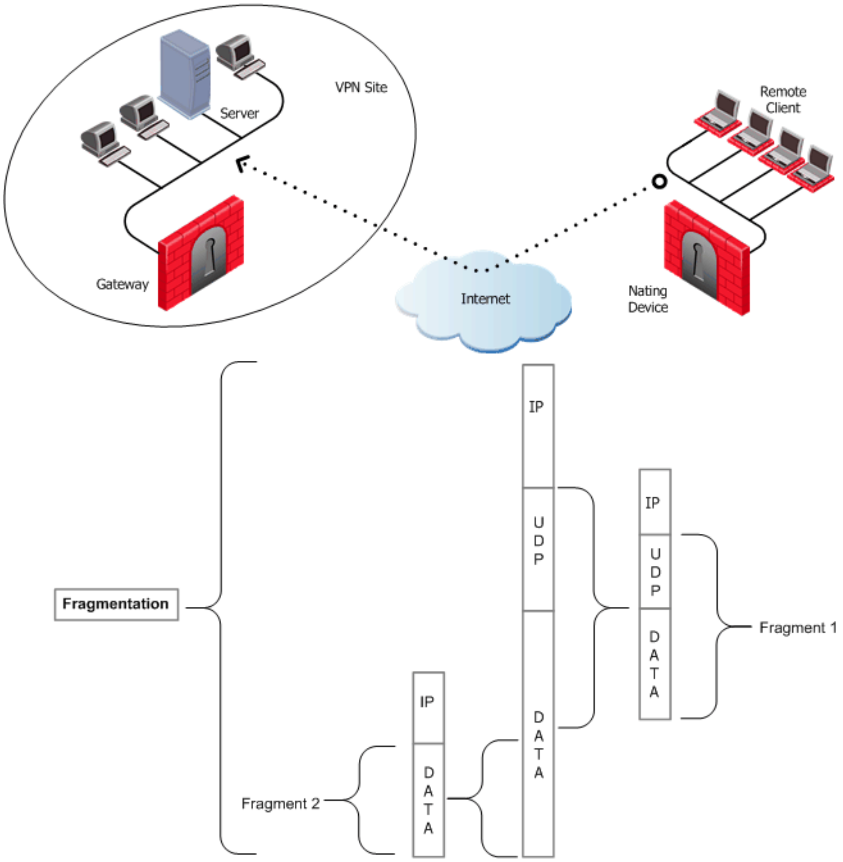
Overcoming NAT Related Issues

NAT related issues arise with *hide* NAT devices that do not support packet fragmentation.

When a remote access client attempts to create a VPN tunnel with its peer gateway, the IKE or IPSec packets may be larger than the Maximum Transmission Unit (MTU) value. If the resulting packets *are* greater than the MTU, the packets are fragmented at the Data Link layer of the Operating System's TCP/IP stack.

Problems arise when the remote access client is behind a hide NAT device that does not support this kind of packet fragmentation:

Figure 31-1 UDP Fragmentation



Hide NAT not only changes the IP header but also the port information contained in the UDP header. In Figure 31-1, the UDP packet is too long so the remote client fragments the packet. The first fragment consists of the IP header plus the UDP header and some portion of the data. The second fragment consists of only the IP header and the second data fragment. The NATing device does not know how to wait for all the fragments, reassemble and NAT them.

When the first fragment arrives, the NAT device successfully translates the address information in the IP header, and port information in the UDP header and forwards the packet. When the second fragment arrives, the NATing device cannot translate the port information because the second packet does not contain a UDP header; the packet is dropped. The IKE negotiation fails.

During IKE phase I

To understand why large UDP packets arise, we need to take a closer look at the first phase of IKE. During IKE phase I, the remote access client and gateway attempt to authenticate each other. One way of authenticating is through the use of certificates. If the certificate or Certificate Revocation List (CRL) is long, large UDP packets result, which are then fragmented by the operating system of the remote client.

Note - If the VPN peers authenticate each other using pre-shared secrets, large UDP packets are not created; however, certificates are more secure, and thus recommended.

IKE Over TCP

IKE over TCP solves the problem of large UDP packets created during IKE phase I. The IKE negotiation is performed using TCP packets. TCP packets are not fragmented; in the IP header of a TCP packet, the DF flag (“do not fragment”) is turned on. A full TCP session is opened between the peers for the IKE negotiation during phase I.

During IKE phase II

A remote access client does not have a policy regarding methods of encryption and integrity. Remote access clients negotiate methods for encryption and integrity via a series of proposals, and need to negotiate *all* possible combinations with the gateway. This can lead to large UDP packets which are once again fragmented by the remote client’s OS before sending. The NAT device in front of the remote client drops the packet that has no UDP header (containing port information). Again, the IKE negotiation fails.

Why not use IKE over TCP again, as in phase I?

IKE over TCP solves the fragmentation problem of long packets, but in phase II there are times when the gateway needs to *initiate* the connection to the remote client. (Only the remote client initiates phase I, but either side can identify the need for a phase II renewal of keys; if the gateway identifies the need, the gateway initiates the connection.)

If the gateway initiates the connection, the gateway knows the IP address of the NATing device, but cannot supply a port number that translates to the remote client *behind* the NATing device. (The port number used during previous connections is only temporary, and can quickly change.) The NATing device cannot forward the connection correctly for the remote client; the connection initiated by the gateway fails.

It is possible to use IKE over TCP, but this demands a TCP connection to be always open; the open session reserves the socket on the gateway, taking up valuable system resources. The more reasonable solution is to keep open the port on the NATing device by sending UDP "keep alive" packets to the gateway, and then performing IKE phase II in the usual way. However, there is still a need to shorten the UDP packets to prevent possible fragmentation.

Small IKE Phase II Proposals

Both gateway and remote peer start the IKE negotiation by proposing a small number of methods for encryption and integrity. The more common methods are included in the small proposals.

If proposals match between the remote client and the gateway, the proposed methods are used; if no match is found, a greater number of proposals are made. Usually a match is found with the small proposals, and fragmentation is no longer an issue. However, there are cases where a match is not found, and a larger number of proposals need to be made. (This will most likely happen in instances where the remote gateway uses AES-128 for encryption, and AES-128 is not included in the small proposals.)

A greater number of proposals can result in larger UDP packets. These larger packets are once again fragmented at the Data Link Layer of the TCP/IP stack on the client, and then discarded by the hide NAT device that does not support fragmentation. In the case of AES-128, this method of encryption can be included in the small proposals by defining AES-128 as the preferred method.

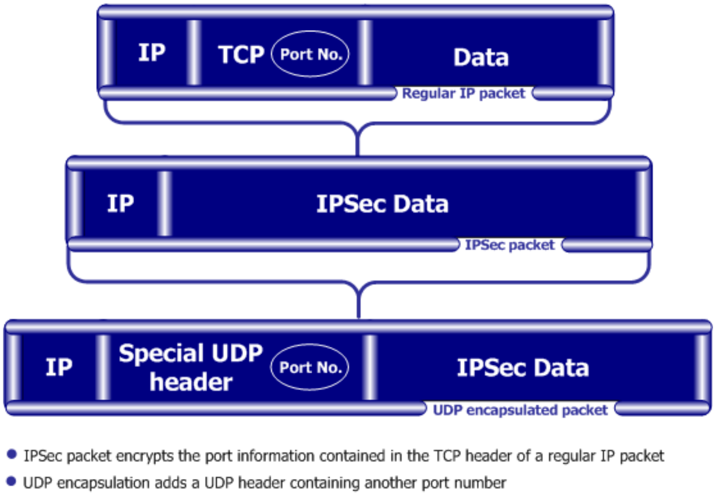
During IPSec

NAT Traversal (UDP Encapsulation for Firewalls and Proxies)

Having successfully negotiated IKE phases I and II, we move into the IPSec stage. Data payloads encrypted with (for example) 3DES and hashed (for integrity) with MD5, are placed within an IPSec packet. However, this IPSec packet no longer contains a TCP or UDP header. A hide NAT device needs to translate the port information inside the header. The TCP/UDP header has been encrypted along with the data payload and can no longer be read by the NATing device.

A port number needs to be added; UDP Encapsulation is a process that adds a special UDP header that contains readable port information to the IPSec packet:

Figure 31-2 UDP Encapsulation:


IP
TCP
Port No.
Data
Regular IP packet
IP
IPSec Data
IPSec packet
IP
Special UDP header
Port No.
IPSec Data
UDP encapsulated packet
IPSec packet encrypts the port information contained in the TCP header of a regular IP packet
UDP encapsulation adds a UDP header containing another port number

The new port information is not the same as the original. The port number 2746 is included in both the source and destination ports. The NAT device uses the source port for the hide operation but the destination address and port number remains the same. When the peer gateway sees 2746 as the port number in the destination address, the gateway calls a routine to decapsulate the packet.

IPSec Path Maximum Transmission Units

IPSec Path MTU is a way of dealing with IPSec packet fragmentation. The Data Link layer imposes an upper limit on the size of the packets that can be sent across the physical network, *the Maximum Transmission Unit*, or MTU. Before sending a packet, the TCP/IP stack of the operating system queries the local interface to obtain its MTU. The IP layer of the TCP/IP stack compares the MTU of the local interface with the size of the packet and fragments the packet if necessary.

When a remote client is communicating across multiple routers with a gateway, it is the smallest MTU of *all* the routers that is important; this is the *path MTU* (PMTU), and for remote access clients there is a special *IPSec PMTU* discovery mechanism to prevent the OS of the client from fragmenting the IPSec packet if the IPSec packet is too large.

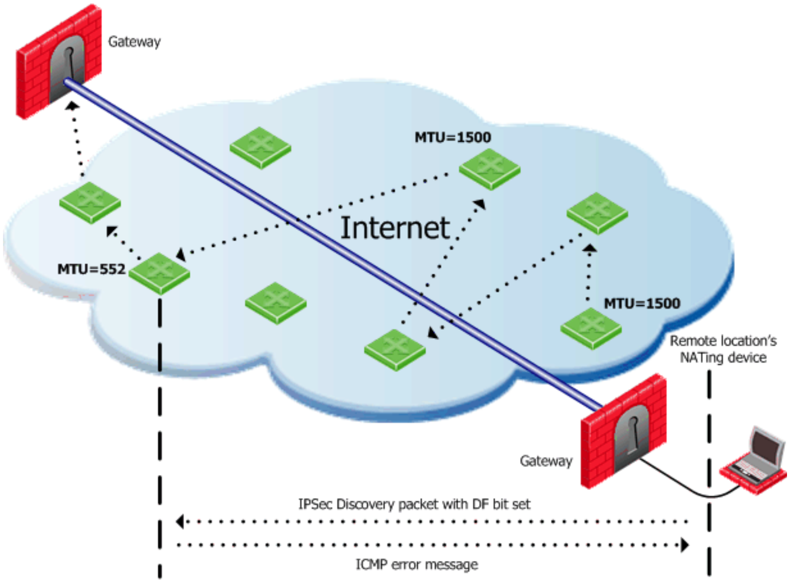
However, the PMTU between the remote client and the gateway will not remain constant, since routing across the Internet is dynamic. The route from gateway to client may not be the same in both directions, hence each direction may have its own PMTU. VPN handles this in two ways:

- Active IPSec PMTU
- Passive IPSec PMTU

Active IPSec PMTU

After IKE phase II but before the IPSec stage, the remote access client sends special discovery IPSec packets of various sizes to the gateway. The DF (do not fragment) bit on the packet is set. If a packet is longer than any router's MTU, the router drops the packet and sends an ICMP error message to the remote client. From the largest packet not fragmented, the remote client resolves an appropriate PMTU. This PMTU is not conveyed directly to the OS. Unknown to the operating system, during the TCP three-way handshake, the Maximum Segment Size (MSS) on the SYN and SYN-ACK packets are changed to reflect the PMTU. This is known as *Active IPSec PMTU*.

Figure 31-3 IPSec discover packets


Gateway
MTU=1500
Internet
MTU=552
MTU=1500
Remote location's NATing device
Gateway
IPSec Discovery packet with DF bit set
ICMP error message

Passive IPSec PMTU

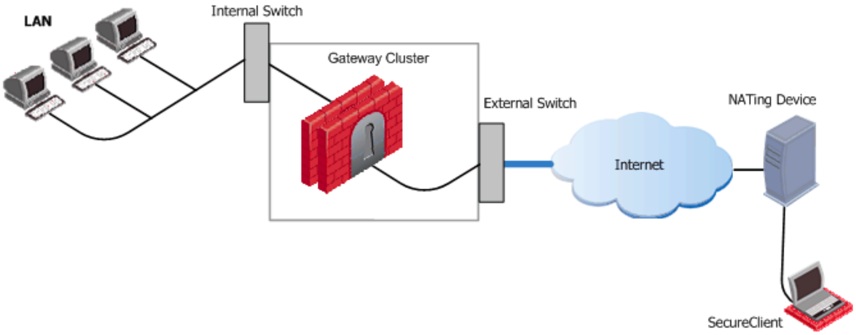
Passive IPSec PMTU solves the problem of dynamic Internet routing. Passive IPSec PTMU is a process that occurs when either side receives an ICMP error message resulting from a change in the routing path. Since routes change dynamically on the Internet, if a different router needs to fragment the packet that has the DF bit set, the router discards the packet and generates an ICMP "cannot fragment" error message. The error message is sent to the VPN peer that sent the packet. When the peer receives this error message, the peer decreases the PMTU and retransmits.

Note - From the system administrator's perspective, there is nothing to configure for PMTU; the IPSec PMTU discovery mechanism, both active and passive, runs automatically.

NAT and Load Sharing Clusters

In Figure 31-4, the remote client is behind a NATing device and connecting to a load-sharing cluster:

Figure 31-4 NAT & Load Sharing Clusters



For the connection to survive a failover between cluster members, the "keep alive" feature must be enabled in **Global Properties > Remote Access > Enable Back connections from gateway to client**

This is also true if the NATing is performed on the gateway cluster side.

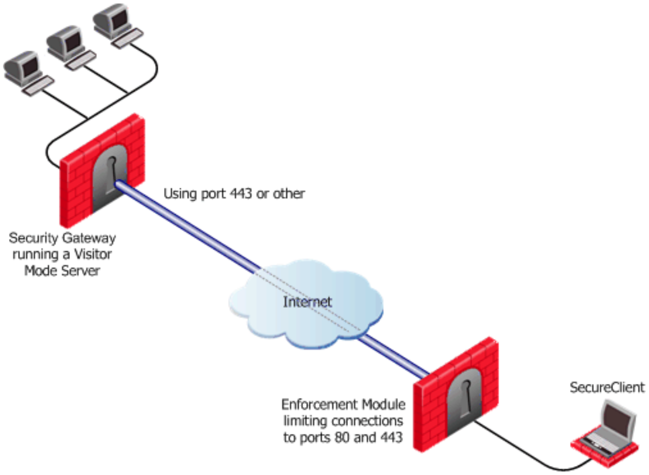
Overcoming Restricted Internet Access

When a user connects to the organization from a remote location such as hotel or the offices of a customer, Internet connectivity may be limited to web browsing using the standard ports designated for HTTP, typically port 80 for HTTP and port 443 for HTTPS. Since the remote client needs to perform an IKE negotiation on port 500 or send IPSec packets (which are not the expected TCP packets; IPSec is a different protocol), a VPN tunnel cannot be established in the usual way. This issue is resolved using **Visitor Mode**, formally known as *TCP Tunneling*.

Visitor Mode

Visitor Mode tunnels *all* client-to-gateway communication through a regular TCP connection on port 443.

Figure 31-5 Visitor Mode


Using port 443 or other
Security Gateway running a Visitor Mode Server
Internet
Enforcement Module limiting connections to ports 80 and 443
SecureClient

All required VPN connectivity (IKE, IPsec, etc.) between the Client and the Server is tunneled inside this TCP connection. This means that the peer gateway needs to run a Visitor Mode (TCP) server on port 443.

Note -

- Even if the remote location's gateway in Figure 31-5 is not a Check Point product (a gateway from another vendor) Visitor mode will still tunnel a connection through it.
- While in Visitor Mode, you can not define a new site.
- Topology update takes place only if the last connection used a profile that enabled Visitor Mode.

Number of Users

To obtain optimal performance of the Visitor Mode server:

- Minimize the number of users allowed Visitor Mode if performance degrades
- Increase the number of sockets available on the OS by editing the appropriate values, for example the socket descriptor on Linux systems

Allocating Customized Ports

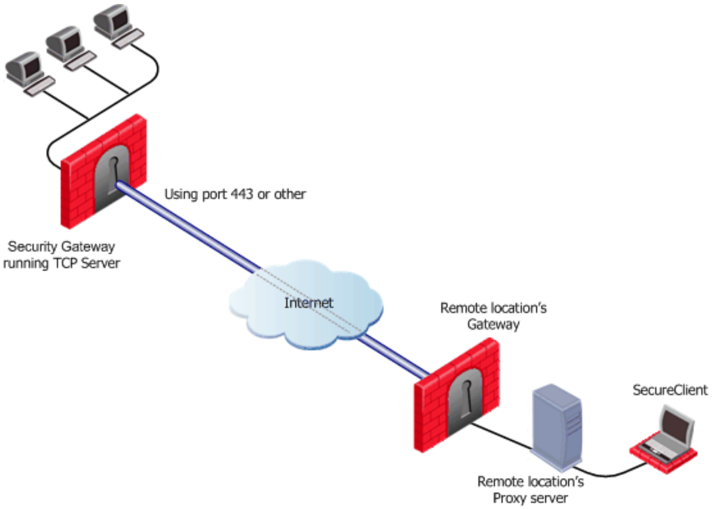
The organization decides that it would like to use a customized port for the Visitor Mode Server other than the typically designated port 443. In this scenario, another port that is *mutually agreed* upon by *all* the remote locations and the home organization, can be used for Visitor Mode. This solution works well with business partners; the partner simply agrees to open a port for the visitor Mode connections. If the chosen port is not represented by a pre-defined service in SmartDashboard, this service must be created in order for the port to be used. If a port has been mutually agreed upon, and there is a proxy, configure the proxy to allow traffic destined to this port.

Note - All partner gateways must agree on the *same* allocated port, since the visitor Mode server on the peer gateway will be listening on only one port.

Visitor Mode and Proxy Servers

Visitor Mode can still be utilized in instances where the remote location runs a proxy server. In this scenario, the remote user enables Visitor Mode connections to pass through the proxy server.

Figure 31-6 Incorporating a Proxy Server


Using port 443 or other
Security Gateway running TCP Server
Internet
Remote location's Gateway
SecureClient
Remote location's Proxy server

Visitor Mode When the Port 443 is Occupied By an HTTPS Server

If the designated port is already in use, for example reserved for HTTPS connections by a Server at the organization's gateway, a log is sent "**Visitor Mode Server failed to bind to xxx.xxx.xxx.xxx:yy (either port was already taken or the IP address does not exist)**" to Security Management server.

If the peer gateway is *already* running a regular HTTP server that also listens on the standard HTTPS port 443, then it must be set up with two external interfaces, both of which have public IP addresses — one for the HTTP server, and one for the Visitor Mode server. This second routable address can be achieved in two ways:

- installing an additional network interface for the Visitor Mode server, *or*
- by utilizing a virtual IP on the same network interface which is blocking the port.

On the gateway object running the Visitor Mode server, **General Properties > Remote Access page >** there is a setting for **Allocated IP address**. All the available IP addresses can be configured to listen on port 443 for Visitor Mode connections.

Visitor Mode with SecurePlatform/Nokia

SecurePlatform running on Linux and Nokia boxes are installed with a pre-configured HTTPS server; the server runs on the gateway and listens on port 443. Installing an additional network interface or utilizing a virtual IP for the Visitor Mode server is not relevant since these HTTPS servers automatically bind to all available IP addresses.

In this case, it is preferable to reserve 443 for Visitor Mode, since users connecting, for example, from a hotel, may only be allowed to connect via ports 80 and 443. These pre-configured HTTPS servers need to be allocated ports that do not conflict with the Visitor Mode server.

Visitor Mode in a MEPed Environment

Visitor Mode also works in a MEPed environment. For more information, see: “Visitor Mode and MEP” on page 599.

Interface Resolution

For *interface resolution* in a Visitor Mode environment, it is recommended to use static IP resolution or dedicate a single interface for Visitor Mode.

Note - Visitor mode is only supported for Internet Explorer 4.0 and up.

Configuring Remote Access Connectivity

In this Section:

Configuring IKE Over TCP

1. For the gateway, open **Global Properties > Remote Access** page **> VPN-Basic** sub-page **> IKE over TCP** section. Select **Gateways support IKE over TCP**.
2. Enable IKE over TCP in a connection profile; the remote user works in connect mode to automatically receive the profile. To configure:
   a. From the file menu, **Manage > Remote Access > Connection profiles...** the **Connection Profiles** window opens. Click **New...**
   b. **Connection Profile Properties** window opens. On the **Advanced** tab, select **Support IKE over TCP**.

If the user is not working in connect mode, the user has to manually enable IKE over TCP on the client.

When IKE over TCP is enabled on the gateway, the gateway continues to support IKE over UDP as well. For remote clients, IKE over TCP is supported only for as long as the client works with a *profile that enables* IKE over TCP.

Configuring Small IKE phase II Proposals

Small phase II IKE proposals always include AES-256, but not AES-128. Suppose you want to include AES-128 in the small proposals:

1. Open the command line database editing tool **DBedit**. There are two properties that control whether small proposals are used or not, one for pre-*NG with Application Intelligence*, the other for *NG with Application Intelligence*.
   - **phase2_proposal** - determines whether an old client (pre-*NG with Application Intelligence*) will try small proposals - default “false”.
   - **phase2_proposal_size** - determines whether a new client (for *NG with Application Intelligence*) will try small proposals - default “true”.
2. In **Global Properties** > **Remote Access** page > **VPN -Advanced** subpage > **User Encryption Properties** section, select **AES-128**. This configures remote users to offer AES-128 as a small proposal.

Configuring NAT Traversal (UDP Encapsulation)

On the gateway network object, enable UDP encapsulation, and decide on a port to handle UDP encapsulation:

1. **General Properties > Remote Access** page **> NAT Traversal** section, select **Support NAT traversal mechanism (UDP encapsulation)**.
2. From the **Allocated port** drop-down box, select a port. **VPN1_IPSec_encapsulation** is the default.
3. IKE phase II proposals are offered both with and without UDP encapsulation when dealing with remote access. (There is no UDP encapsulation between gateways). There is no need to enable UDP on the client unless you want to shorten the existing small IKE phase II proposals. Enable UDP encapsulation in a connection profile; the remote user works in connect mode to automatically receive the profile. To configure:
   a. From the file menu, **Manage > Remote Access > Connection profiles...** the **Connection Profiles** window opens. Click **New...**.
   b. **Connection Profile Properties** window opens. On the **Advanced** tab, select **Force UDP Encapsulation**.

If the user is not working in connect mode, the user has to manually enable UDP Encapsulation on the client. On the client’s file menu, **Tools > Advanced IKE Settings**, select **Force UDP Encapsulation**.

Selecting UDP encapsulation on the gateway means that the gateway supports both encapsulated VPN traffic and traffic that is not encapsulated.

Note - Microsoft L2TP IPSec clients cannot work with Check Point gateways when UDP encapsulation is required.

Configuring Visitor Mode

Visitor Mode requires the configuration of both the Server and the Client. See also: "Visitor Mode and MEP" on page 599

Server Configuration

To enable the TCP tunnelling feature on the security gateway:

On the gateway object running the Visitor Mode Server, **Remote Access** page > **Visitor Mode** section, select **Support Visitor Mode**.

- If port 443 is the assigned port for TCPT server, do not change the **tcp https** default in the **Allocated Port** section.
- If a customized port (other than the default port) is agreed upon, from the drop-down menu select the service that corresponds to this port. If the chosen port is not represented by a pre-defined service in SmartDashboard, create this service.
- In **Allocated IP Address** the default is **All IPs**. To avoid port conflicts, select the appropriate routable valid IP for the Visitor Mode server. If the server has **Dynamic Interface Resolving Configuration...** enabled (on the **VPN - Advanced** page) it is recommended to allocate a specific address for visitor mode instead of **All IPs**.

Note - When Visitor Mode is activated on the gateway, the RDP interface discovery mechanism does not work. A Visitor Mode handshake is used instead.

These settings configure a Visitor Mode server to run on the gateway.

Visitor Mode and Gateway Clusters

Cluster support is limited. The high availability and Load Sharing solutions must provide "stickiness". That is, the visitor mode connection must always go through the same cluster member.

Failover from cluster member to cluster member in a High Availability scenario is not supported.

Enabling Visitor Mode Using a Connection Profile

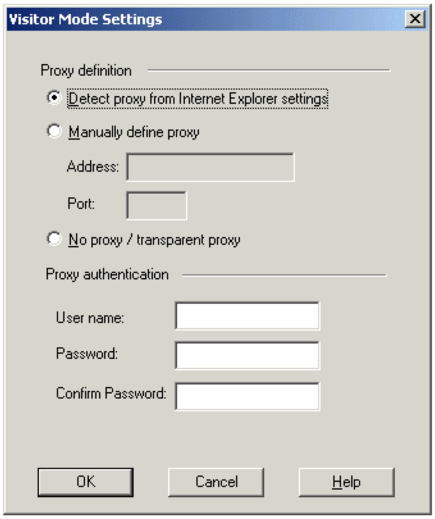
Create a customized connection profile for Visitor Mode users. This profile enables the Visitor Mode feature on the Client side. To create the profile:

1. In SmartDashboard, **Manage** > **Remote Access** > **Connection profiles**... the **Connection Profiles** window opens.
2. Click **New...** to create a new connection profile or **Edit...** to alter an existing profile. The **Connection Profile Properties** window opens.
3. On the **Advanced** tab, select **Visitor Mode**.

On the remote client, configure the user to work in connect mode.

Configuring Remote Clients to Work with Proxy Servers

1. In SecureClient, select **Detect Proxy from Internet Explorer Settings**



Visitor Mode Settings

Proxy definition

● Detect proxy from Internet Explorer settings

○ Manually define proxy

Address:

Port:

○ No proxy / transparent proxy

Proxy authentication

User name:

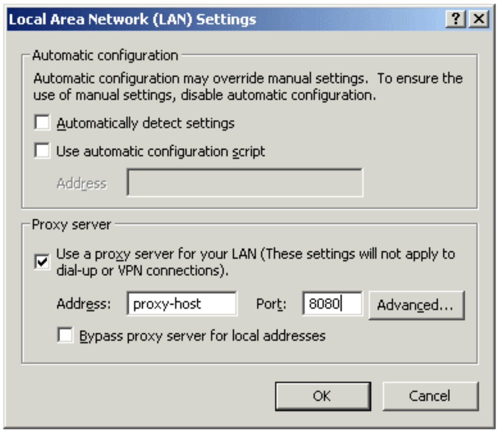
Password:

Confirm Password:

In previous versions, the proxy had to be manually defined.

2. Provide a username and password for proxy authentication. This information is latter transferred with the “connect” command to the proxy server.

Figure 31-7 Proxy settings in Internet Explorer



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

☐ Automatically detect settings

☐ Use automatic configuration script

Address

Proxy server

☑ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: proxy-host Port: 8080

☐ Bypass proxy server for local addresses

Now Secure Client can read any of the settings shown in Figure 31-7 but only if:

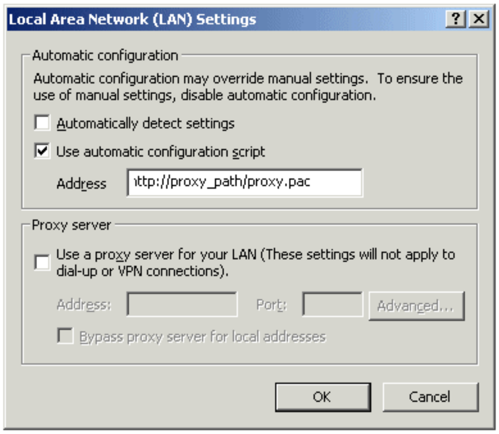
- SecureClient is connected to a LAN or WLAN (not dial-up)
- Secure Domain Logon (SDL) is *not* enabled.

Note - Visitor mode attempts to connect to the proxy server without authenticating. If a user name and password is required by the proxy, the error message “proxy requires authentication appears”.

Windows Proxy Replacement

If SecureClient is on a LAN\WLAN and a proxy server is configured on the LAN, SecureClient replaces the proxy settings so that new connections are not sent to the VPN domain via the proxy but go directly to the LAN\WLAN’s gateway. This feature works with and without Visitor Mode. SecureClient must be on a WAN\WLAN and not using a dial-up connection.

When SC replaces the proxy file, it generates a similar plain script PAC file containing the entire VPN domain IP ranges and DNS names (to be returned as “DIRECT”). This file is stored locally, since the windows OS must receive this information as a plain script PAC file. This file replaces the automatic configuration script as defined in Internet Explorer:



Local Area Network (LAN) Settings

Automatic configuration

Automatic configuration may override manual settings. To ensure the use of manual settings, disable automatic configuration.

□ Automatically detect settings

☑ Use automatic configuration script

Address: ıttp://proxy_path/proxy.pac

Proxy server

□ Use a proxy server for your LAN (These settings will not apply to dial-up or VPN connections).

Address: Port: Advanced...

□ Bypass proxy server for local addresses

OK Cancel

Special Considerations for Windows Proxy Replacement

Sensitive information regarding the site’s IP Address and DNS settings are contained in SecureClient’s `userc.C` file. For this reason, the file is obfuscated by an algorithm that hides the real content (but does not encrypt it). When the proxy replacement feature is used, the same information is written to the plain text PAC file. For this reason, administrators should be aware that the Windows Proxy Replacement feature exposes the VPN domain by writing Site IP addresses and DNS settings as Java Script code in this plain text PAC file, which can be viewed by any end user.

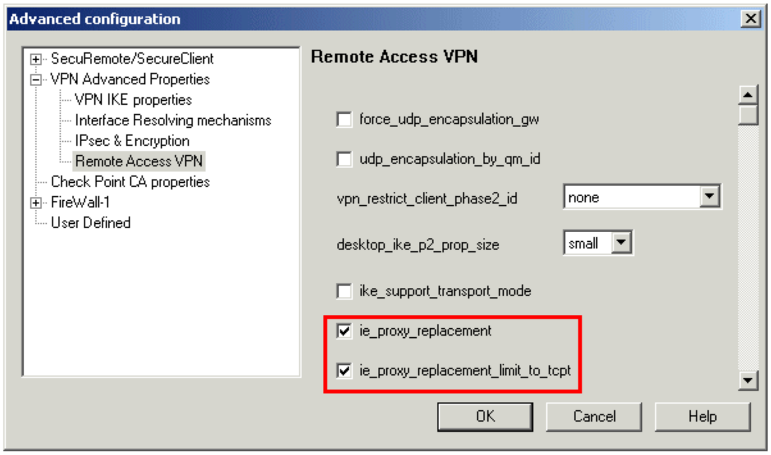
Configuring windows Proxy Replacement

Windows proxy replacement is configured either on the gateway or SecureClient Client.

On the gateway:

1. **Global Properties > SmartDashboard Customization**
2. Click **Configure**

The **Advanced Configuration** window opens:



☐ force_udp_encapsulation_gw

☐ udp_encapsulation_by_qm_id

vpn_restrict_client_phase2_id: none

desktop_ike_p2_prop_size: small

☐ ike_support_transport_mode

☑ ie_proxy_replacement

☑ ie_proxy_replacement_limit_to_tcpt

3. Select either:
   - **ie_proxy_replacement**. If option is selected, windows proxy replacement is always performed, even if visitor mode is not enabled.
   - **ie_proxy_replacement_limit_to_tcpt**. If this option is selected, then proxy replacement takes place *only* when visitor mode is enabled.

When SecureClient performs an update, the policy regarding windows proxy replacement is downloaded and put into effect.

On SecureClient

Alternatively, these two properties can be set in the `userc.c` file on the remote client:

```
:ie_proxy_replacement (true)
:ie_proxy_replacement_limit_to_tcpt (true)
```

Chapter 32
Clientless VPN

In This Chapter

The Need for Clientless VPN

While VPN and SecuRemote technologies provide a comprehensive solution to the problem of securing and protecting data, there are instances where the administrator needs to enable secure communication with client machines that are not configured for VPN. For example, an employee might unexpectedly need to log into the company mail server from an Internet cafe. The employee cannot install software on the browser or even configure it, yet the connection to the company mail server must still be secure, and the employee must have some means of authentication.

Consider a software company wishing to provide its beta sites with access to bug tracking software via a web server. The beta peers do not employ VPN technology. The connection must be secure and the beta users must be authenticated.

Another scenario: an e-commerce company is suffering attacks over the Internet. The company needs to be able to monitor incoming packets for harmful content, yet packets in a standard SSL-based connection passing *directly* between client and server are encrypted. The administrator must both secure this connection and make the packets available for monitoring.

The common factor in these scenarios is that a secure connection must be established in a situation where VPN technology is not available on the client side. A solution is required that:

- Supports secure connections
- Does not involve installation or configuration of software on the client side
- Allows users to be authenticated
- Allows packets to be inspected by the gateway

The Check Point Solution for Clientless VPN

Securing connections when VPN technology is not available to the client (and software can be neither installed or configured on the client side) is accomplished through the use of Clientless VPN. Clientless VPN provides secure SSL-based communication between clients and servers that support HTTPS. In addition:

- The gateway accepts any encryption method that is proposed by the client and supported in VPN
- The gateway can enforce the use of strong encryption, for example 3DES
- Clientless VPN supports user authentication

How it Works

Clientless VPN connections can be broken down into two clear phases:

- *Establishing a Secure Channel*
- *Communication Phase*

Establishing a Secure Channel

A secure connection is established in the following way:

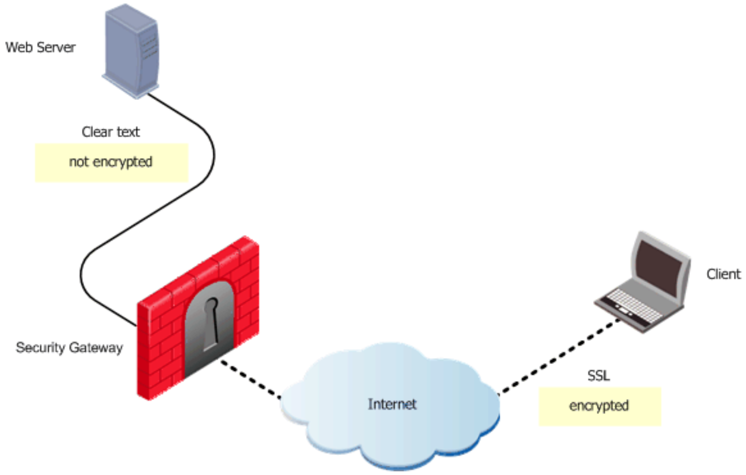
1. The client’s browser makes an HTTPS request to the web server.
2. The request reaches the security gateway.
3. The security gateway checks the Security Policy to see if the connection matches a rule for HTTPS.
4. If a rule is matched, and Clientless VPN is enabled on the gateway, the gateway diverts the connection to the Clientless VPN Security Server.

   Clientless VPN makes use of a special Security Server on the gateway. The Clientless VPN Security Server is a daemon process invoked by the gateway to handle Clientless VPN connections. Once invoked, the Clientless VPN Security Server continues to run as a background process.
5. SSL negotiation takes place between the client and the security gateway, during which the gateway authenticates itself to the client. The gateway uses a certificate signed by a Certificate Authority that the client trusts.
6. A secure channel is established between the security gateway and the client.

Communication Phase

In Figure 32-1, the VPN Security Server opens a connection to the web server. The client connects to the server via the security gateway. From now on, all connections from the client to the security gateway are encrypted. Every packet is sent encrypted to the gateway. The gateway decrypts the packets and forward the packets “in clear” to the web server. From the client’s perspective, the client and web server appear to be communicating directly, but in fact the SSL secure channel terminates at the gateway.

Figure 32-1 Communication phase



Content Security with Clientless VPN

Since the security gateway now sees the packet in clear, the packet can be inspected for content, if content security is required.

User Authentication

Users can authenticate if this is required. Consider the example the employee needing to read company email from an Internet cafe. Typically:

1. The user is presented with a pop-up window requesting a user name and password.
2. The user supplies login details.
3. The gateway verifies the user name and authenticates the password.

Alternatively, the user can authenticate through the use of certificates.

User certificates

Clientless VPN supports the use of certificates for user authentication. The client’s certificate is validated during the SSL phase. In addition, the client’s identification details are extracted from the certificate and then processed during the user authentication phase. In this way, the user does not have to enter a user name and password or keep authenticating on each connection. Authentication is handled through the certificates.

Special considerations for Clientless VPN

There are a number of considerations for Clientless VPN:

- Which certificate does the gateway present?
- How many Security Servers should be run?
- What level of encryption is required?

Certificate Presented by the Gateway

This consideration is related to the certificate the gateway presents to the client in order to authenticate itself during the SSL negotiation. The easiest option, from the client side, is for the gateway to present to the client a certificate signed by a CA that the client is configured to trust by default, for example a certificate supplied by *Verisign*. This requires no configuration on the client side. But if the company has a policy that requires a much higher level of security, it might be preferable for the certificate to come from a CA owned by the company itself.

If the administrator decides to use a certificate from the internal CA (known and trusted) the CA certificate must be supplied to the client and the client configured to trust it. If the administrator decides to work with an external Certificate Authority, the administrator must:

1. Obtain the CA certificate.
2. Configure the Security Management server to trust this CA.
3. Obtain and configure a certificate for the gateway.
4. Supply the CA certificate to the client.

The administrator must now instruct the user to configure the client to trust this CA.

Number of Security Servers to Run

In order to balance the load when there are many Clientless VPN connections, the administrator will have to decide how many Security Servers to run. Up to ten Security Servers can be run on the same gateway. Check Point recommends running one VPN Security Server per 150 *active users*. Be careful not to confuse *active users* with *all registered user*s. For example, if you have 700 registered Clientless VPN users in the database yet only an average of 70 users make Clientless VPN connections concurrently, run only one Security Server.

Level of Encryption

While the gateway can be configured to enforce the use of strong encryption such as 3DES, the use of 3DES can affect performance of the gateway machine. The client browser must also support 3DES.

Configuring Clientless VPN

For the most part, Clientless VPN is configured on the gateway. Exceptions occur when:

- The gateway authenticates itself to the client using a certificate which is not one of the client's default certificates.
- User authentication is required, and that authentication is done through the use of certificates.

In both cases, the relevant certificates must be supplied to the client out of band and the client configured to work with them.

Configuring the Gateway

General Overview

To configure the gateway for Clientless VPN:

- Obtain a certificate for the gateway; this is the certificate the gateway uses to authenticate itself to the client during the initial connection.
- Configure the gateway for Clientless VPN on the **Clientless VPN** page of the gateway object.
- Define users and user authentication schemes.
- Create appropriate rules in the Security Policy Rule Base.

Implementation

Obtaining a Certificate for the Gateway

If you decide to use a certificate issued by an external Certificate Authority instead of the certificate issued by the internal CA:

- Configure VPN Pro to trust this CA
- Obtain and configure a certificate for the gateway

For more information, refer to *Third Party PKI*.

Configuring Clientless VPN on the Gateway Object

On the **Clientless VPN** page of the gateway object's property window:

1. Select **Support Clientless VPN.**
2. Select a **Certificate for gateway authentication**.
3. Select an option for **Client authentication**:
    - Select **Require the client to present a certificate** if user authentication is required and that authentication is performed through the use of certificates.
    - Select **Ask the client to present a certificate** if user authentication is required but you know that not all of the Clientless VPN users have certificates, and that some are authenticating in other ways.
    - Select **Do not ask the client to present a certificate** if user authentication through certificates is *not* required or performed through other means.
4. **Number of concurrent servers/processes** refers to the number of VPN Security Servers to run. Depending on the load, the administrator can run more than one Security Server.

    See the section on "Number of Security Servers to Run" on page 702.
5. **Accept 3DES for Clientless VPN connections** when strong encryption must be enforced.

Defining Users

If user authentication is required:

1. Define users, either in the internal database or external LDAP server.

   For more information, refer to the *Security Management server Administration Guide*.

2. Enable authentication methods by:

   - Configuring the gateway to *support* the required authentication methods
   - Configuring the appropriate authentication methods for users, for example user name/password or through the use of certificates.

     If the user authenticates through certificates, obtain the certificate from the relevant CA and supply the user with the certificate. If the certificate is a certificate issued by the Internal CA, refer to *VPN for Remote clients* on how to issue these certificates.
   - Configuring authentication servers, such as RADIUS, if authentication servers are employed.

   Refer to: *Authentication* in the *Firewall Administration Guide*.

Creating Appropriate Rules in the Security Policy Rule Base

A requirement of Clientless VPN is that, for every connection, the Security Server must be employed. This fact influences how rules allowing Clientless VPN are created in the Security Policy Rule Base.

For no User Authentication

If user authentication is not required, a rule *must* be defined that implements the URI resource. For example:

Table 32-1

Source	Destination	Service	Action
Any	Web server	HTTPS - URI resource	Accept

The rules states that when any source attempts an HTTPS connection to the web server the connection is accepted. A “URI resource” is required in order to force the Security Server to be employed on every connection, as required by Clientless VPN.

For more information on defining URI resources refer to Content Security in the *Firewall Administration Guide*.

For User Authentication

In the Security Policy Rule base, define a rule for either *Client authentication* or *User authentication*:

Table 32-2

Source	Destination	Service	Action
Any	Web server	HTTPS	User Auth

The rule states that when any source tries to connect to the web server using HTTPS, then user authentication must be performed. Select User Auth as the action if you need users to be authenticated but *no* content security performed on incoming packets.

Alternatively:

Table 32-3

Source	Destination	Service	Action
Any	Web server	HTTPS - URI resource	Client Auth

The rule states that when any source tries to connect to the web server using HTTPS, then client authentication must be performed. Select Client Auth as the action if users need to be authenticated and incoming packets inspected for content.

Configuring the Client

If one of the following conditions are true:

- The Security Server authenticates itself to the client using a certificate which is not signed by one of the client’s default Certificate Authorities or
- If user authentication is required, and that authentication is performed via certificates.

then these certificates must be supplied to the client out of band, and the client’s browser configured to use them. Otherwise, no configuration needs to be performed on the client side.

Configuring the Client's Browser

The certificate can be installed on the client's browser in two ways:

1. The client receives the certificate on a diskette.
2. The client right-clicks the certificate, selects “Install Certificate”.

Alternatively:

1. The client inserts the diskette containing the certificate.
2. The client opens the browser.
3. From **Tools > Internet options >** select **Contents** tab.
4. Click **certificates**.
5. Click **Import.**

   The Certificate Import Wizard opens.
6. Import the Certificate from the diskette.

Appendix

Appendix A

VPN Command Line Interface

In This Chapter

VPN Commands

The following command lines relate to VPN and are also documented in the *Command Line Interface (CLI)* Guide.

Table A-1 VPN Command Line interface

Command	**Description**
`VPN`	This command and subcommands are used for working with various aspects of VPN. VPN commands executed on the command line generate status information regarding VPN processes, or are used to stop and start specific VPN services.
`vpn accel`	This command performs operations on accelerator cards (encryption only cards, not the full SecureXL cards) and VPNx. VPNx is a software module that takes advantage of multiple CPUs to accelerate VPN operations.
`vpn compreset`	This command resets the compression/decompression statistics to zero.
`vpn compstat`	This command displays compression/decompression statistics.
`vpn crl_zap`	This command is used to erase all Certificate Revocation Lists (CRLs) from the cache.
`vpn crlview`	This command retrieves the Certificate Revocation List (CRL) from various distribution points and displays it for the user.
`vpn debug`	This command instructs the VPN daemon to write debug messages to the log file: `$FWDIR/log/vpnd.elg`.
`vpn drv`	This command installs the VPN kernel (vpnk) and connects it to the FireWall kernel (fwk), attaching the VPN driver to the FireWall driver.
`vpn export_p12`	This command exports information contained in the network objects database and writes it in the PKCS#12 format to a file with the p12 extension.
`vpn macutil`	This command is related to Remote Access VPN, specifically Office mode, generating a MAC address per remote user. This command is relevant only when allocating IP addresses via DHCP.

Table A-1 VPN Command Line interface

Command	Description
vpn mep_refresh	This command causes all MEP tunnels to fail-back to the best available gateway, providing that backup stickiness has been configured.
`vpn nssm_toplogy`	This command generates and uploads a topology (in NSSM format) to a Nokia NSSM server for use by Nokia clients.
`vpn overlap_encdom`	This command displays all overlapping VPN domains. Some IP addresses might belong to two or more VPN domains. The command alerts for overlapping encryption domains if one or both of the following conditions exist: • The same VPN domain is defined for both gateways • If the gateway has multiple interfaces, and one or more of the interfaces has the same IP address and netmask.
`vpn sw_topology`	This command downloads the topology for a SofaWare gateway.
`vpn ver`	This command displays the VPN major version number and build number.
`vpn tu`	This command launches the TunnelUtil tool which is used to control VPN tunnels.

SecureClient Commands

The following commands relate to SecureClient.

Table A-2 SecureClient command line interface

Command	Explanation
`SCC`	VPN commands executed on SecureClient are used to generate status information, stop and start services, or connect to defines sites using specific user profiles.
`scc connect`	This command connects to the site using the specified profile, and waits for the connection to be established. In other words, the OS does not put this command into the background and executes the next command in the queue.
`scc connectnowait`	This command connects asynchronously to the site using the specified profile. This means, the OS moves onto the next command in the queue and this command is run in the background.
`scc disconnect`	This command disconnects from the site using a specific profile.
`scc erasecreds`	This command unsets authorization credentials.
`scc listprofiles`	This command lists all profiles.
`scc numprofiles`	This command displays the number of profiles.
`scc restartsc`	This command restarts SecureClient services.
`scc passcert`	This command sets the user's authentication credentials when authentication is performed using certificates.
`scc setmode <mode>`	This command switches the SecuRemote/SecureClient mode.
Command	**Description**
`scc setpolicy`	This command enables or disables the current default security policy.
`scc sp`	This command displays the current default security policy.
`scc startsc`	This command starts SecureClient services.
`scc status`	This is command displays the connection status.

Table A-2 SecureClient command line interface

`scc stopsc`	This command stops SecureClient services.
`scc suppressdialogs`	This command enables or suppresses dialog popups. By default, suppressdialogs is off.
`scc userpass`	This commands sets the user's authentication credentials -- username, and password.
`scc ver`	This command displays the current SecureClient version.
`scc icacertenroll`	This command enrolls a certificate with the internal CA, and currently receives 4 parameters - site, registration key, filename and password.Currently the command only supports the creation of p12 files.
`scc sethotspotreg`	This command line interface now includes HotSpot/Hotel registration support.

Desktop Policy Commands

The following command lines relate to the Desktop Policy.

Table A-3 Desktop Policy command line interface

Command	Description
`dtps ver`	This command displays the policy server version.
`dtps debug [on|off]`	This command starts or stops the debug printouts to `$FWDIR/log/dtps.elg`
`fwm psload <path to desktop policy file> <target>`	This command loads the desktop policy onto the module. The target is the name of the module where the desktop policy is being loaded and should be entered as it appears in SmartDashboard. This command should be run from the management. For example: fwm psload `$FWDIR/conf/Standard.S Server_1`
`fwm sdsload <path to SDS objects file> <target>`	This command loads the SDS database onto the module. The target is the name of the module where the SDS objects file is being loaded and should be entered as it appears in SmartDashboard. This command should be run from the management. For example: fwm sdsload `$FWDIR/conf/SDS_objects.C Server_1`

Appendix B

Converting a Traditional Policy to a Community Based Policy

In This Chapter

Introduction to Converting to Simplified VPN Mode

Building VPNs using Simplified Mode has many benefits. Simplified Mode makes it possible to maintain and create simpler, and therefore less error prone and more secure VPNs.

Simplified Mode separates the VPN definitions from the Access Control Security Policy. This makes it easier to understand the VPN topology of an organization, and to understand who is allowed to securely communicate with who. In addition, such as VPN routing are supported only with a Simplified Mode Security Policy.

In order to manage all existing policies in a unified way and utilize the latest features of the current release, it is recommended to convert Traditional Mode Security Policies to Simplified Mode. For new policies, it is recommended to use Simplified Mode, the default option.

A security policy configured in Traditional Mode can be converted to the Simplified VPN Mode using the Security Policy Converter Wizard.

After using the converter wizard, it is possible to greatly simplify many security policies by moving rules and grouping rules together.

The process is simple, and both automatic and manual changes are explained here in detail. The intention is to give you the confidence to move your Traditional VPN Policies to the Simplified VPN Mode.

To start the converter wizard, save the Policy, and from the SmartDashboard main menu, select **Policy > Convert to > Simplified VPN...**

How Traditional VPN Mode Differs from a Simplified VPN Mode

A Traditional Mode Security Policy differs from a Simplified Mode Policy in the following ways:

In Traditional VPN Mode, a single rule, with the Encrypt rule action, deals with both access control and encryption. VPN properties are defined per gateway.

In Simplified VPN Mode, the Security Rule Base deals only with access control. In other words, the Rule Base determines only what is allowed. VPN properties, on the other hand, are dealt with per VPN community.

VPN communities are groups of gateways. The community defines the encryption methods for the VPN. All communication between community members is encrypted, and all other communication is not encrypted.

Simplified VPN Mode and communities are described in Chapter 4, “Introduction to Site to Site VPN”.

The simplified VPN policy makes it easier for the administrator to configure a VPN. However, Traditional policies allow VPNs to be created with greater granularity than Simplified policies, because

- Whether or not to encrypt can be defined per rule (source, destination and service)
- Simplified policies requires all the connections between two gateways to encrypted using the same methods, using the Community definitions.

What this means is that after running the wizard, some manual optimization of the Rule Base may be required.

Note - The terms “VPN Domain” and “Encryption Domain” mean the same thing. Usually, “VPN Domain” is used in the context of Simplified policies, and “Encryption Domain” for Traditional policies.

How an Encrypt Rule Works in Traditional Mode

When a Traditional policy is converted to a Simplified policy, an Encrypt rule is converted to rules that use communities. In order to understand the conversion, it is important to understand how an Encrypt rule works.

Figure B-1 will be used to understand the conversion, and the limitations of the conversion process. It shows a VPN between gateways, and the Encryption Domain of each gateway. Net_A and Net_B are the encryption Domain of gateway 1, and Net_D is the encryption Domain of gateway 2.

Figure B-1 A VPN between Gateways, and the Encryption (VPN) Domain of each Gateway

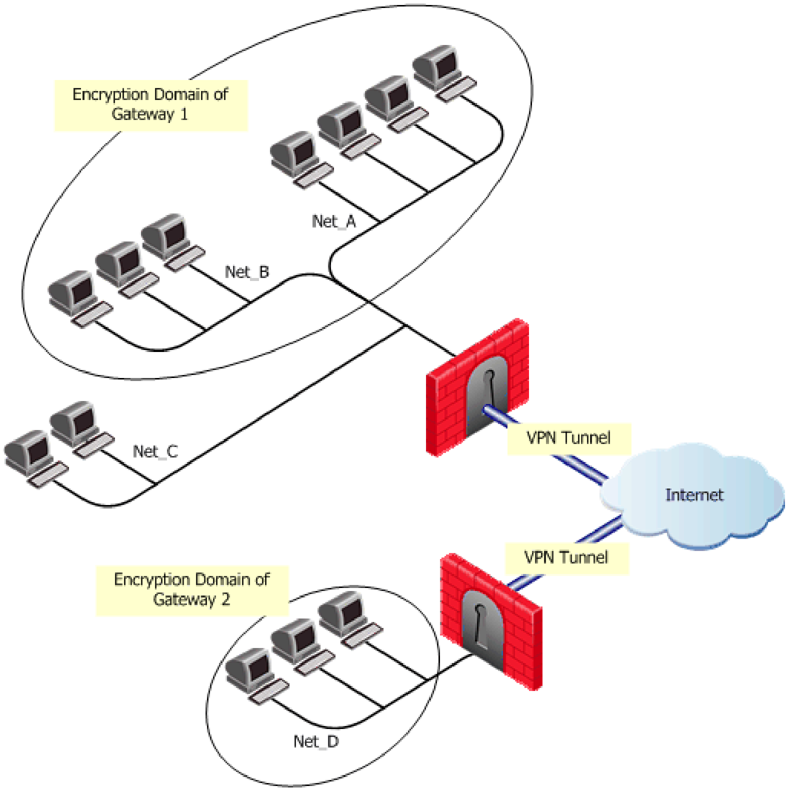

Encryption Domain of Gateway 1
Net_A
Net_B
Net_C
VPN Tunnel
Internet
VPN Tunnel
Encryption Domain of Gateway 2
Net_D

Table B-1 shows how the VPN is implemented in an Encrypt rule.

Table B-1 Sample Encrypt rule in a Traditional Rule Base

Source	Destination	Service	Action	Track	Install On
X	Y	My_Services	Encrypt	Log	Policy Targets

A connection that matches an Encrypt rule is *encrypted* (or *decrypted*) and forwarded by the gateways enforcing the policy. There are two exceptions:

1. If the source or the destination are behind the security gateway, but are not in the VPN Domain of the gateway, the connection is *dropped*.

   For example, referring to Figure B-1 and Table B-1, if Source X is in Net_C and Destination Y is in Net_D, gateway 1 drops the connection. This is because the Action says Encrypt but the connection cannot be encrypted because the source is not in the Encryption Domain of gateway 1.

2. If the source and destination are inside the encryption Domain of the same gateway. In this case, the connection is *accepted in the clear*.

   For example, referring to Figure B-1 and Table B-1, if Source X is in Net_A and Destination Y is in Net_B, the connection originates at X and reaches the gateway, which forwards the response back to Y. The connection is not encrypted because there is no peer gateway for Y that could decrypt the connection. A SmartView Tracker log is issued "`Both endpoint are in the Encryption Domain`".

Principles of the Conversion to Simplified Mode

The converter Wizard attempts to maintain the best possible balance between connectivity and security, by using the following principles:

- Refuse all traffic that could have been refused in traditional Mode. This may mean that some connections may be dropped that were allowed by the traditional rule base.
- Encrypt at least all traffic that would be encrypted in traditional policy. This means that the converted policy may encrypt more connections than the original policy.

What this means is that not all traditional policies can be converted in a way that exactly preserves the policy that is specified in the Security Rule Base. The converted rule(s) in the Simplified VPN can under certain circumstances behave somewhat differently than the encryption rule for Traditional VPNs (described in "How an Encrypt Rule Works in Traditional Mode" on page 720).

Running the converter is a simple two or three step process. After running the wizard, you should review the Security Rule Base to make sure that it has maintained its required functionality, and optimize it if needed.

Placing the Gateways into the Communities

The first step in converting a traditional VPN to a simplified VPN is to create VPN communities that describe the topology of the organization. The conversion wizard requires the administrator to place gateways into communities. It cannot do this automatically because it is very difficult to deduce from the traditional policy what communities should be defined between gateways.

The wizard allows you define communities, and to drag-and-drop gateways into the communities. Referring to Figure B-1, the administrator must make gateway 1 and gateway 2 members of the same community by dragging both the gateway objects into the same site-to-site community object.

You may prefer to create several communities with different encryption properties to reflect the way that the traditional VPN policy works.

If no communities have been previously defined, there are by default two predefined, empty community objects. One is a site-to-site VPN “intranet” community (a Mesh community), and the other is a remote access community. If these are the only two Communities, the wizard gives you the choice of simply placing all gateways into the Site-to-Site Community, and placing all Remote Access gateways into the Remote Access Community.

Conversion of Encrypt Rule

After gateways have been placed into Communities, the Encrypt rules are converted. The converted rule base preserve the behavior of the Encrypt rule in Simplified VPN Mode to the greatest extent possible.

Encrypt rules are converted by the Conversion wizard to two rules:

Table B-2 A converted rule in a simplified Rule Base

Src.	Dest.	VPN	Service	Action	Track	Install On
X	Y	All_GW_to_GW	My_Services	Accept	Log	Policy Targets
X	Y	Any	My_Services	Drop	Log	Policy Targets

The first rule says that the connection is matched and is allowed, if the connection originates at X and its destination is Y, within any Site-to-Site Community.

The second rule says that if a connection originates at X and has the destination Y, but is not encrypted (or decrypted) by any site-to-site community, the connection should be dropped.

The second rule (the Drop rule) is needed where either the source or the destination are not in the VPN Domain. In the Traditional policy, the Encrypt rule would drop this connection. If there were no drop rule in the Simplified policy, the connection may be matched to and allowed by a rule further down in the Rule Base.

When the Converted Rule Base is too Restrictive

This translation of Encrypt rules into two Simplified Mode rule is at least as restrictive as the original rule. However, in the converted Rule Base shown in Table B-2, some connections that were matched to and allowed by the original rule (Table B-1) may be dropped. This may happen with connections between two hosts in the encryption domain of the same gateway. For example, referring to Figure B-1, connections from a node in Net_A to a Node in Net_B will be dropped by the converted Rule Base. This is because community rules define traffic between VPN Domains, and do not relate to traffic within a VPN Domain.

To allow these connections in the converted rule-base, you must explicitly allow them. To do this, add one rule between the first rule and the second rule, for each policy target appearing in the "install on" field. For example, the two Rules in Table B-2 become three rules, as in Table B-3.

Table B-3 Manually Added Rule in the converted Encrypt Rule Base

Src.	Dest.	VPN	Service	Action	Track	Install On
X	Y	All_GW_to_GW	My_Services	Accept	Log	Policy Targets
Net_A	Net_B	Any	My_Services	Accept	Log	Gateway 1
X	Y	Any	My_Services	Drop	Log	Policy Targets

In most cases it is not necessary to add these rules. Only add them when connections inside the encryption domain are matched by the Encrypt rule. An indication of this is the appearance of the log in SmartView Tracker "`Both endpoint are in the Encryption Domain.`"

Conversion of Client Encrypt Rules

Each Client Encrypt rule translates to a single rule that preserves the behavior of the client Encrypt rule. For example, the Traditional Mode rule in Table B-4 allows Remote Access users to access Net_D.

Table B-4 Remote Access Rule in Traditional Mode

Source	Destination	Service	Action	Track
All_Users@alaska	Net_D	My_Services	Client Encrypt	Log

The translated rule is shown in Table B-5. The Remote Access community is put in the VPN field, and the Action of the rule is Accept:

Table B-5 Translated Remote Access Rule in Simplified Mode

Source	Dest.	VPN	Service	Action	Track
All_Users@alaska	Net_D	Remote Access Community	My_Services	Accept	Log

Conversion of Auth+Encrypt Rules

In a Traditional Mode policy, Auth+Encrypt Rules are rules with User, Client or Session Authentication, together with **Add Encryption** selected in the Action of the Rule.

For Auth+Encrypt rules, as shown in Table B-6 with Client Authentication, the Source specifies both a restriction on the source location, and also the authorized users. Any connection matching the rule must be authenticated and encrypted. If encryption is not possible, the connection is dropped.

Table B-6 Auth+Encrypt Rule in Traditional Mode

Source	Destination	Service	Action	Track
All_Users@alaska	Net_D	My_Services	Client_Auth	Log

Since the identification of users is possible only in authentication rules, and not in drop rules, it is not possible to define a rule that drops connections that were not encrypted.

Add the Services that should not be encrypted inside the Community to the Excluded Services list. For example, if you have explicitly defined implied rules in the Traditional Policy. See “How to Authorize Firewall Control Connections in VPN Communities” on page 88.

Because of this, Auth+Encrypt rules cannot be automatically translated in such a way that the translated Rule Base is at least as restrictive as the original rule. Instead, the Converter wizard translates Auth+Encrypt rules to a single rule, and does not add a Drop rule, as shown in Table B-7. This is a security problem, because connections that match the Source location, where the users authenticated successfully, but were not encrypted, may be accepted further down in the translated Rule Base if some later rule specifies Accept for the same Source.

Table B-7 Insecure Translated Auth+Encrypt Rule in Simplified Mode

Source	Dest.	VPN	Service	Action	Track
All_Users@alaska	Net_D	All_GwToGw	My_Services	Client Auth	Log

When the converter encounters Auth+Encrypt rules, it warns the administrator by displaying an error stating that the converter cannot translate such rules automatically. In this case it is important to review the translated rule base before installing it, in order to avoid security breaches. It may be necessary to add rules to make sure that all the traffic that was previously dropped by the original Rule Base is dropped in the translated Rule Base.

How the Converter Handles Disabled Rules

If a rule in the Traditional VPN Rule Base was disabled, the translated rule in the simplified Rule Base will also be disabled.

After Running the Wizard

After running the Wizard, examine the Rule Base to see that it has retained the desired functionality, and if necessary, optimize the Rule Base, and make other changes, as follows. These points have been covered in the earlier discussion, but are summarized here for convenience:

Take out Unneeded Drop Rules

In some cases you can delete the second Drop rule generated by the conversion of an Encrypt rule because it will never match any connection, and the first rule is sufficient. This is the case for rules where the following are true:

- The source and destination are located in the encryption domain of gateways appearing in the “Installed on” column of the rule.
- A Community links all gateways protecting addresses in the Source and also links gateways protecting addresses in the Destination.

Another case where you can delete the second Drop rule generated by the conversion of an Encrypt rule is where connections that do not match the first rule are dropped by rules that appear later in the Rule Base. Sometimes you can group several Drop rules generated by the conversion of several Encrypt rules into a single Drop rule.

Add Rules Allowing Communication Inside the VPN Domain

Connections matching Encrypt rules where both endpoints are located inside the encryption domain of the same gateway, are accepted in a Traditional Rule Base. To achieve the same effect in the simplified rule base, you must manually add rules that accept the traffic inside the encryption domains of the gateways. In most cases it is not necessary to add these rules. Add them if you see the SmartView Tracker log message: "`Both endpoint are in the Encryption Domain`".

Auth+Encrypt Rules

Auth+Encrypt rules are not converted automatically. When such rules appear in the Rule Base, review the converted Rule Base and make sure that the security of these rules are maintained.

Appendix C
VPN Shell

Configuring a Virtual Interface Using the VPN Shell

The VPN Shell, used for creating Virtual VPN Tunnel Interfaces, is composed of menus and commands. The shell can be used interactively or as a single command line. Invoking the command - `vpn shell` - without any other arguments starts the interactive shell. Adding arguments after `vpn shell` is interpreted as a direct command and executed.

`VPN shell` — starts the interactive mode

Expressions and meanings for the vpn shell as shown in Table C-1.

- The basic format of the command is: `[path/path/path arguments]`, for example `interface/add` takes you directly to the menu for adding numbered interfaces.
- Within the vpn shell, command line completion is available, for example `i/a/n` is completed to `interface/add/numbered` and executed provided there are not two commands starting with the same letter.
- Use Control-D to exit the vpn shell/end of line (when including vpn shell commands in a script)

Table C-1 vpn shell commands/arguments

Expression	Meaning
?	Shows available commands
/	Returns to the top of the main menu
.. (two dots)	Moves up one menu level
/quit	Exists the VPN shell
show/interface/summary	Shows summary of all interfaces or of a specific interface
show/interface/detailed	Shows summary of all interfaces or of a specific interface with greater detail
interface/add/numbered	Adds a numbered interface (Local IP, remote IP, peer name and interface name required)
interface/add/unnumbered	Adds an unnumbered interface (Peer name and interface name required)
interface/modify/peer/mtu	Modify the MTU of an interface by peer name

Table C-1 vpn shell commands/arguments

Expression	Meaning
interface/modify/peer/netmask	Modify the netmask of an interface by peer name
interface/modify/ifname/mtu	Modify the MTU of an interface by given interface name
interface/modify/ifname/netmask	Modify the netmask of an interface by given interface name
interface/delete/peer	Delete interface by given peer name
interface/delete/ifname	Delete interface by given interface name
interface/show/summary	Shows summary of all interfaces or of a specific interface
interface/show/detailed	Shows summary of all interfaces or of a specific interface with greater detail
tunnels/show/IKE/all	Displays all valid SA's
tunnels/show/IKE/peer	Displays valid SA for a specific peer (gateway IP address required)
tunnels/show/IPSec/all	Displays all IPSec tunnels
tunnels/show/IPSec/peer	Displays IPSec tunnels for a specific peer
tunnels/delete/IKE/peer	Deletes valid SA's for a specific peer (gateway IP address required)
tunnels/delete/IKE/user	Deletes valid SA's for a specific user (internal IP address and user name required)
tunnels/delete/IKE/all	Deletes all valid SA's
tunnels/delete/IPSec/peer	Deletes IPSec tunnels for a specific peer (gateway IP address required)
tunnels/delete/IPSec/user	Deletes IPSec tunnels for a specific user (internal IP address and user name required)
tunnels/delete/IPSec/all	Deletes all IPSec tunnels
tunnels/delete/all	Deletes all SA's and IPSec tunnels