

**Intel® NetStructure™
Virtual Private Networking
Concepts Guide**

Intel Network Systems, Inc.
December 2000

Disclaimer

Information in this document is provided in connection with Intel® products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in Intel Network System, Inc.'s Terms and Conditions of Sale for such products, Intel Network Systems, Inc. assumes no liability whatsoever, and Intel Network Systems, Inc. disclaims any express or implied warranty, relating to sale and/or use of Intel® products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right. Intel Network Systems, Inc. products are not intended for use in medical, life saving, or life sustaining applications.

Intel Network Systems, Inc. may make changes to specifications and product descriptions at any time, without notice.

This *Intel® NetStructure™ Virtual Private Networking Concepts Guide*, as well as the software described in it is furnished under license and may only be used or copied in accordance with the terms of the license. The information in this manual is furnished for informational use only, is subject to change without notice, and should not be construed as a commitment by Intel Network Systems, Inc. Intel Network Systems, Inc. assumes no responsibility or liability for any errors or inaccuracies that may appear in this document or any software that may be provided in association with this document.

Except as permitted by such license, no part of this document may be reproduced, stored in a retrieval system, or transmitted in any form or by any means without the express written consent of Intel Network Systems, Inc.

Copyright © Intel Network Systems, Inc. 2000. *Other brands and names are the property of their respective owners.

Contents

| | |
|---|------------|
| Intel® NetStructure™ VPN Concepts Guide Overview | 1-1 |
| Intel NetStructure VPN Concepts Guide Overview | 1-1 |
| Intel NetStructure VPN Suite Overview | 1-2 |
| Operational Overview | 1-5 |
| TCP/IP Basics Overview | 1-6 |
| Cryptographic Systems and Encryption Terminology | 2-1 |
| Cryptographic Systems and Encryption Terminology Overview | 2-1 |
| Symmetric Cryptographic Systems | 2-3 |
| Data Encryption Standard (DES) | 2-4 |
| Triple Pass DES | 2-5 |
| 3DES | 2-7 |
| Outer Cipher Block Chaining (CBC) | 2-8 |
| Asymmetric Cryptographic Systems | 2-9 |
| Symmetric Vs. Asymmetric Cryptography | 2-10 |
| Diffie-Hellman Session Key Exchange | 2-11 |
| Key Space and Brute Force Attacks | 2-13 |
| Encapsulation and Packet Handling | 3-1 |
| Encapsulation Overview | 3-1 |
| Secure Profiles | 3-2 |
| ESP Encapsulation | 3-4 |
| SST Encapsulation | 3-6 |
| Packet Handling | 3-7 |
| Packet Keys | 3-8 |
| Authentication Methods | 4-1 |
| Authentication Methods Overview | 4-1 |
| Certificate Authentication | 4-2 |
| Challenge Phrase Authentication | 4-3 |
| SecurID Authentication | 4-4 |
| RADIUS Authentication | 4-5 |
| Entrust Authentication | 4-6 |
| Firewalls and Tunnels | 5-1 |
| Firewall and Tunnels Overview | 5-1 |
| Firewall Functions | 5-2 |
| Filters | 5-6 |
| Tunnel Types | 5-8 |
| Site-to-Site Tunnels | 5-9 |

| | |
|---|------------|
| Single-User Tunnels | 5-12 |
| Multiuser Tunnels | 5-16 |
| Tunnel Modes | 5-20 |
| One-Way In Firewall Rules. | 5-22 |
| One-Way Out Firewall Rules | 5-24 |
| Outbound Proxy | 5-26 |
| Inbound Proxy | 5-28 |
| Tunnel Termination and Firewall Rules | 5-31 |
| Load Balancing and Redundancy. | 6-1 |
| Load Balancing | 6-1 |
| Redundancy. | 6-2 |

Intel NetStructure VPN Concepts Guide Overview

| | |
|--|-----|
| Intel® NetStructure™ VPN Concepts Guide Overview | 1-1 |
| Intel NetStructure VPN Suite Overview | 1-2 |
| Operational Overview | 1-5 |
| TCP/IP Basics Overview | 1-6 |

Intel® NetStructure™ VPN Concepts Guide Overview

The purpose of this Intel NetStructure VPN Concepts Guide is to provide you with information on the Intel NetStructure virtual private networking (VPN) suite, consisting of five modular components that work together to provide secure communications across any network. The term VPN Gateway is used in this document to refer to the LanRover™ VPN Gateway, the LanRover VPN Gateway PLUS, and the Intel NetStructure 3110, 3120, 3125, and 3130 VPN Gateway devices.

In addition, the *Intel NetStructure Virtual Private Networking Concepts Guide* provides background information and theory on topics ranging from firewall functions and cryptographic systems to authentication types and encapsulation.

Contents

| |
|--|
| Intel NetStructure VPN Suite Overview (page 1-2) |
| Operational Overview (page 1-5) |
| TCP/IP Basics Overview (page 1-6) |
| Encapsulation Overview (page 3-1) |
| Packet Handling (page 3-7) |
| Authentication Methods Overview (page 4-1) |
| Cryptographic Systems and Encryption Terminology Overview (page 2-1) |
| Firewall and Tunnels Overview (page 5-1) |
| Load Balancing (page 6-1) |
| Redundancy (page 6-2) |

Intel NetStructure VPN Suite Overview

The Intel NetStructure virtual private networking (VPN) suite consists of four modular components that work together to provide secure communications across any network:

- VPN Gateway
- Intel NetStructure VPN Manager
- Intel NetStructure VPN Client
- Shiva Certificate Authority

VPN Gateway

The VPN Gateway is a hardware/software security system, responsible for processing data packets as they pass between the public side and the private side of a network. The VPN Gateway is designed to perform three major functions:

- At the communications level, the VPN Gateway can act as either a router or as a bridge.
- As a packet encryptor, the VPN Gateway can selectively encrypt and decrypt data based on source and destination addresses and ports. This provides the flexibility of sending both encrypted and clear data using the same infrastructure, without compromising your centrally managed security policy.
- As a firewall, the VPN Gateway can be used as a packet filter and a stateful inspection proxy. The VPN Gateway goes further than traditional firewalls, however, by adding authentication to the firewall function, which allows the creation of truly secure virtual private networks.

The VPN Gateway includes an industry-standard PCI bus card, which accelerates encryption and decryption to Local Area Network speeds. The card incorporates a dedicated ASIC chip optimized for DES and Triple Pass DES encryption and provides a significant increase in throughput over software-only encryption implementations.

Intel NetStructure VPN Manager

The Intel NetStructure VPN Manager is a software package based in Windows* 95 or Windows NT* that centrally monitors and configures the Intel NetStructure VPN Gateway products in your network. Using a powerful graphical user interface (GUI), you can configure and monitor VPN Gateways deployed in the field. The Intel NetStructure

VPN Manager is also used to define and grant access to Intel NetStructure VPN Client users.

Intel NetStructure VPN Client

The Intel NetStructure VPN Client is a software package based in Windows 95 or Windows NT that provides desktop-to-gateway security within a LAN or across any WAN.

Because all Intel NetStructure VPN products operate at the network layer, the Intel NetStructure VPN Client is completely transparent to users and works with any application. With the Intel NetStructure VPN Client, users can dial in to any Internet service provider (ISP) and create a secure channel back to your network, which eliminates the need for expensive dial-in equipment and toll-charges.

Shiva Certificate Authority Server

The Shiva Certificate Authority Server is a software package based in Windows NT or Solaris* that is responsible for certifying and managing the public keys of all Intel NetStructure VPN products, allowing Intel NetStructure VPN components to positively identify each other using unique certificates virtually impossible to forge.

The Shiva Certificate Authority Client is a graphical user interface based in Windows 95 or Windows NT that manages the Shiva Certificate Authority Server.

Intel NetStructure VPN Product Suite

The Intel NetStructure VPN product suite supports the use of secure tokens. These tokens are a tamper-resistant PCMCIA card designed to meet FIPS-140-1 level 2 criteria. The token stores and performs all public key operations while keeping private keys secure from attacks.

The Intel NetStructure VPN products are designed to grow with your network. If you only have a few sites, you can operate them with only a few VPN Gateways. As your network grows, you can add additional VPN Gateways, remote clients, and central management at any time. These components are illustrated next in a typical network configuration.

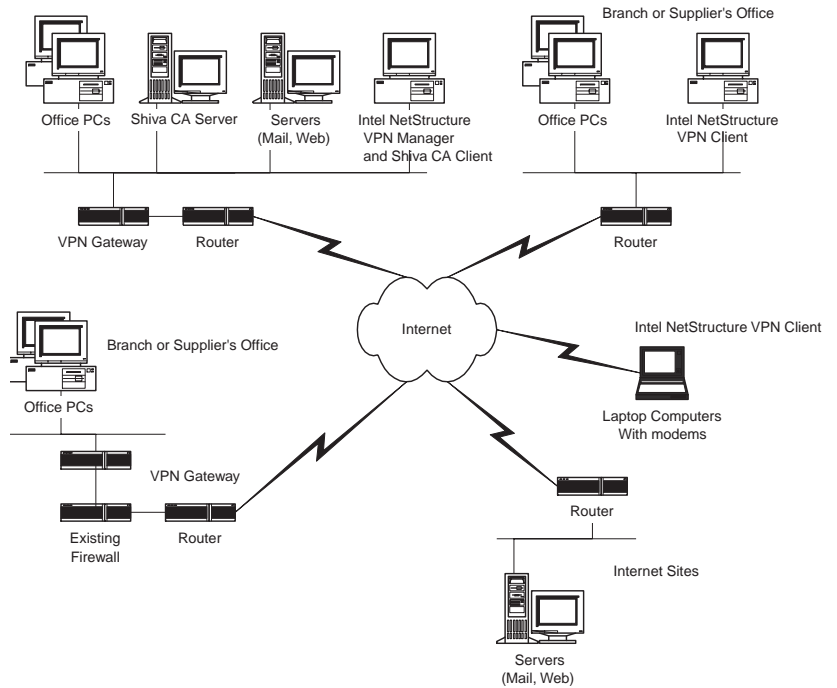


Figure: Typical Network Configuration

Related Information

Operational Overview (page 1-5)

TCP/IP Basics Overview (page 1-6)

Intel NetStructure VPN Concepts Guide Overview (page 1-1)

Operational Overview

The Intel NetStructure VPN components fit into typical network configurations in various locations. VPN Gateways often sit at the gateway between LANs and WANs. All data into and out of a protected LAN passes through the VPN Gateway for processing. The Intel NetStructure VPN Client software package runs on PCs either directly connected to a LAN or remotely located and connect to the WAN by means of a dial-up connection.

VPN Gateways are configured by using the Intel NetStructure VPN Manager (which runs on a Windows 95 or Windows NT workstation), a command line interface from a console, or through a Telnet session from a computer on the VPN's trusted network.

The Shiva Certificate Authority Server runs on a Solaris workstation or Windows NT workstation or server, and is managed through the Shiva Certificate Authority Client. The VPN Gateway can operate in a standalone mode, which means that the Intel NetStructure VPN Manager and the Shiva Certificate Authority do not need to be running or available for the VPN Gateway to function.

Related Information

[Intel NetStructure VPN Concepts Guide Overview \(page 1-1\)](#)

[TCP/IP Basics Overview \(page 1-6\)](#)

TCP/IP Basics Overview

The Intel NetStructure VPN products operate on Transmission Control Protocol/Internet Protocol (TCP/IP) networks. TCP/IP is the foundation of the Internet. To fully appreciate how the Intel NetStructure VPN components work, you need to understand some basic TCP/IP terms.

Packets and Packet Headers

Communications in a TCP/IP network are broken into small chunks called packets. The typical maximum packet size carried over TCP/IP networks is 1500 bytes. Each packet carries some user data called payload. The payload could be part of an e-mail message or a Web page. Every packet also has some control information that indicates where the packet originated, where it is going, and what application should receive it when it arrives. This information is referred to as the packet header. A simplified packet example is shown in the following diagram.

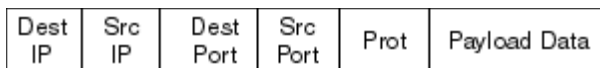


Figure: Simple Packet Diagram

IP Address

All devices on a TCP/IP network must have at least one address called an IP address. This address uniquely identifies the device on a network (actually in the entire world). For example, "Test Company's" Web server has the following IP address: 205.250.128.2.

There are some reserved IP addresses that are never assigned, which are called unroutable. Anyone can use these addresses on a closed network. Well-known unroutable IP addresses start with 10.x.x.x and 192.168.x.x, where x is any number between 1 and 254.

Subnet Mask

One function of a subnet mask is to tell a device what other addresses it can directly communicate with. An example of a subnet mask is 255.255.255.0, which defines a class C subnet. Each component of the subnet mask (either 255 or 0 in the example) is called an octet. A class C subnet mask means that there are 254 addresses with which the device can directly communicate.

For example, "Test Company" is assigned a full class C. This means "Test Company" can use any address between 205.250.128.1 up to

205.250.128.254. The addresses 205.250.128.0 and 205.250.128.255 are also part of the addresses in the class C subnet, but are reserved for broadcasting and cannot be assigned to any devices on the network (often called boundary addresses).

If you want to break your class C into separate networks, you do this by varying the last octet of the subnet mask. If you make your subnet mask 255.255.255.128, your class C is split into 2 parts. This gives you one subnet containing the addresses from 205.250.128.1 to 205.250.128.126 and another subnet containing 205.250.128.129 to 205.250.128.254.

When you work with the full class C, there are 2 boundary addresses reserved for broadcasts. Every subnet requires 2 addresses for broadcasts. When you split your class C into 2 parts, you must still have broadcast addresses in each subnet. The first subnet uses 205.250.128.0 and 205.250.128.127 for broadcasts while the second uses 205.250.128.128 and 205.250.128.255.

When you have the full class C, there are 254 addresses you can use. Once the class C is split into two subnets, there are 126 addresses in each subnet for a total of 252 addresses.

The following values, if placed in the last octet of the subnet mask, divide a class C subnet into smaller subnets.

| Decimal Value (Binary Value) | Number of Subnets | Number of Addresses in Each Subnet |
|---|--------------------------|---|
| 255 (1111-1111) | 254 | 1 |
| 254 (1111-1110) | 128 | 0 |
| 252 (1111-1100) | 64 | 2 |
| 248 (1111-1000) | 32 | 6 |
| 240 (1111-0000) | 16 | 14 |
| 224 (1110-0000) | 8 | 30 |
| 192 (1100-0000) | 4 | 62 |
| 128 (1000-0000) | 2 | 126 |
| 0 (0000-0000) | 1 | 254 |

Note: If you divide your class C into more and more subnets, the number of available addresses becomes smaller and smaller.

Routing Table

When a device creates a packet for transmission, it looks at the destination IP address. If the address is on the same subnet as the device (as defined by the subnet mask), the device looks for the address on its LAN. If the destination device responds, the originating device transmits the packet directly to the destination. However, if the destination device is not found locally, the originating device must decide what to do with the packet.

The rules upon which the device bases the decision are called routes, which are stored in a routing table. The routing table maps network addresses to gateways. Basically, it tells the device that if it has a packet destined for a certain network, the packet should be sent to a specific gateway. The gateway can be any device such as a router or a switch that can send the packet out of the local subnet.

Static routes are entries in the routing table that do not change. They are often defined on routers and switches when network topologies become complex and the network administrator wants to force packets to go in a certain known direction (that is, through a specific gateway). Dynamic routes are entries in the routing table that may change over time. This type of route is usually added automatically, based on some network routing protocol.

Default Gateway

The routing table usually has a route of last resort known as a default gateway. The default gateway is where the originating device sends any packet for which it has no specific rule in its routing table. Most desktop computers do not have static routes added to them and therefore rely on the default gateway being set to be able to communicate outside their local subnet. This implies that the default gateway's IP address must be on the same subnet as the originating device. Computers can directly communicate only with devices on their local subnet (as defined by their IP address and subnet mask).

Default gateways are what make the Internet work. When a packet is created by a desktop computer destined for an address on the Internet, the desktop computer often sends the packet to its default gateway. The default gateway is often an edge router connecting the LAN (on which the desktop computer is sitting) to the Internet. The edge router probably does not have specific routes telling it what to do with the

packet. The edge router, therefore, most likely sends the packet off to its default gateway. This cycle occurs until the packet arrives at a device that knows where to find the destination address.

Application Port

When a computer (or any network device) receives a packet, the computer decides what to do with it. The computer may have many different programs running simultaneously (for example, a mail server and a Web server). Each program expecting to receive or send packets from or to a network opens something called a socket. If you look at an IP address as a street address that identifies a building, then an open socket can be compared to a room number within the building. The number given to a socket is called an application port number.

Each packet contains both a source application port and destination application port in its header. The destination application port number is used by the receiving computer to decide which program should be given the payload of the packet for final processing.

Many application port numbers are standard. Some common numbers are port 80, which is associated with http (www) packets; port 25, which is associated with SMTP mail; port 110 (POP3 mail); port 23 (Telnet); and port 21 (FTP). Therefore, when Web servers start, they usually connect to port 80 and listen for requests to come in. Note that a Web server can be configured to listen on another port, but most follow the standard.

The former Shiva Corporation (now Intel Network Systems, Inc.) registered application port 2233. Packets with the source and destination application ports set to 2233 are encrypted with a LanRover VPN Gateway device.

Related Information

Intel® NetStructure™ VPN Concepts Guide Overview (page 1-1)
Operational Overview (page 1-5)

Cryptographic Systems and Encryption Terminology

| | |
|---|------|
| Cryptographic Systems and Encryption Terminology Overview | 2-1 |
| Symmetric Cryptographic Systems | 2-3 |
| Data Encryption Standard (DES) | 2-4 |
| Triple Pass DES. | 2-5 |
| 3DES | 2-7 |
| Outer Cipher Block Chaining (CBC) | 2-8 |
| Asymmetric Cryptographic Systems | 2-9 |
| Symmetric Vs. Asymmetric Cryptography | 2-10 |
| Diffie-Hellman Session Key Exchange | 2-11 |
| Key Space and Brute Force Attacks | 2-13 |

Cryptographic Systems and Encryption Terminology Overview

When Julius Caesar sent messages to his trusted acquaintances, he did not trust the messengers. So he replaced every A with a D, every B with an E, and so on throughout the alphabet. This was the beginning of cryptography. Only those who knew the "shift by 3" rule could decipher his messages.

A cryptographic system is a method of disguising messages so that only certain people can see through the disguise. Cryptography is the art of creating and using cryptographic systems.

The original message is called a plaintext. The disguised message is called ciphertext. Encryption means any procedure to convert plaintext into ciphertext. Decryption means any procedure to convert ciphertext into plaintext.

The term cryptographic system refers to a set of encryption and decryption algorithms. The algorithms are labeled and the labels are called keys. For example, Caesar probably used "shift by n" encryption for several different values of n. It is natural to say that n is the key here.

Two general types of cryptographic systems exist: symmetric cryptographic systems and asymmetric cryptographic systems.

Encryption

Encryption is a mathematical operation that transforms data from clear text to cipher text. Usually the mathematical operation requires that a key be supplied along with the clear text.

Encryption, therefore, can be expressed as the formula:

$$\text{Cipher Text} = f(\text{Clear Text}, K_e)$$

In this formula, f represents some mathematical operation or algorithm and K_e represents a key.

Decryption is the opposite of encryption, a mathematical operation that transforms cipher text to clear text. Decryption usually requires a key and can be expressed as the formula:

$$\text{Clear Text} = g(\text{Cipher Text}, K_d)$$

In this formula, g represents a mathematical operation, which "undoes" the steps performed by the algorithm f, and K_d represents a key.

Related Information

Symmetric Cryptographic Systems (page 2-3)

Asymmetric Cryptographic Systems (page 2-9)

Symmetric Vs. Asymmetric Cryptography (page 2-10)

Symmetric Cryptographic Systems

A very simple encryption algorithm involves shifting the letters of the alphabet to the right by some offset. For example if you had the clear text "AT" and decided to encrypt this data by shifting each letter 3 letters to the right, you would end up with DW. In this example, the clear text is AT, the key is 3, the algorithm is "shift K letters to the right," and the cipher text is DW. Your encryption formula would look like this:

$$DW = \text{shift-right} (AT , 3)$$

Of course, decryption in this case involves shifting the letters of the cipher text to the left by the same offset used when the data was encrypted. Therefore, your decryption formula would look like this:

$$AT = \text{shift-left} (DW , 3)$$

Note that the key used to encrypt the data is the same key used to decrypt the data.

$$K_e = K_d$$

This algorithm is therefore referred to as symmetric. In this case, the person encrypting the data and the person decrypting the data must both know the same key. The strength of the system relies on the key being kept secret. Symmetric cryptography is therefore often referred to as secret key cryptography.

A real world metaphor for symmetric cryptography is a lock box with a single lock. To safely transfer an object from one person to another, the first person opens the box with a key, puts the object in the box, and then locks the box. The second person needs only a copy of the key, and can then open the box and retrieve the object.

Related Information

Data Encryption Standard (DES) (page 2-4)

Triple Pass DES (page 2-5)

3DES (page 2-7)

Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a well-known and thoroughly tested cryptographic system. The DES algorithm is a very complex symmetric algorithm that specifies that data be encrypted in 64-bit blocks. A 64-bit block of clear text goes into the algorithm along with a 56-bit key. The result is a 64-bit block of cipher text. Since the key size is fixed at 56 bits, the number of keys available (the key space) is 2^{56} different keys (about 72,000,000,000,000,000 keys). This is a huge increase over the size of the key space in simple cryptographic systems.

A recent report by a group of scientists from AT&T Research*, Sun Microsystems*, the MIT Laboratory for Computer Science*, the San Diego Supercomputer Center*, Bell Northern Research* and others, entitled "Minimal Key Lengths for Symmetric Ciphers to Provide Adequate Commercial Security (Blaze, Diffie, Rivest, Schneier, Shimomura, Thompson and Wiener)" found that a pedestrian hacker with US \$400 to spend requires about 38 years of effort to decode data encrypted with DES with its large key space. Unfortunately, they also determined that a large organization with US \$300 million to spend could crack a 56-bit key space in about 12 seconds, using brute force techniques. They estimate that a 90-bit key protects data for about 20 years in the face of expected advances in computing power.

Related Information

Triple Pass DES (page 2-5)

3DES (page 2-7)

Outer Cipher Block Chaining (CBC) (page 2-8)

Triple Pass DES

Triple Pass DES is a cryptographic system that uses multiple passes of the DES algorithm to increase the effective key space available to the system. In triple pass DES, the clear text data is first encrypted with a 56-bit key. The resulting cipher text is then decrypted with a different key. Decrypting cipher text with the wrong key will result in unreadable data. Finally the unreadable data is encrypted again with the first key. This implementation of triple pass DES is known as EDE (for Encrypt, Decrypt, Encrypt) and the technique increases the effective key length from 56 bits to 112 bits. Note that 90-bit keys should protect encrypted data for about 20 years.

Go back to the simple Symmetric Cryptographic Systems (page 2-3) to illustrate the EDE technique. Assuming that the clear text is AT, the following steps are involved:

1. Encrypt with the key set to 3.

$$DW = \text{shift-right}(AT , K_1 = 3)$$

2. Decrypt the result DW with a different key (for example, 5).

$$YR = \text{shift-left}(DW , K_2 = 5)$$

Note that the result in this case is not the original clear text. Now encrypt the result YR with the key used in the first step.

$$BU = \text{shift-right}(YR , K_1 = 3)$$

The final cipher text is BU. When this cipher text is received, the decoding process must be performed in reverse (DED). The decoder must know the 2 keys ($K_1 = 3$ and $K_2 = 5$) and then make 3 passes:

1. Decrypt with the key set to 3.

$$YR = \text{shift-left}(BU , K_1 = 3)$$

2. Encrypt with the key set to 5.

$$DW = \text{shift-right}(YR , K_2 = 5)$$

3. Decrypt with the key set to 3.

$$AT = \text{shift-left}(DW , K_1 = 3)$$

The steps for both the triple pass DES technique and the 3DES technique are illustrated with the simple symmetric cryptographic system in the following table.

| Algorithm | Clear Text | After First Encrypt E | After First Decrypt D | After Second Encrypt E |
|---|-------------------|------------------------------|------------------------------|-------------------------------|
| Triple Pass DES (Key Space = $2 * 26 = 52$) | AT | $K_1 = 3$ DW | $K_2 = 5$ YR | $K_1 = 3$ BU |
| 3DES (Key Space = $3 * 26 = 78$) | AT | $K_1 = 3$ DW | $K_2 = 5$ YR | $K_3 = 4$ CV |

Related Information

3DES (page 2-7)

Data Encryption Standard (DES) (page 2-4)

Outer Cipher Block Chaining (CBC) (page 2-8)

3DES

3DES is a symmetric cryptographic system that uses multiple passes of the DES algorithm to increase the effective key space available to the system even further than triple pass DES. Use the same EDE technique as in Triple Pass DES (page 2-5), except that 3 different keys are used. Therefore, in pass 3 of Triple Pass DES, you would select a third key ($K_3 = 4$), which increases the effective key length from 56 bits for simple DES to 168 bits for 3DES.

The steps for both the triple pass DES technique and the 3DES technique are illustrated with the simple symmetric cryptographic system in the following table.

| Algorithm | Clear Text | After First Encrypt E | After First Decrypt D | After Second Encrypt E |
|--|------------|-----------------------|-----------------------|------------------------|
| Triple Pass DES (Key Space = $2 \times 26 = 52$) | AT | $K_1 = 3$ DW | $K_2 = 5$ YR | $K_1 = 3$ BU |
| 3DES (Key Space = $3 \times 26 = 78$) | AT | $K_1 = 3$ DW | $K_2 = 5$ YR | $K_3 = 4$ CV |

Related Information

Data Encryption Standard (DES) (page 2-4)

Outer Cipher Block Chaining (CBC) (page 2-8)

Outer Cipher Block Chaining (CBC)

Outer Cipher Block Chaining or outer-CBC is a technique used to further strengthen the DES, triple pass DES, and 3DES algorithms. This technique involves injecting random spoiler data into the encryption algorithm so that identical blocks of clear text does not result in the same cipher text even if the same key is used repeatedly. Therefore, if the clear text string "AT" is encrypted a thousand times with the same key, the resulting cipher text would be different each time. This is important since most file structures and application protocols use identical header information.

Related Information

Data Encryption Standard (DES) (page 2-4)

Triple Pass DES (page 2-5)

3DES (page 2-7)

Asymmetric Cryptographic Systems

Some algorithms do not use the same key to encrypt and decrypt. These algorithms are referred to as asymmetric, are usually complex, and often rely on the properties of very large prime numbers. A simple asymmetric algorithm, similar to the symmetric example, uses the same formula for encryption:

$$DW = \text{shift-right} (AT , 3)$$

In the symmetric example the encryption was "undone" using the mathematical operation of "shift-left." If you change the decryption operation to "shift-right," you need a different key to arrive back at the clear text:

$$AT = \text{shift-right} (DW , -3)$$

Note that the key used to decrypt the cipher text in this case is different from the key used to encrypt the clear text. The keys, however, are related. The relationship between the keys in the simple asymmetric algorithm can be expressed:

$$K_e = -1 * K_d$$

When asymmetric cryptography is used, the person doing the encrypting does not need to know the same key as the person doing the decrypting.

Asymmetric cryptography is often referred to as a public key cryptography. The public and private keys used in asymmetric cryptography are sometimes called key pairs, and are always related through some mathematical operation.

Related Information

[Symmetric Cryptographic Systems \(page 2-3\)](#)

[Symmetric Vs. Asymmetric Cryptography \(page 2-10\)](#)

[Key Space and Brute Force Attacks \(page 2-13\)](#)

Symmetric Vs. Asymmetric Cryptography

Symmetric and asymmetric cryptography have some significant differences. Symmetric cryptography tends to be fast compared to asymmetric cryptography. Therefore, symmetric algorithms are often used when large quantities

of data need to be exchanged and the 2 parties are known to each other. Conversely, asymmetric algorithms are used when small quantities of data need to be exchanged or the 2 parties are not known to each other.

Asymmetric cryptography is often used during authentication processes. Another significant difference between the 2 types of cryptographic systems is the length of the keys required by the algorithms. The keys used in symmetric algorithms are usually much smaller than those used in asymmetric algorithms, as described in the following table.

| | Symmetric | Asymmetric |
|-------------|------------------------------------|-------------------|
| Speed | Fast | Slow |
| Key size | Relatively small | Extremely large |
| Key usage | Shared secret | Public/private |
| Usual usage | Bulk data transfer | Authentication |
| Examples | DES, Triple Pass DES, 3DES, rc4 | RSA, PGP |

Related Information

- Asymmetric Cryptographic Systems (page 2-9)
- Symmetric Cryptographic Systems (page 2-3)
- Key Space and Brute Force Attacks (page 2-13)

Diffie-Hellman Session Key Exchange

The Diffie-Hellman key exchange protocol is based on an asymmetric algorithm. In asymmetric cryptographic systems, the key used to encrypt data is different from the key used to decrypt it. The key used to encrypt the data is usually referred to as a public key, while the key used to decrypt the data is called the private key, and the public key is derived from the private key. The length of the public and private keys can be 512 bits, 1024 bits, or 2048 bits.

The problem of key exchange between VPN Gateway components is solved using a protocol known as the Diffie-Hellman key exchange protocol. This protocol must be followed whenever two Intel NetStructure VPN components first begin to communicate, or when a session key expires. The strength of this protocol is that it allows the two components to negotiate or decide on a common session key without ever exchanging the key.

In general, when two devices exchange some data using an asymmetric cryptographic system, each device first requests the public key of the other device. They then use the public key of the other device to encrypt the data. When the other device receives the data, it can then use its private key to decrypt the data. As the name suggests, public keys are not secret and are made known to any device that requests them. Private keys, however, should never be revealed or distributed.

The Diffie-Hellman protocol specifies that the 2 components negotiating a common session key should each select half of a session key. They must also each derive some parameters that can be used to calculate the same half-session key. It is these parameters that are exchanged using the public/private key technique. Once the parameters are exchanged, then the second half of the session key can be calculated.

Notice that the session keys are never actually exchanged. The parameters for calculating half a session key are sent. To derive the full session key, both packets must be trapped and then broken. The effort required to break keys with lengths of 512, 1024, or 2048 bits makes this attack impractical.

The vulnerability of this type of key exchange protocol is the public key exchange.

Crypto Period

A crypto period defines how long a session key is actually used. Key lifetimes (crypto-periods) affect encryption strength because the longer the same session key is used the greater the chance that it is compromised. Additionally, the more data that is secured with a given key, the greater the loss if the key is compromised.

Long crypto-periods (key lives) also provide more ammunition for an adversary to break the key since the adversary potentially has access to significantly more data to work with. Finally, the longer a key is in use, the greater the temptation to break the keys since breaking the key provides the adversary with access to significantly more valuable data.

Related Information

Triple Pass DES (page 2-5)

3DES (page 2-7)

Packet Keys (page 3-8)

Key Space and Brute Force Attacks

Before reading this section, review Symmetric Cryptographic Systems (page 2-3) and Asymmetric Cryptographic Systems (page 2-9).

Key Space

In the simple cryptographic systems, up to 26 different possible keys can be selected. The keys available range from 1 to 26 since there are 26 letters in the alphabet. If 27 is used as your key, it would produce the same cipher text as if 1 was selected for your key. Therefore, your key space contains exactly 26 keys.

The longer the key length, the more possible combinations a potential code-breaker would have to test. The following table shows the number of possibilities for common key length (Source: FreeMarket.Net: Policy Spotlight, October-November 1997).

| Key Length | Possible Keys |
|------------|---|
| 40 bits | 1,099,511,627,776 |
| 56 bits | 72,057,594,037,927,900 |
| 90 bits | 1,237,940,039,285,380,000,000,000,000 |
| 128 bits | 340,282,366,920,938,000,000,000,000,000,000,000,000 |

Brute Force Attacks

A brute force attack captures some cipher text and then tries all 26 different possible keys. Given enough cipher text, a brute force attack could be quite effective. Obviously, if you can increase the number of different keys available, brute force attacks become correspondingly more difficult or time consuming. The trick is to find an algorithm that allows for an extremely large number of keys. The higher the key space, the more difficult the encryption is to break.

Related Information

Symmetric Cryptographic Systems (page 2-3)

Asymmetric Cryptographic Systems (page 2-9)

Symmetric Vs. Asymmetric Cryptography (page 2-10)

Encapsulation and Packet Handling

| | |
|------------------------------|-----|
| Encapsulation Overview | 3-1 |
| Secure Profiles | 3-2 |
| ESP Encapsulation | 3-4 |
| SST Encapsulation | 3-6 |
| Packet Handling | 3-7 |
| Packet Keys | 3-8 |

Encapsulation Overview

There are two types of encapsulation available with Intel NetStructure VPN products. The first is Shiva Smart Tunneling (SST) encapsulation. The second, called Encapsulating Security Payload (ESP) encapsulation, is an emerging standard as defined by IPSec. ESP (both 32- or 64-bit versions) should be used when you communicate with another non-Intel NetStructure VPN device (such as a firewall or router) that has implemented the ESP portion of the IPSec standard.

Encapsulation works in the following manner: when a packet is encrypted, a brand new packet is created. This new packet contains the entire original packet (including the header), which has been encrypted, a new header, and some information required by the device that finally decrypts the packet. The original packet is said to be encapsulated.

Related Information

Secure Profiles (page 3-2)

ESP Encapsulation (page 3-4)

SST Encapsulation (page 3-6)

Secure Profiles

Secure profiles are used to define how packets are encrypted when passing through a tunnel and how the establishment of the communication session is authenticated. Secure profiles must contain the following information to be complete.

Name

The name is a descriptive alphanumeric string used to reference the secure profile when it is applied to a tunnel. Although no naming convention is imposed, it is wise to define one prior to creating your profiles. Suggested naming conventions indicate either the intended use of the profile (for example, Interoffice or Dial-up user), the relative strength of the profile (for example, Strict or Very Strict), or the contents of the profile (for example, ESP-3DES-K1024-C12HRS for ESP encapsulation, 3DES, authentication key with 1024-bit public keys, and a crypto period of 12 hours).

Algorithm

The algorithm can be set to Data Encryption Standard (DES), Triple Pass DES, 3DES, or 40-bit DES for ESPv2 (IPSec) tunnels.

Keepalive

The keepalive interval can be set between 1 and 299 seconds or disabled (0). The keepalive feature is usually specified in profiles that are applied to remote links and has two main uses. The first is to ensure that the link status displayed on the remote Intel NetStructure VPN component accurately reflects the status of the tunnel. The second is to ensure that other Intel NetStructure VPN components can sense that a remote device has dropped its connection and therefore the tunnel must be renegotiated. Note that setting the keepalive to a small value causes many keepalive packets to be sent. This may impact the responsiveness of the remote connection.

Timeout

The keepalive timeout can be set between 2 and 300 seconds. This specifies how long a Intel NetStructure VPN component should wait for a packet from an opposing Intel NetStructureVPN component before declaring the session terminated and attempting to renegotiate the tunnel. If you specify a timeout on one end of a tunnel, you must specify a keepalive on the other end of the tunnel.

Encapsulation

The encapsulation can be set to either Shiva Smart Tunneling (SST) Encapsulation or to Encapsulating Security Payload (ESP) Encapsulation. ESP is the security portion of the IPsec standard. SST encapsulation is recommended for data exchange between Intel NetStructure VPN components, as it is stronger than ESP encapsulation.

ESP (either version) should be used when you communicate with another non-Intel NetStructure VPN device (such as a firewall or router) that has implemented the ESP portion of the IPsec standard.

The ESP implementation in all Intel NetStructure VPN Gateway products is tunnel mode. However, you can use transport mode by selecting ESP (either version), setting the ESP authentication to none, and selecting a value for the Authentication Header (AH). Transport mode encrypts only the payload.

Related Information

SST Encapsulation (page 3-6)

ESP Encapsulation (page 3-4)

Encapsulation Overview (page 3-1)

ESP Encapsulation

When the encapsulation is set to Encapsulating Security Payload (ESP), tunnel mode, the following information must be specified to fully define the security profile.

IV Length (Encapsulation)

The iv (initialization vector) length must be set to either 32 bits or 64 bits. This value is used during the outer cipher block chaining operation to ensure that the same packet encrypted multiple times will not generate the same cipher text. Both 32-bit and 64-bit iv's offer the same level of randomness, but 32-bit iv's use more system CPU and less bandwidth while 64-bit iv's use less CPU and more bandwidth. The actual difference in CPU usage and bandwidth usage is very small, and the industry tendency is to use a 64-bit iv length.

Authentication Header

This value can be set to keyed MD5, HMAC MD5, keyed SHA1, HMAC SHA1, or none. An authentication header (AH) is added to an ESP encapsulated packet (either version) to ensure that the packet is not altered during transmission, and is constructed by hashing the entire encrypted packet.

Setting the AH type specifies which algorithm to use for hashing. The SHA1 hashing algorithm is slightly more secure than MD5, but also slightly slower. MD5 adds 16 bytes of overhead to each packet, while SHA1 adds 20 bytes overhead. HMAC MD5 and SHA1 are slightly more secure than keyed MD5 and SHA1 respectively. Once again, the differences are marginal.

Ensure that the device on the other end (the firewall or router) conforms to the IPsec standards to ensure its interoperability with a Intel NetStructure VPN component.

AH Key Length

If you select either keyed MD5 or keyed SHA1 for your authentication header type, the value must be set between 0 and 55 bytes. If you select either HMAC MD5 or HMAC SHA1 for your authentication header (AH) type, the value must be set between 0 and 64 bytes. This value specifies the length of the key to be used when hashing the packet to produce the authentication header. The longer the key, the more secure the authentication, but the more time-consuming to manually enter.

Related Information

SST Encapsulation (page 3-6)

Packet Handling (page 3-7)

Packet Keys (page 3-8)

SST Encapsulation

When the encapsulation is set to Shiva Smart Tunneling (SST), the following information must be specified to fully define the security profile.

Authentication Method

The authentication method must be set to either certificates, challenge phrases, SecurID*, or RADIUS. Challenge phrases are often referred to as authentication keys. Sometimes challenge phrases are called passwords, but this is not a good synonym.

Public Key Length

The public key length must be set to 512 bits, 1024 bits, or 2048 bits. Note that public keys are used during the authentication and session key exchange processes. The longer the public key length, the more secure the session negotiation will be.

Crypto Period Length

The crypto period length defines how long a session key will be used. The default value for the crypto period is 1 month, although it can be set to as low as 3 hours. Given that a packet encrypted with a 90-bit key will require about 20 years of effort by a well-funded dedicated adversary to crack, it is often sufficient to use the default value for crypto period length.

Related Information

ESP Encapsulation (page 3-4)

Packet Handling (page 3-7)

Packet Keys (page 3-8)

Packet Handling

When a computer or network device communicates over a network (either a LAN or a WAN such as the Internet), the devices all perform similar functions. The application program (for example, a mail program) formulates a message, which is then passed to a set of functions collectively known as the TCP/IP stack. The TCP/IP stack looks at the message to determine if it needs to be sent out of the computer and then breaks the message into small packets and adds some header information to each packet.

The header information includes the following information:

- The destination address of the packet (the IP address of the mail server)
- The application port on the destination computer (for example, port 25 indicates that a SMTP mail server should be the application listening at the destination address)
- The source address of the sender (the IP address of the computer where the e-mail client is running)
- The application port that the sending machine used (usually randomly assigned)
- The protocol used (for SMTP mail, the protocol is TCP)

The maximum size of these packets including the header is usually 1500 bytes. Therefore, if your e-mail message is longer than 1500 bytes (1500 characters), it will be broken into several packets before being sent to the network layer and finally being transmitted onto the network. A simplified packet as released by the TCP/IP stack is shown next.

| | | | | | |
|------------|-----------|--------------|-------------|------|--------------|
| Dest IP | Src IP | Dest Port | Src Port | Prot | Payload Data |
|------------|-----------|--------------|-------------|------|--------------|

Figure: Simplified Packet

Related Information

Packet Keys (page 3-8)

Encapsulation Overview (page 3-1)

Packet Keys

The key (or keys in the case of triple pass DES or 3DES) used to encrypt a packet in SST encapsulation is called a packet key. A new packet key is randomly generated for every packet. This step, along with the outer-CBC technique, ensures that no matter how many identical original packets are sent, the new encrypted packets are significantly different each time. A simplified packet as released by a VPN Gateway is shown next.

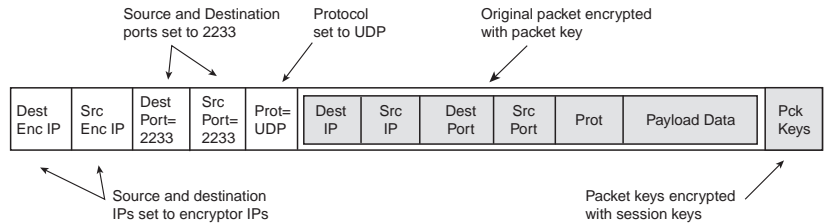


Figure: Encrypted Packet

This new packet has many interesting features. Note that the destination and source IP addresses of the original packet are different from the destination and source IP addresses of the new packet. The new IP addresses are the IP addresses of the VPN Gateway device that encrypted the packet. In many cases, these addresses are the IP addresses of the WAN interfaces of the VPN Gateway that secure the communication.

In a typical network configuration, a packet traveling from the Web server at the main office to a PC on the Branch office network has the IP addresses set to the WAN side IP addresses of the VPN Gateways at the gateways to these networks. The IP address of the Web server and the PC are hidden from anyone intercepting the packet and the interceptor gains no knowledge about the LANs.

Note also that the destination and source ports are both set to 2233. This application port number indicates only that the packet is encrypted. The source port in the original packet would be set to port 80 to indicate that this is World Wide Web traffic. Therefore, the nature of the packet is hidden from anyone intercepting the packet.

The protocol has been modified and set to UDP. The original packet, if it was an http (www) packet, has its protocol set to TCP and indicates to an intruder that an acknowledgment packet is expected. Trapping acknowledgment packets is a good way to gain some

knowledge of the contents of an encrypted packet, which can be used to help break the encryption. Setting all the encrypted packet protocols to UDP removes this bit of knowledge and further secures the communication.

The entire original packet is encrypted. Some other solutions only encrypt the payload data and expose a wealth of information about the nature of the packet and the source and destination networks.

Finally, the packet keys are encrypted with session keys and appended to the new packet. Remember that DES, triple pass DES, and 3DES are symmetric algorithms. Therefore, both the device encrypting the packet and the device decrypting the packet must know the same keys. The packet keys, however, are randomly generated for each packet. Assuming that both the encryptor and the decryptor know the same session keys, this technique makes the encryption more secure in 2 ways. Attempts to break the packet keys are not practical since it changes with every packet. The most that can be gained is about 1400 bytes of data from an operation that will take years. The session keys are used to encrypt a very small amount of data (only the packet keys), which is random. If the session keys are changed periodically, then even this small target is moving and attacks are made more difficult. The frequency with which session keys are changed is called the crypto period.

Related Information

Packet Handling (page 3-7)

Authentication Methods

| | |
|---|-----|
| Authentication Methods Overview | 4-1 |
| Certificate Authentication | 4-2 |
| Challenge Phrase Authentication | 4-3 |
| SecurID Authentication | 4-4 |
| RADIUS Authentication | 4-5 |
| Entrust Authentication | 4-6 |

Authentication Methods Overview

An authentication method defines how a Intel NetStructure VPN component validates the identity of another component. The identity of a component includes its name, its IP address, and its public key. When the packet encapsulation type is set to Shiva Smart Tunneling (SST), there are five possible authentication methods:

- Certificates by means of the Intel NetStructure Certificate Authority
- Challenge Phrase
- SecurID
- RADIUS
- Entrust* by means of the Entrust Certificate Authority

Related Information

Certificate Authentication (page 4-2)

Challenge Phrase Authentication (page 4-3)

SecurID Authentication (page 4-4)

RADIUS Authentication (page 4-5)

Entrust Authentication (page 4-6)

Certificate Authentication

The first thing that two Intel NetStructure VPN components do when they enter into a communication is to exchange their certificates. Next, they verify the authenticity of the certificates by ensuring that:

- The identifying information and the digital signature are separated.
- A new MD5 digest of the identifying information is generated.
- The digital signature is decrypted.

The result is the MD5 digest (or summary) of the identifying information that was generated by the Intel NetStructure Certificate Authority when the certificate was created.

The new MD5 digest and the digest extracted from the digital signature are then compared. If they are exactly the same, the device is sure that the certificate is valid.

Note that the Intel NetStructure Certificate Authority is not involved in the authentication process. Once the authentication process is complete on both sides, the 2 devices can then begin the session key exchange process or negotiation.

Related Information

SecurID Authentication (page 4-4)

RADIUS Authentication (page 4-5)

Challenge Phrase Authentication (page 4-3)

Entrust Authentication (page 4-6)

Challenge Phrase Authentication

Authentication using challenge phrases is very similar to authentication using certificates. The difference is that a certificate authority is not present to create and certify a certificate. Therefore, the Intel NetStructure VPN components must create a certificate for themselves. This type of certificate is essentially the same as a certificate generated by the Intel NetStructure Certificate Authority except that the digital signature is encrypted with a challenge phrase rather than with the private key of the certificate authority. The implication is that when two devices attempt to authenticate each other for the first time, they must both know the challenge phrase of the other device. Therefore, the challenge phrase for a particular device must be input on the device and must also be input on any other device with which it needs to communicate.

Related Information

SecurID Authentication (page 4-4)

RADIUS Authentication (page 4-5)

Entrust Authentication (page 4-6)

SecurID Authentication

SecurID is an authentication method licensed from Security Dynamics that the Intel NetStructure VPN Suite supports. SecurID is used only between a Intel NetStructure VPN Client and a VPN Gateway. As with certificates, SecurID enlists a trusted third party to positively identify a device. Here, the third party is an ACE/Server.

Unlike the Intel NetStructure Certificate Authority Server, however, the ACE/Server must be available whenever a secure tunnel is being established. Whenever a remote user attempts to establish a secure tunnel with a VPN Gateway, the user must provide a user name and a time-dependent pass code that the VPN Gateway then verifies with the ACE/Server before allowing the tunnel to be established. Typically, the pass code is composed of two parts: a PIN number and a SecurID access code.

For further information on using SecurID, consult Security Dynamics' SecurID documentation.

Related Information

[RADIUS Authentication \(page 4-5\)](#)

[Challenge Phrase Authentication \(page 4-3\)](#)

[Entrust Authentication \(page 4-6\)](#)

RADIUS Authentication

The RADIUS authentication method is very similar to the SecurID authentication method in that it uses a trusted third party to authenticate tunnels between Intel NetStructure VPN Clients and VPN Gateway devices. The trusted third party is a RADIUS Authentication Server. When an Intel NetStructure VPN Client attempts to establish a tunnel with a VPN Gateway, the VPN Gateway asks the Intel NetStructure VPN Client to provide its RADIUS user name and password. The VPN Gateway then uses its own secret key to contact the RADIUS Authentication Server to verify the Intel NetStructure VPN Client's identity.

There is a second type of RADIUS server supported by the Intel NetStructure VPN Suite: a RADIUS Accounting Server. This server keeps track of those remote users who have established connections to VPN Gateway devices, and the amount of time each connection is maintained. It is not necessary to have a RADIUS Accounting Server to use the RADIUS method of authentication.

Related Information

Challenge Phrase Authentication (page 4-3)

SecurID Authentication (page 4-4)

Entrust Authentication (page 4-6)

Entrust Authentication

Entrust authentication is an authentication method licensed from Entrust Technologies that the Intel NetStructure VPN suite supports. Entrust authentication is supported for tunnels made between two LanRover VPN Gateway devices (including IPSec tunnels) and between an Intel NetStructure VPN Client and a VPN Gateway using the Shiva Smart Tunneling (SST) protocol. Entrust enlists a trusted third party to positively identify a device using X.509 certificates and performs key and certificate functions.

The Entrust Server maintains a list of all of the public keys that have been created and also issues, revokes, recovers certificates, and maintains a revocation list. The VPN Gateway acts as an Entrust client using Entrust services, has its own certificate issued by the Certificate Authority, and updates its own revocation by means of the Certificate Authority.

Related Information

[SecurID Authentication \(page 4-4\)](#)

[RADIUS Authentication \(page 4-5\)](#)

Firewalls and Tunnels

| | |
|---|------|
| Firewall and Tunnels Overview | 5-1 |
| Firewall Functions | 5-2 |
| Filters | 5-6 |
| Tunnel Types | 5-8 |
| Site-to-Site Tunnels | 5-9 |
| Single-User Tunnels | 5-12 |
| Multiuser Tunnels | 5-16 |
| Tunnel Modes | 5-20 |
| One-Way In Firewall Rules | 5-22 |
| One-Way Out Firewall Rules | 5-24 |
| Outbound Proxy | 5-26 |
| Inbound Proxy | 5-28 |
| Tunnel Termination and Firewall Rules | 5-31 |

Firewall and Tunnels Overview

Firewalls and tunnels are the core parts of a network that control the flow of data packets in and out of a trusted and untrusted network.

Firewalls

Firewalls control access between a red (trusted) network and a black (untrusted) network. The black (untrusted) network is often the Internet. A VPN Gateway can act like a firewall that can be configured to contain rules. Firewall rules determine which packets can pass through the gateway between the trusted and untrusted network.

Using a firewall around a network helps to protect that network from unwanted data packets entering or leaving the network, but it has some fundamental flaws. First, the data packets can be captured as they move through the firewall connecting the networks. Data could be extracted from the packets or a new packet could take the place of the original packet. All a hacker needs to do is replace the original packet with a new packet to gain access to the destination network.

Tunnels

The term tunnel, when used in the context of a network and firewall solution, can be explained by the following:

A tunnel acts as a means of transport for data packets. In most cases, a tunnel encrypts the data packets, making them unusable should they be intercepted by an unintended user and hackers. A tunnel also transports the packets to their destination and decrypts them, providing an overall secure means of transportation.

Related Information

[Firewall Functions \(page 5-2\)](#)

[Tunnel Types \(page 5-8\)](#)

[Tunnel Modes \(page 5-20\)](#)

[Tunnel Termination and Firewall Rules \(page 5-31\)](#)

Firewall Functions

Each VPN Gateway has at least two physical interfaces (that is, two Ethernet cards). Each interface is assigned a color, either red or black. If both interfaces have the same color, the VPN Gateway will not perform any firewall functions between the interfaces. In this case, the VPN Gateway becomes a router (or bridge) and an encryptor.

When two interfaces on a VPN Gateway have different colors, packets arriving at one interface must pass through the firewall to move to the other interface.

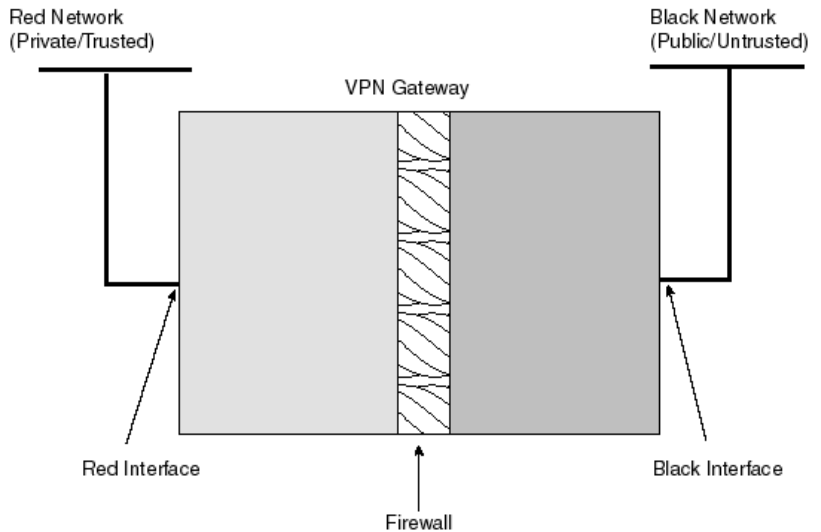


Figure: VPN Gateway as a Firewall

Stateless

The VPN Gateway is instructed to allow or disallow all packets traveling between the red (trusted) and black (untrusted) network. The VPN Gateway checks each packet as it arrives to ensure it is valid. If the packet matches the filter rule (shown in the following table), it passes from one interface to the other. The VPN Gateway then immediately looks for the next incoming packet. This is called stateless filtering, since the VPN Gateway does not remember that a packet passed through a filter rule. If a packet is considered invalid, it is simply not allowed entry to the red (trusted) network.

| Parameter Description | Parameter Value | Comments |
|------------------------------|------------------------|---|
| From IP address | 10.1.1.193 | User chris is assigned Client IP 10.1.1.193. |
| From subnet mask | 255.255.255.224 | A maximum of 30 users with addresses starting from 10.1.1.193 are allowed through the firewall. |
| From application port | ALL | The application port used to make the HTTP (www) request is usually unknown. |
| To IP address | 10.1.1.2 | The Web Server's IP address. |
| To subnet mask | 255.255.255.255 | Access Web Server only. |
| To application port | 80 | Web servers usually listen on this port. |
| Action | Stateful | |
| Direction | Inbound | The group comes from the black (untrusted) and crosses to the red (trusted). |
| NAT | No | |
| Protocol | TCP | HTTP is transported by means of TCP, not UDP. |

Stateful

All other firewall rules are stateful, which means that a communication session is established between a device inside the firewall (on the red network) and a device outside the firewall (on the black network). In this way when a device on a red (trusted) network (in the case of a one-way outbound link or outbound proxy) makes a request to a device on a black (untrusted) network that requires a response, the response is allowed back into the network.

The VPN Gateway is also configured to allow data packets from the black (untrusted) network to establish a link with a specific IP address inside the red (trusted) network. In this case, the VPN Gateway stores the IP address of the computer sending the data packets from the black (untrusted) network, so if the link is dropped and tries to reestablish, the VPN Gateway remembers the IP address of the computer that created the initial link. This type of stateful connection is known as an inbound proxy or one-way in firewall rule. The only difference between an inbound proxy and a one-way in firewall rule is the point at which a data packet is removed from encapsulation and the firewall rules are applied.

The inbound data packets from the originating device look directly for the IP address of the VPN Gateway, not the real address of the destination computer. When the data packet arrives at the gateway, the gateway checks the validity of the packet, maintains the state of the transmission, and the packets are permitted or denied based on the stateful rules configured in the VPN Gateway. Only if the packet is permitted by the firewall rule is it then routed to the destination computer according to the IP addressing information it carries.

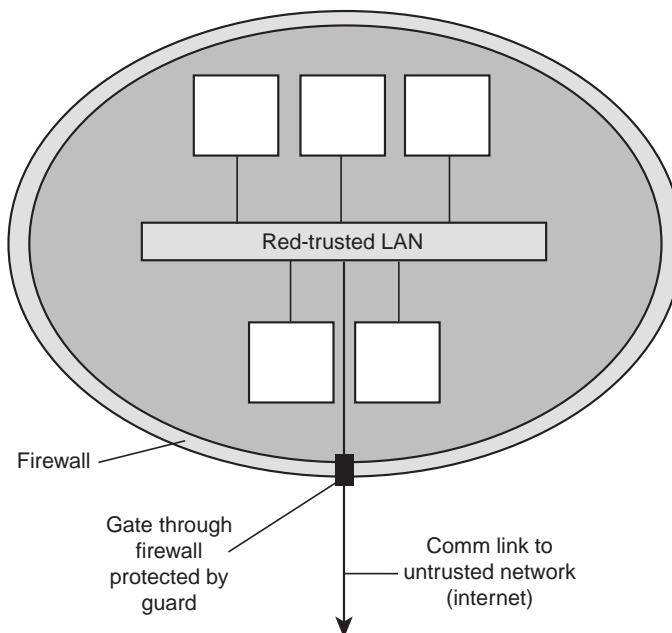


Figure: A Firewalled LAN

Related Information

One-Way Out Firewall Rules (page 5-24)

One-Way In Firewall Rules (page 5-22)

Tunnel Types (page 5-8)

Filters

Filters are used to allow or block (permit or deny) the flow of packets through the VPN Gateway. The source device initiating the session can be either on the red (trusted) or the black (untrusted) subnet. Think of a filter as a hole through the firewall through which specified devices can communicate. Packets passing through a filter are not modified in any way and no state information is maintained.

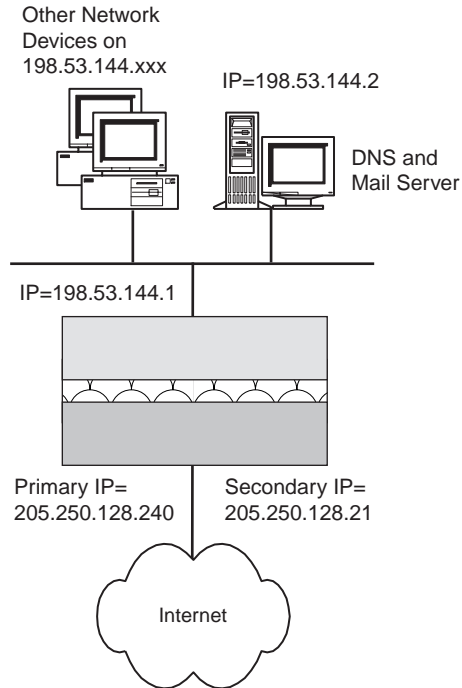


Figure: Example of a Filter

If you want a public domain name server (DNS) to execute on a machine on a red network, define a filter as described in the following table.

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|--|
| To IP address | 0.0.0.0 | Do not limit which addresses can access or be accessed by the DNS. |
| To subnet | 0.0.0.0 | |
| To port | 53 | DNS updates are requested on this port. |
| From IP address | 198.53.144.2 | You allow only the DNS machine to be addressed on the red (trusted) network. |
| From subnet | 255.255.255.255 | |
| From port | 53 | Make DNS requests on this port. |
| Protocol | TCP | Make DNS requests and refreshes over TCP, not UDP. |
| Action | permit | You allow access. |

Related Information

[Firewall and Tunnels Overview \(page 5-1\)](#)

[Tunnel Types \(page 5-8\)](#)

[Tunnel Termination and Firewall Rules \(page 5-31\)](#)

Tunnel Types

There are three types of tunnels:

- Site-to-Site
- Single-User
- Multiuser

If two networks want to communicate and not be subject to the packets being hijacked while en route, tunnels can be established between the networks. This assumes, of course, that two networks want to communicate safely and are both protected by firewalls. The tunnels can be started either inside or outside of a firewall. When a tunnel is started inside a firewall, then the packets entering or leaving the tunnel do not need to pass through the gateway and are not subject to the firewall rules that the gateway is configured to follow. If a tunnel is started outside the firewall, then packets entering or leaving the tunnel must pass through the gateway. They are then subjected to the firewall rules before passing through the gateway.

The Intel NetStructure VPN products implement tunnels using authentication methods and encryption techniques. Since the traffic passing between two Intel NetStructure VPN components is encrypted, it is as if the data is traveling in a tunnel.

Related Information

[Site-to-Site Tunnels \(page 5-9\)](#)

[Single-User Tunnels \(page 5-12\)](#)

[Multiuser Tunnels \(page 5-16\)](#)

Site-to-Site Tunnels

A site-to-site tunnel is defined between two devices with fixed IP addresses. A fixed IP address implies that the device is always present and the Intel NetStructure VPN component on the other end of the tunnel can initiate communication with the fixed device. This behavior can be overridden on one end of the tunnel, if desired. A site-to-site tunnel is usually defined when the tunnel is between two networks and both ends of the tunnel are available through VPN Gateway devices.

A site-to-site tunnel is fully defined with the following components:

- IP address of the opposing VPN Gateway
- Secure profile to be applied to the communication
- Color (mode) of the tunnel
- IP route pushing packets into the tunnel

The IP address of the opposing VPN Gateway highlights the fact that a tunnel cannot exist without a Intel NetStructure VPN component on the other end. A secure profile defines how the establishment of the tunnel should be authenticated and how the communication should be secured. The mode of the tunnel specifies where the tunnel terminates. Finally, the IP route specifies which packets should enter the tunnel.

The following example illustrates a secure tunnel, which secures all communication between two networks.

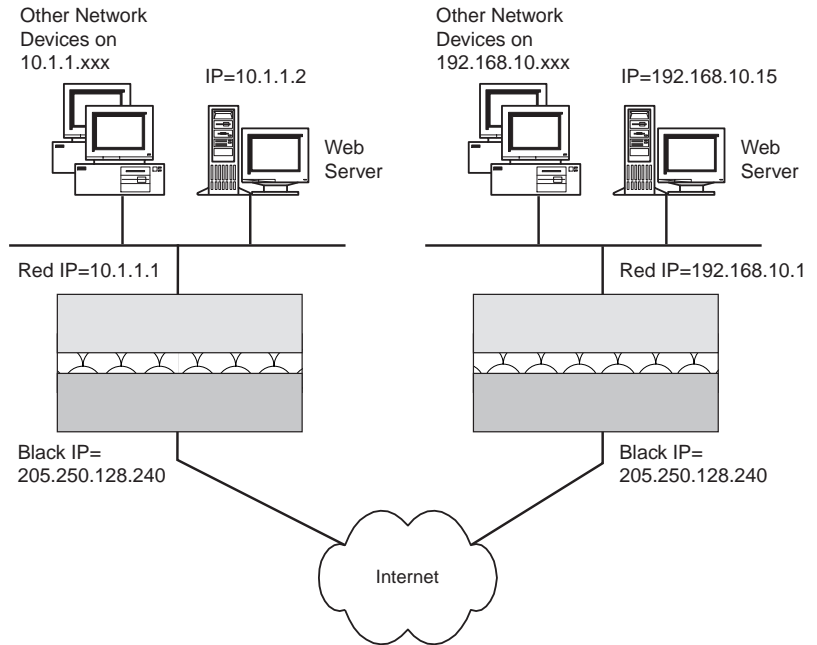


Figure: A Secure Tunnel

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|---|--|
| Opposing gateway | 198.53.144.120 | 205.250.128.240 |
| Secure profile (must be previously defined) | Very strict | Very strict |
| Tunnel mode | Red | Red |
| IP route | IP route 192.168.10.0 255.255.255.0 198.53.144.120 | IP route 10.1.1.0 255.255.255.0 205.250.128.240 |

Note that the tunnel has to be defined on both VPN Gateway devices. Therefore, when you specify the opposing VPN Gateway on device A, point at device B. Also, define the same secure profile on both VPN Gateway devices. The tunnel mode, however, can be different on each VPN Gateway. Finally, the route statements tell the VPN Gateway devices which packets should enter the tunnel.

Related Information

Single-User Tunnels (page 5-12)

Multiuser Tunnels (page 5-16)

Tunnel Types (page 5-8)

Single-User Tunnels

A single-user tunnel is defined between a fixed device and one with no fixed IP address, which implies that the device on the other end of the tunnel is not always present or may change its address. A single-user tunnel is usually defined on a VPN Gateway when the other end of the tunnel is an Intel NetStructure VPN Client.

You can assign a known IP address to the remote device using network address translation (NAT). This address is known as the Client IP. When a tunnel has been established with the remote device, all packets coming from the remote device will have their actual source address replaced with the Client IP address.

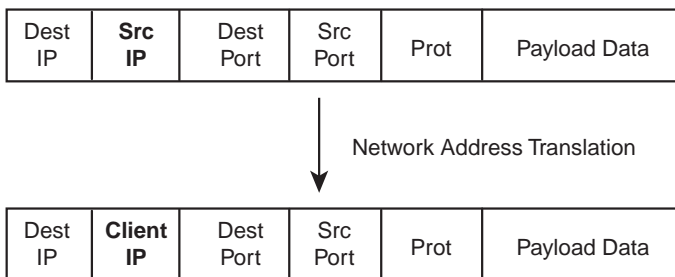


Figure: Network Address Translation

A single-user tunnel is fully defined with the following components:

- User name of the opposing Intel NetStructure VPN Client
- Secure profile to be applied to the communication
- Color (mode) of the tunnel
- Client IP if NAT is being used

Identify the opposing Intel NetStructure VPN device by a user name instead of an IP address. The secure profile defines how the establishment of the tunnel is authenticated and how the communication is secured. The mode of the tunnel specifies where the tunnel terminates. The IP route is no longer required.

Full Access

The following table describes a tunnel that allows a remote user (called chris) full access to the red (trusted) network available through VPN Gateway A, while not allowing access to the network available through VPN Gateway B.

| Tunnel Definition Parameters | VPN Gateway A | Intel NetStructure VPN Client |
|---|------------------------|--|
| Remote user name | chris | (the VPN's name) |
| Secure profile (must be previously defined) | dial-up | Accept peer proposal or same parameters as dial-up profile |
| Tunnel mode | Red | Not applicable |
| IP route | Not required | Not applicable |
| Client IP | 0.0.0.0 (not required) | Not applicable |

In the previous table, user chris is given complete access to the trusted network.

Limited Access

The following figure shows how to use a combination of a tunnel and a firewall rule to give a remote user limited access to the trusted network.

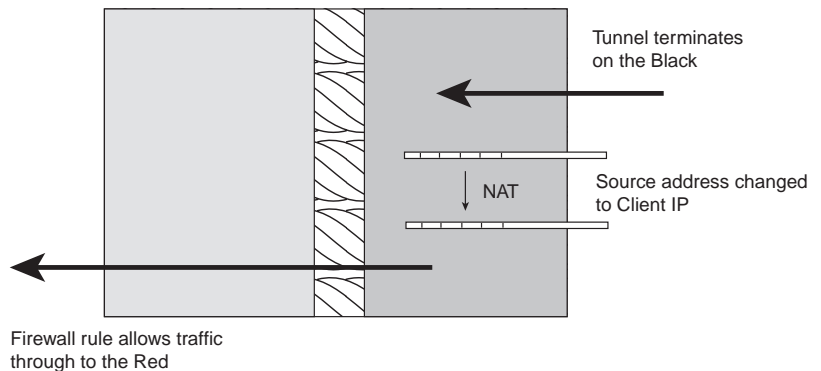


Figure: Source Address Change

For example, to allow a remote user (called leslie) access to the Web server available through VPN Gateway A while not allowing access to the rest of that network or to the network available through VPN Gateway B, a tunnel is defined for the user to the black (untrusted) side of the VPN Gateway and a firewall rule is created to allow the traffic from the black (untrusted) network to the red (trusted) network. In this case a Client IP is used to assign the remote user a known IP address on the red (trusted) network. This address is needed in order to identify the remote user in the firewall rule.

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|---------------|----------------|
| Remote user name | leslie | No access |
| Secure profile (must be previously defined) | dialup | Not applicable |
| Tunnel mode | Black | Not applicable |
| IP route | Not required | Not applicable |
| Client IP | 10.1.1.193 | Not applicable |

Firewall Rule

The following table describes the firewall rule.

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|--|
| From IP address | 10.1.1.193 | User leslie is being assigned Client IP 10.1.1.193. |
| From subnet mask | 255.255.255.255 | |
| From application port | ALL | The application port used to make the HTTP (www) request is usually unknown. |

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|--|
| To IP address | 10.1.1.2 | The Web Server's IP address. |
| To subnet mask | 255.255.255.255 | Access Web Server only. |
| To application port | 80 | Web servers usually listen on this port. |
| Action | Stateful | |
| Direction | Inbound | The group comes from the black (untrusted) network and crosses to the red (trusted) network. |
| NAT | No | |
| Protocol | TCP | HTTP is transported by means of TCP, not UDP. |

Related Information

Site-to-Site Tunnels (page 5-9)

Multiuser Tunnels (page 5-16)

Tunnel Types (page 5-8)

Multiuser Tunnels

A multiuser tunnel is defined between a fixed device and a group of remote users, which implies that the devices on the other end of the tunnel are not always present or may change their addresses. A multiuser tunnel is usually defined on a VPN Gateway for the ease of administration, simplification of the overall configuration, and to limit the number of Intel NetStructure VPN Client users that can access the network through the VPN Gateway at any given time.

Any member of the remote user group that attempts to connect through the tunnel when the preset number of other users are already connected is refused. This feature is useful for large organizations in that it allows them to prioritize access through the VPN Gateway by groups, thereby avoiding situations where important tunnel requests are refused because all 1024 available sessions are in use.

Any remote device that connects successfully is given one of a preset group of IP addresses with which it appears on the network, accessible through the Gateway. Hence, all connections using multiuser tunnels use network address translation (NAT). A multiuser tunnel is fully defined with the following components:

- Group name
- Number of users that can establish tunnels at any given time and associated NAT IP addresses (known as Client IP)
- Secure profile to be applied to the communication
- Color (mode) of the tunnel

The group of opposing Intel NetStructure VPN devices is now identified by a group name. The secure profile defines how the establishment of the tunnel should be authenticated and how the communication should be secured. The mode of the tunnel specifies where the tunnel terminates. The IP route is no longer required.

Note: If the authentication method specified in the secure profile associated with a multiuser tunnel is a challenge phrase, the same challenge phrase must be given out to each member of the group. This is not recommended.

Full Access

The following table shows a tunnel that would allow a group (called audit) full access to the red (trusted) network available through VPN Gateway A, while not allowing access to the network available through VPN Gateway B. Note that a maximum of 30 members of the group will be allowed to use the tunnel at once.

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|----------------------|----------------------|
| Group name | audit | No access |
| Client IP | 10.1.1.193 | Not applicable |
| Number of clients | 30 | |
| Secure profile (must be previously defined) | dial-up | Not applicable |
| Tunnel mode | Red | Not applicable |
| IP route | Not required | Not applicable |

In the previous table, group audit is given complete access to the trusted network.

Limited Access

The next table shows how to use a combination of a tunnel and a firewall rule to give a group limited access to the red (trusted) network. For example, to allow a group called sales access to the Web server available through VPN Gateway A while not allowing access to the rest of that network or to the network available through VPN Gateway B, a tunnel is defined for the group to the black side of the VPN Gateway and a firewall rule is created to allow the traffic from the black (untrusted) network to the red (trusted) network.

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|----------------------|----------------------|
| Group name | sales | No access |
| Client IP | 10.1.1.193 | Not applicable |
| Number of clients | 30 | Not applicable |
| Secure profile (must be previously defined) | dial-up | Not applicable |
| Tunnel mode | Black | Not applicable |
| IP route | Not required | Not applicable |

Firewall Rule

The firewall rule is explained in the following table.

| Parameter Description | Parameter Value | Comments |
|------------------------------|------------------------|---|
| From IP address | 10.1.1.192 | |
| From subnet mask | 255.255.255.224 | A maximum of 30 users with addresses starting from 10.1.1.193 are allowed through the firewall. |
| From application port | ALL | The application port used to make the HTTP (www) request is usually unknown. |
| To IP address | 10.1.1.2 | The Web Server's IP address. |
| To subnet mask | 255.255.255.255 | Access Web Server only. |

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|--|
| To application port | 80 | Web servers usually listen on this port. |
| Action | Stateful | |
| Direction | Inbound | The group comes from the black (untrusted) network and crosses to the red (trusted) network. |
| NAT | No | |
| Protocol | TCP | HTTP is transported by means of TCP, not UDP. |

Related Information

Site-to-Site Tunnels (page 5-9)

Single-User Tunnels (page 5-12)

Tunnel Types (page 5-8)

Tunnel Modes

Intel NetStructure VPN tunnels are assigned a mode of either red or black. The color of the tunnel indicates whether the device on the other end of the tunnel is trusted; red is trusted and black is untrusted.

When a tunnel starts inside a trusted network, it indicates that the packets entering or leaving the tunnel are trusted. This is known as a red tunnel. Conversely, when a tunnel starts outside the trusted network, it indicates that the data packets are not trusted. This is known as a black tunnel. In both cases, the data packets can travel between the two networks safely.

There are three possible ways exist to build a tunnel, depending on where the two ends terminate:

- If both ends of the tunnel terminate inside the trusted network, then the tunnel is called a red-red network. In this case, the two networks trust each other.
- If both ends of the tunnel terminate outside the network, the tunnel is called a black-black network, and neither network trusts the other completely.
- Finally, if one end of a tunnel terminates inside a network, while the other end terminates outside the network, then the tunnel is called a red-black network or a black-red network. In this case, one network trusts the other while the trust is not reciprocated.

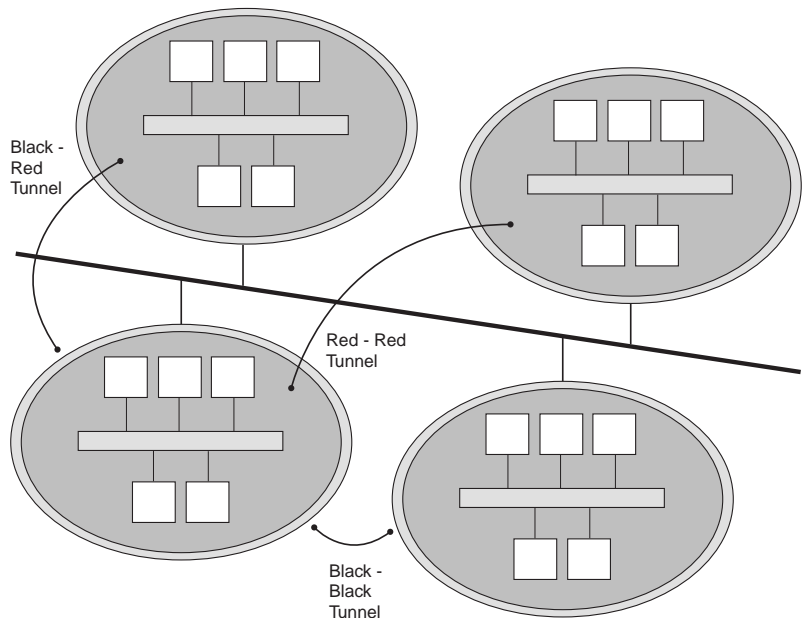


Figure: Firewallled LANs With Encrypted Tunnels

Related Information

Tunnel Types (page 5-8)

Tunnel Termination and Firewall Rules (page 5-31)

One-Way In Firewall Rules

One-way in firewall rules allow devices on a black (untrusted) network to establish communication sessions with devices on the red (trusted) network. No network address translation (NAT) is performed when a session is established through a stateful one-way in firewall rule. One-way in firewall rules can grant access to services executing on devices on a red (trusted) subnet having routed IP addresses.

If you want to allow SMTP mail from people on the Internet to be sent into the mail server, define a one-way in rule as described in the following table.

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|---|
| From IP address | 0.0.0.0 | The mail can come from any IP address. |
| From subnet mask | 0.0.0.0 | |
| From application port | ALL | The application port used to send the mail is usually unknown. |
| To IP address | 198.53.144.2 | Assumes that the mail record associated with your domain name points to this address. |
| To subnet mask | 255.255.255.255 | The mail must arrive at this IP address only. |
| To application port | 25 | The SMTP mail server listens on this port. |
| Protocol | TCP | SMTP is transported by means of TCP, not UDP. |

Related Information

Inbound Proxy (page 5-28)

Outbound Proxy (page 5-26)

One-Way Out Firewall Rules (page 5-24)

One-Way Out Firewall Rules

One-way out firewall rules allow devices on a red (trusted) network to establish communication sessions with devices on a black (untrusted) network. One-way out firewall rules allow users on routed red (trusted) subnets to have access to services on a black (untrusted) subnet.

No network address translation (NAT) is performed when a session is established through a one-way out firewall rule. Therefore, the source address of the packets leaving the red (trusted) network must be routable on the black (untrusted) network. Routable means that the devices on the black (untrusted) network know how to send packets to the source address.

If you want to allow people on the red (trusted) network to browse the World Wide Web on the Internet, define a oneway out firewall rule as described in the following table.

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|--|
| From IP address | 198.53.144.0 | This address allows anyone on the red (trusted) network whose IP address starts with 198.53.144. |
| From subnet mask | 255.255.255.0 | |
| From application port | ALL | The application port used to make the HTTP (www) request is usually unknown. |
| To IP address | 0.0.0.0 | This address allows you to go to any Web site on the Internet. |
| To subnet mask | 0.0.0.0 | |

| Parameter Description | Parameter Value | Comments |
|------------------------------|------------------------|---|
| To application port | 80 | Web servers usually listen on this port. |
| Protocol | TCP | HTTP is transported by means of TCP, not UDP. |

Related Information

Inbound Proxy (page 5-28)

Outbound Proxy (page 5-26)

One-Way In Firewall Rules (page 5-22)

Outbound Proxy

Outbound proxies allow devices on a red subnet to establish communication with devices on black subnets. The outbound proxy function performs a network address translation (NAT) on any packets passing through the proxy. Outbound proxies are, therefore, often used to allow users on unrouted red subnets to have access to services on a black subnet.

If you want to allow people on the red network to browse the World Wide Web on the Internet, define an outbound proxy as described in the following table.

| Parameter Description | Parameter Value | Comments |
|-----------------------|-----------------|---|
| Outbound proxy IP | 205.250.128.240 | The address the packets take on as they exit the red (trusted) network. |
| From IP address | 10.1.1.0 | This address allows anyone on the red (trusted) network whose IP address starts with 10.1.1 to go out to the black (untrusted) network. |
| From subnet mask | 255.255.255.0 | |
| From application port | ALL | The application port used to make the HTTP (www) request is usually unknown. |
| To IP address | 0.0.0.0 | This address allows you to go to any Web site on the Internet. |
| To subnet mask | 0.0.0.0 | |

| Parameter Description | Parameter Value | Comments |
|------------------------------|------------------------|---|
| To application port | 80 | Web servers usually listen on this port. |
| Protocol | TCP | HTTP is transported by means of TCP, not UDP. |

Related Information

Inbound Proxy (page 5-28)

One-Way Out Firewall Rules (page 5-24)

One-Way In Firewall Rules (page 5-22)

Inbound Proxy

Inbound proxies allow devices on a black (untrusted) subnet to establish communication sessions with a device on a red (trusted) subnet. Inbound proxies can grant access to services executing on devices on a red (trusted) subnet having unrouted or private IP addresses. When you define an inbound proxy, the devices on the black (untrusted) network must address their packets to the black (untrusted) interface of the VPN Gateway. The VPN Gateway then looks at where the packet originated, what the destination address is, what the destination port is, and decides to which address on the red (trusted) network to send the packet.

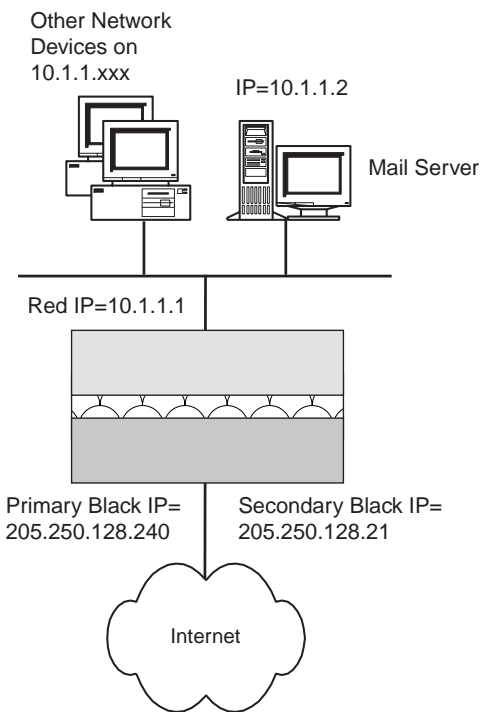


Figure: Inbound and Outbound Proxies

If you want to allow SMTP mail from people on the Internet to be sent into a mail server, define an inbound proxy as described in the following table.

| Parameter Description | Parameter Value | Comments |
|------------------------------|------------------------|---|
| Inbound proxy IP | 10.1.1.2 | This is where the packets should end up. |
| From IP address | 0.0.0.0 | The mail could come from any IP address. |
| From subnet mask | 0.0.0.0 | |
| From application port | ALL | The application port used to send the mail is usually unknown. |
| To IP address | 205.250.128.21 | Assumes that the mail record associated with your domain name points to this address. |
| To subnet mask | 255.255.255.255 | The mail must arrive at this IP address only. |
| To application port | 25 | The SMTP mail server listens on this port. |
| Protocol | TCP | SMTP is transported by means of TCP, not UDP. |

Related Information

Outbound Proxy (page 5-26)

One-Way Out Firewall Rules (page 5-24)

One-Way In Firewall Rules (page 5-22)

Tunnel Termination and Firewall Rules

When a tunnel terminates outside a firewall, a packet must be compared to the firewall rules, which determine whether or not to let the packet through the gateway. In this way, tunnels and firewall rules can be used together to specify what traffic passes through the VPN Gateway. Four basic permutations of tunnel termination and traffic destinations exist:

- The tunnel terminates on the red (trusted) network or interface and the traffic is destined for the red (trusted) network or interface.
- The tunnel terminates on the black (untrusted) network or interface, but the traffic is destined for the red (trusted) network or interface.
- The tunnel terminates on the red (trusted) network or interface, but the traffic is destined for the black (untrusted) network or interface.
- The tunnel terminates on the black (untrusted) network or interface and the traffic is destined for the black (untrusted) network or interface.

Note: The terms network and interface are used interchangeably.

Tunnel Terminates in the Red (Trusted) Network

The case where a tunnel terminates in the red (trusted) network and the traffic is destined for the red (trusted) network is the typical case of giving a remote device complete access to the trusted side of the VPN Gateway. Because the tunnel bypasses the firewall, the destination addresses of the traffic are examined only for the purpose of routing the packets to their destination.

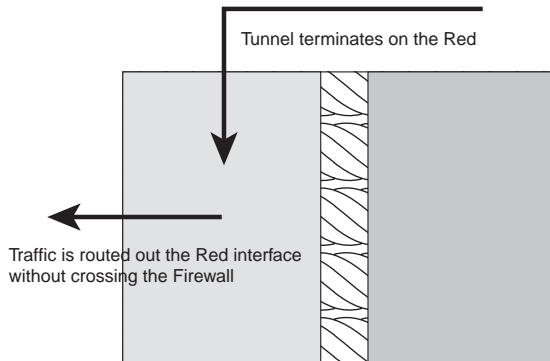


Figure: Tunnel Terminates in the Red (Trusted) Network

Tunnel Terminates in the Black (Untrusted) Network

A tunnel that terminates in the black (untrusted) network but where the traffic is destined for the red (trusted) network gets the traffic to the VPN Gateway safely and then blocks it at the firewall. A firewall rule must be in place to allow the traffic through.

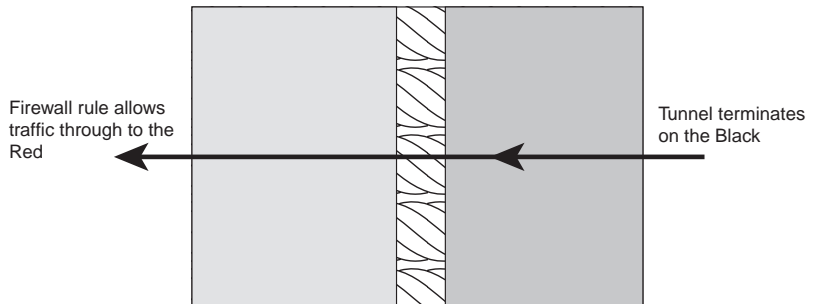


Figure: Tunnel Terminates in the Black (Untrusted) Network

Tunnel Terminates in the Red (Trusted) Network, Destined for the Black (Untrusted) Network

The third possibility is that the tunnel terminates in the red (trusted) network, but the traffic is destined for the black (untrusted) network. In other words, although the traffic is destined for an untrusted location, the opposing device has sent the traffic through a safe tunnel to the trusted side of the network. The packets must then pass through the firewall back to the black (untrusted) interface.

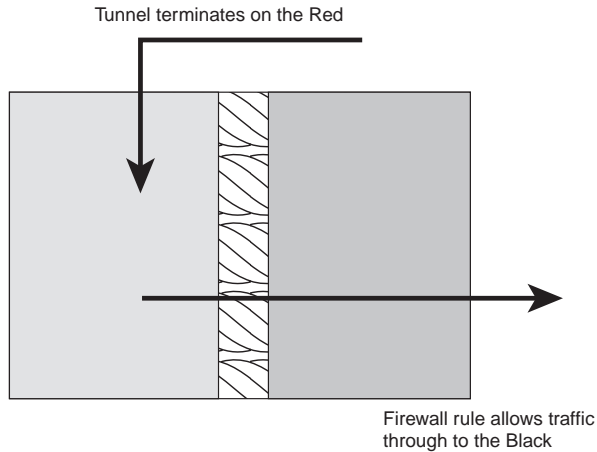


Figure: Tunnel Terminates on the Red (Trusted) Network, Destined for the Black (Untrusted) Network

Tunnel Terminates on the Black (Untrusted) Network, Destined for the Black (Untrusted) Network

Finally, the tunnel may terminate on the black (untrusted) network and the traffic be destined for the black (untrusted) network. In this case the packets do not need to cross the firewall.

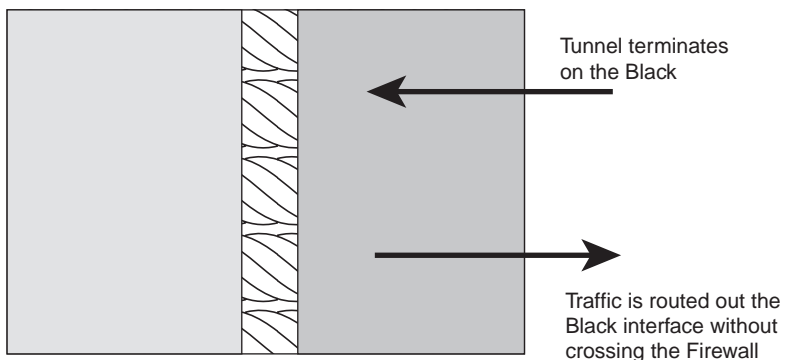


Figure: Tunnel Terminates on the Black (Untrusted) Network, Destined for the Black (Untrusted) Network

Related Information

Tunnel Modes (page 5-20)

One-Way Out Firewall Rules (page 5-24)

One-Way In Firewall Rules (page 5-22)

Load Balancing and Redundancy

| | |
|----------------------|-----|
| Load Balancing | 6-1 |
| Redundancy | 6-2 |

Load Balancing

Given the presence of more than one VPN Gateway in parallel, it makes sense that each VPN Gateway handles an equal portion of the traffic. This equal portioning is called load balancing, which is accomplished in two ways. Given that a tunnel is established with the VPN Gateway that answers first and that the VPN Gateway that answers first does so because it is not busy, the load should be fairly evenly distributed.

In addition, the number of clients set for Client IP can be used to divide up the total load among the VPN Gateway devices. As shown in the following example, if the total number of clients desired is 60 and there are two VPN Gateway devices, the number of clients on each VPN Gateway should be set to 30. In this way a maximum of 30 clients could establish tunnels with each VPN Gateway.

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|----------------------|----------------------|
| Group name | sales | sales |
| Client IP | 10.1.1.193 | 10.1.1.225 |
| Number of clients | 30 | 30 |
| Secure profile (must be previously defined) | dialup | dialup |
| Tunnel mode | Red | Red |
| IP route | Not required | Not required |

Related Information

[Redundancy \(page 6-2\)](#)

[Tunnel Modes \(page 5-20\)](#)

[Tunnel Types \(page 5-8\)](#)

Redundancy

Because the VPN Gateway is such a critical component of a virtual private network (VPN), you should have more than one VPN Gateway supporting the network. By placing more than one VPN Gateway in parallel, the network can continue functioning even if one of the VPN Gateway devices has to be shut down for any reason. This is known as redundancy. Another reason for having more than one VPN Gateway in parallel is to handle more than 1024 active sessions, which is the maximum for a single VPN Gateway.

Redundancy can be implemented for single-user tunnels and for multiuser tunnels only. You cannot apply redundancy to site-to-site tunnels. The reason for this is that redundancy relies on the Client IP address, which only exists for remote user tunnels. You need the Client IP for the device on the red network to know which VPN Gateway to send its replies to. In other words, a different set of Client IPs is used on each gateway.

An example of redundancy is shown in the following figure.

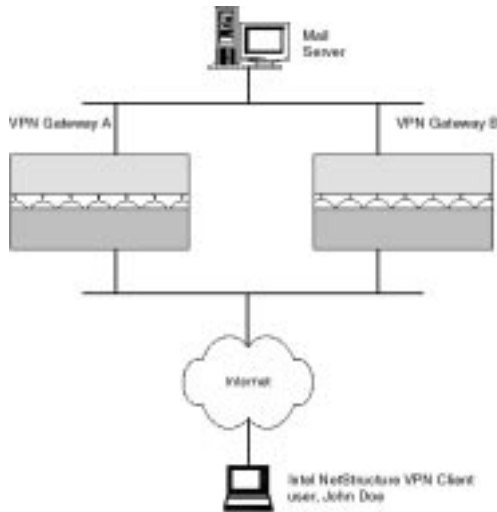


Figure: Enterprise Redundancy

If a client user named John Doe wants to check his mail on the mail server on the red network, he can do so through either VPN Gateway A or VPN Gateway B. If the link definition on the client includes both VPN Gateway devices, the tunnel to the red side is established with the VPN Gateway that responds first. The question for the mail server becomes which VPN Gateway to send its replies through.

Since the tunnel is established only on one VPN Gateway, all replies must go through that VPN Gateway. This is accomplished using Client IPs. Since the set of Client IPs is different on each VPN Gateway, when the mail server uses the Client IP as the destination address on its replies, only the VPN Gateway on which the tunnel has been established accepts the packets for processing. The tunnel definitions for the two VPN Gateway devices appear as shown in the following table.

| Tunnel Definition Parameters | VPN Gateway A | VPN Gateway B |
|---|----------------------|----------------------|
| Group name | sales | sales |
| Client IP | 10.1.1.193 | 10.1.1.225 |
| Number of clients | 30 | 30 |
| Secure profile (must be previously defined) | dialup | dialup |
| Tunnel mode | Red | Red |
| IP route | Not required | Not required |

Related Information

Load Balancing (page 6-1)

Tunnel Modes (page 5-20)

Tunnel Types (page 5-8)

Index

Numerics

3DES..... 2-7

A

AH key length..... 3-4

algorithms 3-2

See also secure profiles

application ports 1-9

asymmetric cryptographic systems..... 2-9, 2-10

authentication headers..... 3-4

authentication methods 3-6, 4-1–4-6

 certificate authentication 4-2

 challenge phrase authentication 4-3

 Entrust authentication 4-6

 RADIUS authentication 4-5

 SecurID authentication..... 4-4

B

black networks 5-2

black tunnels 5-20

brute force attacks 2-13

C

CBC (outer cipher block chaining) 2-8

certificate authentication 4-2

challenge phrase authentication 4-3

crypto period length 3-6

crypto periods 2-12

cryptographic systems..... 2-1–2-10

 3DES..... 2-7

 asymmetric 2-9

 symmetric 2-3

 symmetric v. asymmetric..... 2-10

 triple pass DES..... 2-5

D

Data Encryption Standard (DES)..... 2-4

default gateways 1-8

DES (Data Encryption Standard)..... 2-4

Diffie-Hellman key exchange protocol 2-11

E

Encapsulating Security Payload (ESP)

 AH key length..... 3-4

 authentication headers..... 3-4

 iv length..... 3-4

See also encapsulation

encapsulation 3-3

 Encapsulating Security Payload (ESP) .. 3-1

 Shiva Smart Tunneling (SST)..... 3-1

See also secure profiles

encryption 2-1

Entrust authentication 4-6

F

filters..... 5-6

firewall rules

 and tunnel termination 5-31

 multiuser tunnels 5-18

 one-way in firewall rules 5-22

 one-way out firewall rules 5-24

 single-user tunnels 5-14

 stateful..... 5-4

 stateless 5-2

firewalls..... 5-1

full access

 multiuser tunnels 5-17

 single-user tunnels 5-12

functions of

 Intel NetStructure VPN Client 1-3

 Intel NetStructure VPN Manager 1-2

 Shiva Certificate Authority Server..... 1-3

 VPN Gateway 1-2

I

inbound proxies 5-28

IP addresses 1-6

 network address translation (NAT)..... 5-12

iv (initialization vector) length..... 3-4

K

keepalive..... 3-2

See also secure profiles

| | |
|----------------------|----------|
| key operations | 1-3, 2-9 |
| key pairs | 2-10 |
| key spaces | 2-13 |

L

| | |
|---------------------------|------|
| limited access | |
| multiuser tunnels | 5-17 |
| single-user tunnels | 5-13 |
| load balancing | 6-1 |

M

| | |
|---------------------------------------|-----------|
| modular components of VPN suite | 1-2 |
| multiuser tunnels | 5-16–5-19 |
| firewall rule | 5-18 |
| full access | 5-17 |
| limited access..... | 5-17 |

N

| | |
|--|------|
| names | 3-2 |
| <i>See also</i> secure profiles | |
| network address translation (NAT)..... | 5-12 |
| network configurations of VPN components | 1-5 |
| networks | 5-20 |

O

| | |
|---|------|
| one-way in firewall rules | 5-22 |
| one-way out firewall rules | 5-24 |
| outbound proxies | 5-26 |
| outer cipher block chaining (CBC) | 2-8 |

P

| | |
|----------------------------------|-----------|
| packet handling | 3-7 |
| packet keys | 3-8 |
| packets and packet headers | 1-6 |
| private keys | 1-3, 2-9 |
| proxies..... | 5-26–5-30 |
| public key length | 3-6 |
| public keys | 1-3, 2-9 |

R

| | |
|-----------------------------|------|
| RADIUS authentication | 4-5 |
| red networks | 5-2 |
| red tunnels..... | 5-20 |

| | |
|----------------------|-----|
| redundancy | 6-2 |
| routing tables | 1-8 |

S

| | |
|--------------------------------------|-----------|
| secure profiles | 3-2–3-3 |
| algorithms | 3-2 |
| encapsulation | 3-3 |
| keepalive..... | 3-2 |
| names | 3-2 |
| timeout | 3-2 |
| secure tokens | 1-3 |
| SecurID authentication..... | 4-4 |
| Shiva Smart Tunneling (SST) | |
| authentication methods | 3-6 |
| crypto period length | 3-6 |
| public key length | 3-6 |
| <i>See also</i> encapsulation | |
| single-user tunnels | 5-12–5-15 |
| firewall rule | 5-14 |
| full access | 5-12 |
| limited access..... | 5-13 |
| site-to-site tunnels | 5-9 |
| stateful filtering | 5-4 |
| stateless filtering..... | 5-2 |
| subnet masks..... | 1-6 |
| symmetric cryptographic systems..... | 2-3, 2-10 |

T

| | |
|---|------------|
| TCP/IP | 1-6 |
| IP addresses of devices | 1-6 |
| timeout | 3-2 |
| <i>See also</i> secure profiles | |
| Transmission Control Protocol/Internet Protocol (TCP/IP) | 1-6 |
| trusted networks | 5-20 |
| trusted tunnels | 5-20 |
| tunnel modes..... | 5-20 |
| tunnel termination and firewall rules..... | 5-31 |
| tunnels | 5-1–5-8 |
| firewall rules..... | 5-14, 5-18 |
| full access with multiuser | 5-17 |
| full access with single-user | 5-12 |

Index

| | |
|---------------------------------------|-----------|
| limited access with multiuser..... | 5-17 |
| limited access with single-user | 5-13 |
| modes | 5-20 |
| multiuser | 5-16–5-19 |
| single-user | 5-12–5-15 |
| site-to-site | 5-9 |
| trusted | 5-20 |
| untrusted | 5-20 |
| U | |
| untrusted networks | 5-20 |
| untrusted tunnels..... | 5-20 |
| V | |
| virtual private networking suite..... | 1-1 |
| VPN Gateway | |
| firewall functions | 5-2 |
| standalone mode | 1-5 |