IPSec VPN Guide
Users Manual

4.0

Head Office

VPN Gateway

Partner offices

Internet

Branch offices

Remote/Home/Mobile Workers

Contents

VI. References 133

1   Introduction

IPSec is a transparent security layer for TCP/IP that is commonly used to create and operate Virtual Private Networks (VPNs).

The InJoy IPSec implementation is one of the few end-to-end VPN solutions that is both standards-based and available for multiple Operating Systems.

1.1. Document Scope

This document is designed to provide a concise introduction to IPSec and guide you through its configuration.

Before setting up a VPN, you should be familiar with TCP/IP and the InJoy products of choice. TCP/IP networking should be functioning properly between any hosts you plan to include in your VPN.

Because IPSec represents an addition to your operating system's networking layer, rather than an application or tool, the number of possible configurations and uses for IPSec are endless. This document will provide you with enough instruction to address most common needs.

1.2. Reading This Document

This document has been divided into several distinct parts according to the amount of information different types of readers are likely to need:

Part I.	Learning about InJoy IPSec
Part II.	Getting Started Guide
Part III.	Setting up a VPN
Part IV.	Advanced Features Guide
Part V.	Deployment Examples
Part VI.	References

If you are looking to set up a VPN in the fastest possible way, please refer to section 6, "Using the Quick VPN Wizard".

For a more comprehensive real-world example, please refer to section 10, "A VPN Case Study."

Part I
Learning about InJoy IPSec

2   IPSec Technology Overview

This section is designed to give you a basic overview of IPSec and what it can do for you. For a discussion that supplies more information about the operational details of IPSec, refer to Section 4.

2.1. What is a Virtual Private Network (VPN)?

A Virtual Private Network is a network of related hosts that communicates over the public Internet while at the same time using encryption and authentication technologies to keep the data that it carries private—hidden from public view and protected against unauthorized access or theft.

The typical VPN connects remote branch offices, partners, home workers and mobile users securely over the public Internet:

Head Office

Partner offices

Branch offices

VPN Gateway

VPN Gateway

VPN Gateway

Internet

Remote/Home/Mobile Workers
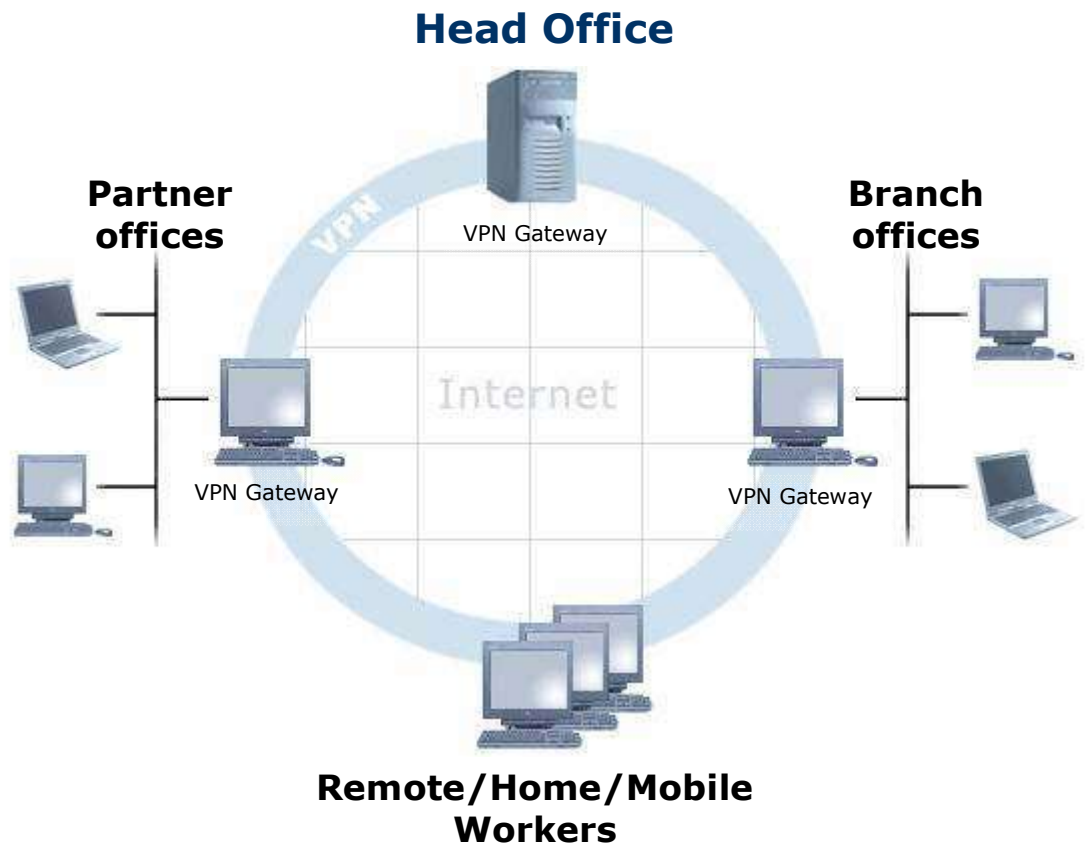
You are most likely to need and use a Virtual Private Network if you own or work for an organization faced with one of these common needs:

- The need to securely link networks or hosts at distant locations or branch offices

- The need to unify disparate IP address ranges into a single, more convenient address space

- The need to secure sensitive network communications from prying eyes

- The need to ensure the integrity of data being exchanged

- The need to provide telecommuters a secure path to the workplace network

- The need to securely replace expensive Frame Relay connections with dial-up or DSL based connections

2.2. Introduction to IPSec

IPSec is the most common technology in use today for creating Virtual Private Networks. As an IPSec user, you'll be taking advantage of a range of robust technologies that are widely used by governments and businesses for secure communication and infrastructure deployment.

IPSec is standards-based, hardware- and software-platform interoperable, well-documented and almost effortlessly scalable. As you work to construct your own Virtual Private Network with IPSec, you will become familiar with three major IPSec components:

1 The **Key Exchange Server**, which establishes the security and authentication policy for each connection and maintains encryption keys

2 The **IPSec Engine**, which carries out the actual encryption and authentication tasks in a running VPN

3 The **Authentication Database**, which contains the information used by the IKE Server to authenticate users and hosts on the Virtual Private Network

IPSec uses IP protocols 50 and 51; a discussion of the complete body of IPSec standards can be found in RFC documents 1828-1829, 2085, 2104, 2401-2412, 2451 and 2857.

IPSec Goals

IPSec was designed to address a few simple goals that are shared by users of TCP/IP networks around the world. By addressing these goals, IPSec provides the following features:

- **Data Origin Authentication**
  When your hosts exchange important information over a public network, you want to ensure that the sending and receiving hosts are known to and trusted by one another.

- **Data Integrity**
  Corrupted information can cost your business wasted time, wasted money, and worse. Any robust network must be able to confirm that the data your hosts receive is identical to the data they send.

- **Data Confidentiality**

Proprietary business data is valuable; any networking technology you use must therefore protect your data from prying eyes.

- **Replay Protection**
  Any networking technology which is vulnerable to attacks or spoofs is a liability when operated on public networks. IPSec protects your data by preventing replay attacks.

- **Automated Key Management**
  By automating the sequencing and periodic exchange of new encryption keys, IPSec ensures that your data is not made public if a key is somehow compromised.

The Key Exchange Server
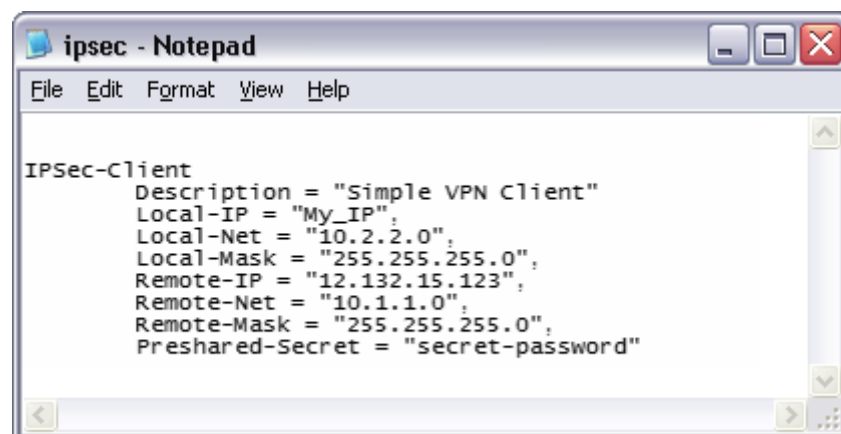
The ability to efficiently exchange keys lies at the heart of IPSec's encryption and authentication model.

The Key Exchange (IKE) Server is a stand-alone IPSec component that performs a number of tasks for VPN hosts. The IKE Server uses the Internet Security Association and Key Management Protocol (ISAKMP) and UDP ports 500 and 4500 for:

- Automated key serialization and exchange

- Negotiation of tunnel encryption

- Negotiation of tunnel authentication

- Negotiation of Network Address Translation (NAT) traversal

The Security Associations Database

The Security Association (SA) is the essential configuration entity in IPSec. To understand the basic responsibilities of an SA and how it may look in a configuration file, take time to study the following example:

One SA specifies the network host, the local internal network (IP addresses), authentication options, and basically all other options relating to an IPSec VPN tunnel.

As IPSec operates it negotiates the user-configured SAs with remote IPSec end-points (through the IKE Server) and the result is a run-time list of

authentication and encryption properties for each connected host in the Virtual Private Network. Together, this list of properties comprises the Security Associations database, a functional summary of network-wide security policy.

The figure below illustrates the use of the IKE Server and the SA between two IPSec end-points:
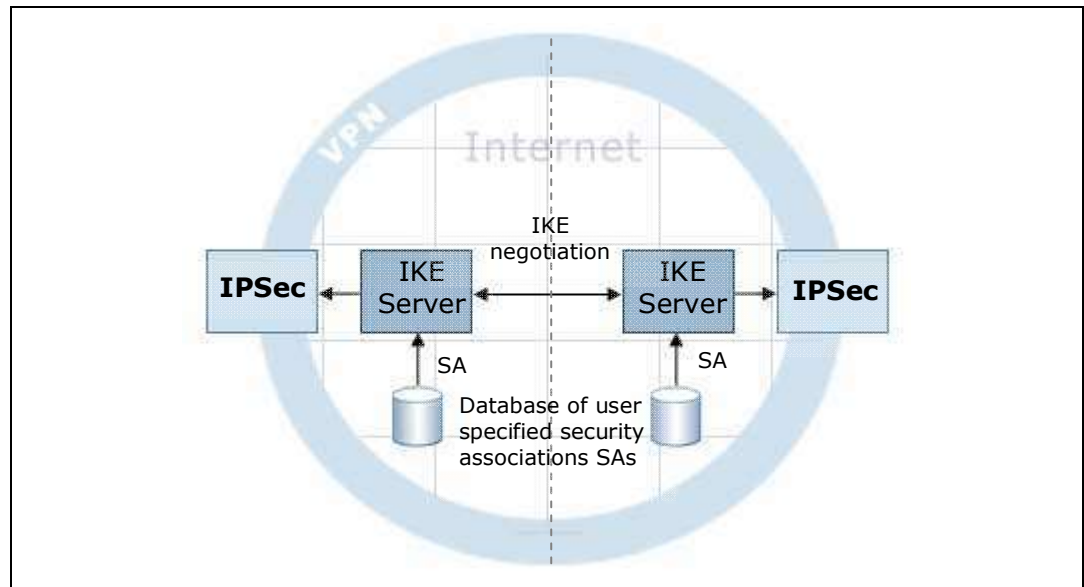
Above, the user-configured SA is provided to the IKE Server for negotiation with the remote IPSec end-point(s). Upon successful negotiation, the run-time SA is provided to the IPSec module, allowing it to apply packet transformation to all IP packets that match the SA

Tunneling

IPSec Tunneling allows IP packets transferred by your network to be completely and securely encapsulated by IPSec. Each packet receives a new header; all address and connection information present in the original packet headers is hidden from public view.

The figure below illustrates the process:

Encrypted Data Transmission

Original Datagram — Tunnel Encapsulation ... Internet ... Tunnel De-encapsulation — Original Datagram
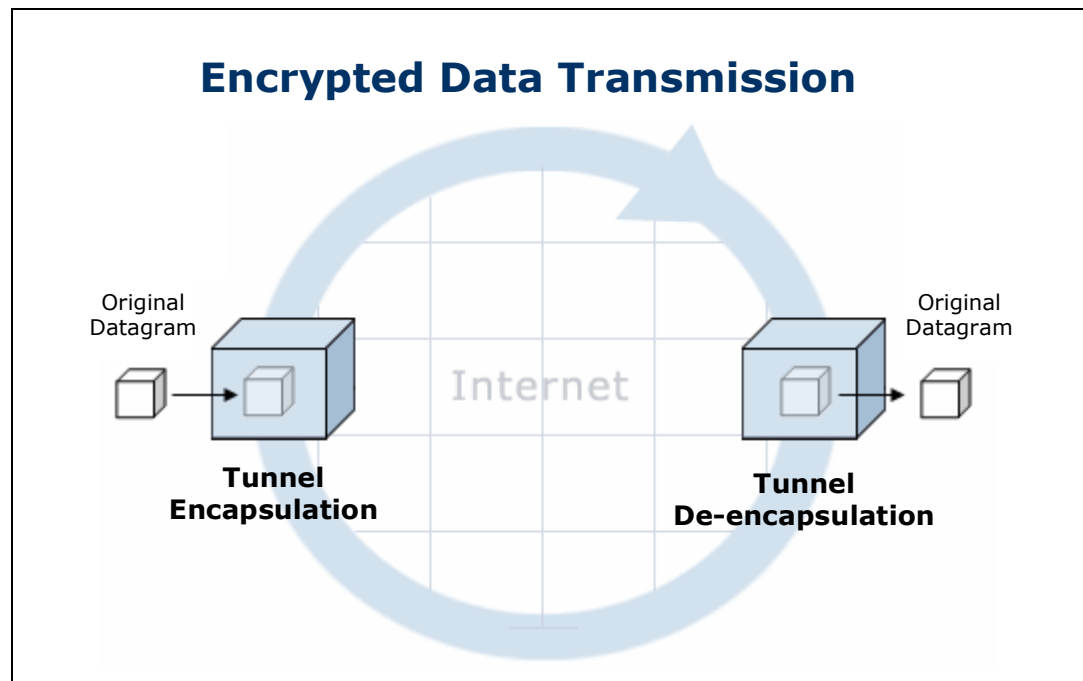
For hosts in the VPN this feature is transparent; tunneled traffic appears to be local and peer-to-peer, even if the datagrams themselves must travel across the public Internet to reach their destinations.

Split tunneling allows a host to maintain tunneled virtual network communication with other hosts in the VPN while at the same time communicating with public Internet hosts (such as a normal web server on the Internet) directly, outside the tunnel. This reduces both processing and traffic overhead on the private network.

Authentication

Authentication encompasses a number of tools used by IPSec to guarantee the identities of remote users and hosts, who can then be "trusted" as members of the VPN.

Encryption

Encryption is a way of making data unreadable to third parties, using one of several mathematically complex scrambling techniques. Even if intercepted, encrypted data is difficult or impossible for everyone but the intended sender and receiver to decode.

3   InJoy IPSec Features

This section is intended to give you an overview of the operational details of InJoy IPSec—the types of authentication and encryption that can be used and the ways in which key generation and exchange can be handled are among these topics. This information will be helpful to you once you begin to make decisions about the deployment of your own InJoy IPSec VPN.

3.1. Traffic Encapsulation Modes

You can choose between two methods of traffic encapsulation when using InJoy IPSec—tunneled mode or transport mode.

- **Tunneled Mode (recommended)**
  In tunneled mode, the IPSec engine completely encapsulates original packets and their headers, and then encrypts the data. Then, new headers are generated for transport. Slightly more processor and network resources are required for tunneled IPSec connections, but it's required for any gateway-to-gateway or host-to-gateway connections.

- **Transport Mode**
  In transport mode, the IPSec engine applies packet transforms only to packet payloads, leaving immutable fields in the original headers intact. This causes less processing and network overhead than tunneled mode, but requires that all IP addressing be host-to-host (without support for subnets)—fine for simple peer-to-peer connections, but problematic for Network Address Translation (NAT-Traversal is not supported in transport mode) situations or road warriors (see Section 3.4).

In almost all IPSec scenarios, the tunneling of traffic is of great importance and of great benefit, thus the use of transport mode is rare.

3.2. Encryption Methods

You can choose between several types of encryption to protect your data and keep it confidential while using InJoy IPSec:

- **DES**
  The DES (**D**ata **E**ncryption **S**tandard) cipher, still in widespread use today (after 3 decades of successful use), uses a 56-bit key and incurs relatively little processing overhead on modern computer systems. However, the DES cipher is also relatively weak; current high-end computing platforms can break DES encryption in a matter of days or even hours. For non-critical proprietary data, DES is an adequate choice.

- **3DES**
  The 3DES cipher is the more current widespread, well-studied, safe and compatible choice. It uses a composite 192-bit key composed of three 64-bit keys to encrypt and decrypt data. As a result, it runs up to three

times slower than DES. The security robustness of 3DES, however, is much better; 3DES is more difficult to break than DES by several orders of magnitude. For critical data, 3DES remains a good and widely compatible choice.

- **Blowfish**
Blowfish is among the most modern and fastest encryption methods. It supports long keys, and is well-respected in the industry. Blowfish run many times faster than DES and it offers several different key lengths: 32, 48, 56, 128 and 448. Each version runs at the same speed. The various key lengths are generally required for compliance with certain export control laws.

- **AES (recommended)**
Short for **A**dvanced **E**ncryption **S**tandard, a symmetric 128 to 256-bit block data encryption technique. AES is a high-speed encryption protocol (comparable to blowfish), which generally out-performs 3DES by factor 2-3.

  The U.S government adopted the algorithm as its encryption technique in October 2000, replacing the DES encryption it used. The National Institute of Standards and Technology (NIST) of the U.S. Department of Commerce selected the AES algorithm, also called Rijndael (pronounced Rhine Dahl), out of a group of five algorithms under consideration.

  As a measure for the AES security, the NIST homepage makes available the following information: "Assuming that one could build a machine that could recover a DES key in a second (i.e., try 255 keys per second), then it would take that machine approximately 149 thousand-billion (149 trillion) years to crack a 128-bit AES key. To put that into perspective, the universe is believed to be less than 20 billion years old".

The InJoy Firewall™ was developed and compiled outside of the USA and is not subject to US export restrictions.

Note: You can use NULL-ESP when you need to be able to tunnel and authenticate data, but don't want the overhead or need the privacy provided by data encryption.

3.3. Authentication Methods

Before secure communication can take place between your InJoy VPN hosts, they must first be able to authenticate themselves to one another. Several authentication types can come into play when new IPSec connections are established.

- **Pre-shared Keys (mostly required)**
Pre-shared Keys, also known as Shared Secrets or Pre-shared Secrets, are the most basic type of authentication used by IPSec. IPSec hosts authenticate themselves by this the Pre-shared key during early negotiation, before any other type of exchange takes place. A Pre-shared Key is simply a password which has been agreed upon in advance by two hosts. Because they are easily compromised, Pre-shared

Keys should be used only when more secure types of early authentication are unavailable.

- **Extended Authentication (recommended)**
Extended Authentication (Xauth) brings user-based authentication to IPSec tunnels. Xauth associates a username and password to each IPSec end-point; after first authenticating with Pre-shared Keys, RSA Digital Signatures or X.509 certificates, a host must be able to supply the correct username and password for the IP address being used before any further exchange of information can occur.

- **RSA Digital Signatures**
The RSA Digital Signature Standard (DSS) can be used in place of Pre-shared Keys when strong security is needed. RSA DSS is a popular, secure mechanism that uses public-private key pairs for authentication. Each host is associated with a pair of cryptographic keys—a private key known only to the host, and a public key that is shared with others. In order to authenticate itself, a host must be able to decrypt (using its private key) a message which has been encrypted (using its public key) by the other party.

- **X.509 Certificates**
X.509 is an RSA-like protocol, generally considered the most comprehensive and safe authentication method in IPSec. Unlike RSA Keys, X.509 public keys are stored in "packages" (basically a file) referred to as the X.509 certificate. Along with the public key, information about the certificate owner (e.g. company and department) is also stored in the certificate. To strengthen security and provide flexible management options, X.509 certificates rely on a trust-inheritance scheme, where each certificate must be signed by a root certificate (or another upper-level certificate from the hierarchy). With this scheme in place, an outside-user with a seemingly valid certificate cannot bypass the server security check, since the user wouldn't have been able to sign his certificate with a proper upper-level certificate (an outside user simply wouldn't have access to the upper-level certificates). The X.509 authentication method also provides a possibility to declare any certificate invalid, for cases where a certificate is stolen or otherwise compromised.

- **Group authentication**
Group authentication is used when VPN requires even more security and provides additional layer of authentication by using additional login/password pair and replacing Pre-shared Keys. However, Group authentication is not well-spreaded among IPSec/IKE implementations.

3.4. Road Warriors

You can use the IPSec Road Warrior feature when you need a VPN Server to accept IPSec connections from hosts whose IP addresses you do not know in advance. This is usually the case for users of dial-up Internet access or other types of dynamic IP networks.

Any security policy you implement for the special IP address 0.0.0.0 will be applied to hosts that have dynamic IP addresses, provided they are able to complete the authentication process.

With Road Warrior support, where the IP address doesn't help identify the peer and where all dynamic IP end-points must share the same pre-shared key, use of a user based authentication method is strongly recommended.

Road Warrior support is not a widely available IPSec standard.

3.5. Key Management

Depending on your needs, you can choose one of three different methods for managing the exchange of encryption keys while using InJoy IPSec.

- **Main Mode (recommended)**
  In main mode, negotiation and key exchange occurs using a secure exchange of six packets between IKE Servers. This method of key exchange is the safest, but is also the slowest. Main mode also offers automatic selection of the peer's proposal, generally making it easier to configure.

- **Aggressive Mode**
  In aggressive mode, negotiation and key exchange occurs using an exchange of three packets between IKE Servers. While faster, this method is less secure because some host information is exchanged in cleartext.

- **Manual Keying**
  In manual keying mode, a pre-defined set of keys is used by both hosts, obviating the need for any type of key exchange at all (and thus for the IKE Server as well). While it is the fastest key management method, manual keying is by far the least secure.

3.6. IPSec Extensions

As an InJoy IPSec user, several extensions to the IPSec standard are available to you; each of these provides extra functionality that is useful for a common set of circumstances.

- **IP Compression**
  IP Compression provides data compression for your IPSec connections. This can provide a significant increase in network throughput, especially for users of slower dial-up connections.

- **NAT Traversal**
  Though it requires extra configuration and imposes some limitations on functionality, InJoy IPSec hosts can be made to communicate through NAT connections.

- **Inner IP**
  Inner-IP is a feature that allows the VPN administrator to assign a virtual IP address (a "Red Node IP") to any connecting IPSec client. For instance, a remote dial-up user with a constantly changing ISP assigned dynamic IP address can be assigned a static internal IP address such as 10.2.2.1 (through the use of NAT within IPSec). This greatly simplifies the administration tasks faced by network administrators—all addresses on the VPN can fall within a single, unified internal address range.

Part II
Getting Started

4 Starting IPSec

This section is designed to help you to activate the software components necessary to use InJoy IPSec. This process involves three simple steps:

Step 1:	Step 2:	Step 3:
Ensure the software requirements are met.	Enable the IPSec plugin in the InJoy software.	Verify the IPSec support is loaded.

4.1. InJoy IPSec Requirements

Before attempting to activate InJoy IPSec, ensure your system meets the following basic requirements:

- **The loopback interface (127.0.0.1) must be configured.**
  The loopback interface is a special TCP/IP interface that provides TCP/IP applications, such as the Pluto IKE Server, with a static IP address for connecting back to the local PC. The IP address of the loopback interface is typically defined to be 127.0.0.1.

  The loopback interface exists by default in Windows 2000 and XP installations. On Linux and OS/2, it is likely to exist and if not, it can be created with the simple command:

  ```
  ifconfig lo 127.0.0.1
  ```

  For more information about the loopback interface, please consult your Operating System specific TCP/IP literature.

- **No existing IPSec software can be running**
  The Windows Operating System runs its own limited version of IPSec. To use InJoy's IPSec, turn Windows IPSec off in the list of Services. On W2K, the Services are located here:

  ```
  Start->Settings->Control Panel->Administrative Tools->Services
  ```

  On Windows XP, the Services are located here:

  ```
  Start->Administrative Tools->Services
  ```

  In the list of services, locate "IPSEC Services" (and possibly the "IPSEC Policy Agent" – on Windows 2000) and stop the service. Changing the service to "Manual" prevents future auto-starting of the service.

Latest versions of InJoy software automatically stop the built-in Windows version of IPSec, so the above steps are not necessary.

- **Your InJoy product software level must include IPSec support**
  IPSec is an advanced feature, not included in all software levels of the InJoy products. Before you attempt to configure and start InJoy IPSec, ensure your InJoy software is licensed at a level that includes IPSec support.

4.2. Enabling the IPSec Software Components

The InJoy IPSec Plugin and Pluto IKE Server are included in the InJoy software distribution. To enable the plugin in the InJoy Firewall™ product, open the Firewall Properties dialog, select the "Firewall Server" tab to access the plugin selections and check "IPSec VPN Support".
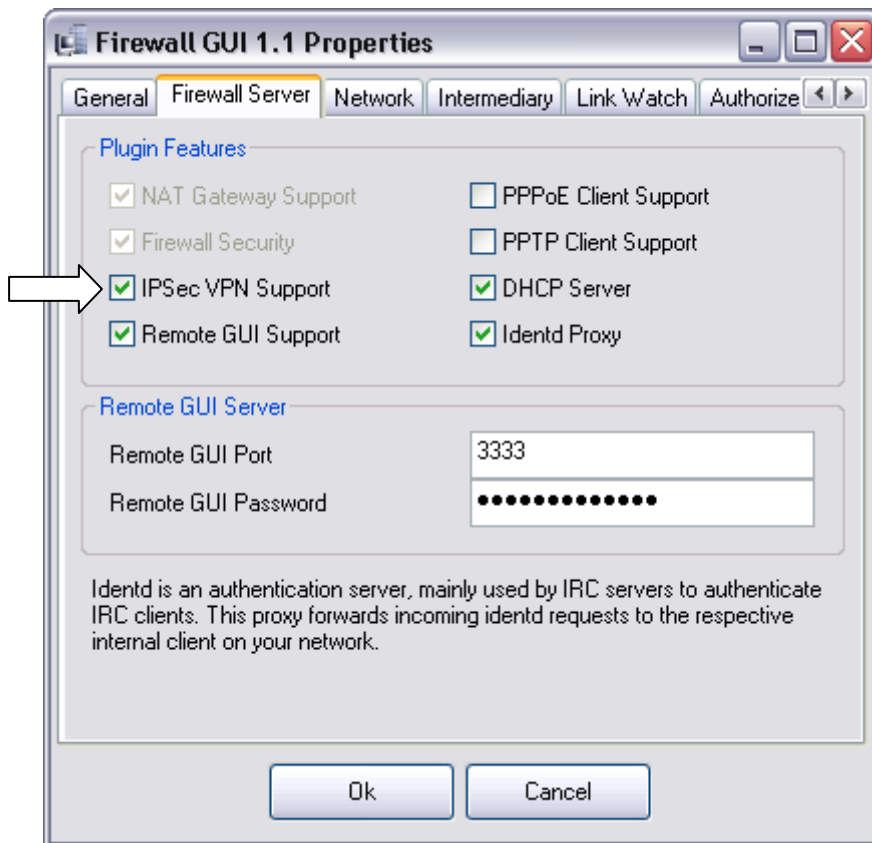
The Firewall Server must be restarted to activate any changes.

4.3. Verifying IPSec Support

In the InJoy Firewall™ product, you can ensure that IPSec support is successfully loaded by verifying two things:

1 The Firewall GUI must offer a menu to configure IPSec tunnels.

```
File              ▶
Firewall          ▶
IPSec             ▶    Tunnel Workshop...
DHCP Server       ▶    User Administration...

Window            ▶    VPN Quick Wizard...
Monitors          ▶
Minimize               Reload Configuration

Exit
```
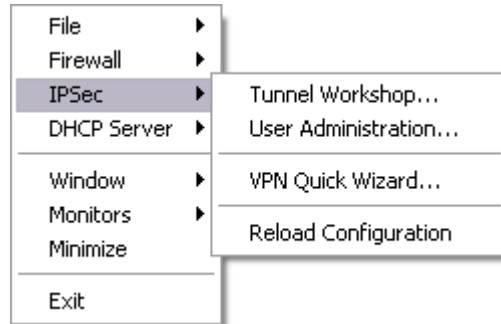
2 New messages should be written to **logs/ipsec.log** and **logs/pluto**, indicating the operational status of the IPSec and IKE Server components, respectively.

With IPSec and the IKE Server being operational and ready to be configured, you are ready to further study the configuration options and plan your VPN.

5 Configuration Overview

This section is designed to help you quickly become familiar with the IPSec configuration, including:

- GUI tools for IPSec configuration
- IPSec configuration files and their formats
- How to activate configuration changes.

5.1. What Needs Configuration?

Before you can deploy IPSec, you must define the involved network endpoints in what is known as a **Security Association (SA)**. Each SA contains information about an IPSec tunnel, including:

- The public IP addresses of the involved VPN end-points.
- The IP address range of the private intranets (if any), behind the IPSec end-point.
- Protocols to use for authentication, encryption and KEY management.
- IPSec extensions, such as IP compression and NAT traversal.

5.2. How Is IPSec Configured?

The following wizard-style configuration dialogs are available in the InJoy products to configure IPSec locally or remotely:

- The **VPN Quick Wizard**
- The **Tunnel Workshop**

The graphical configuration is available from the IPSec sub-menu of the InJoy Firewall GUI – illustrated below:
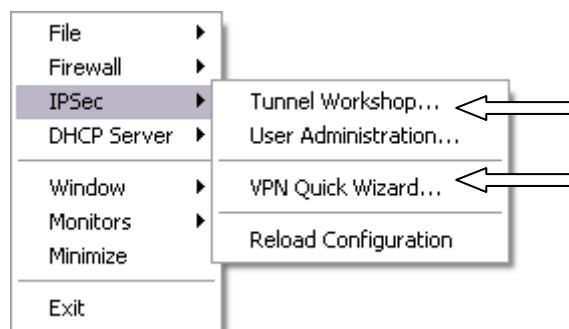
In addition to the graphical configuration, IPSec can also be configured through plain-text configuration files, covered later in this chapter.

The VPN Quick Wizard

The VPN Wizard makes it easy for anyone to quickly configure a powerful VPN through simple modifications to pre-configured template SAs.
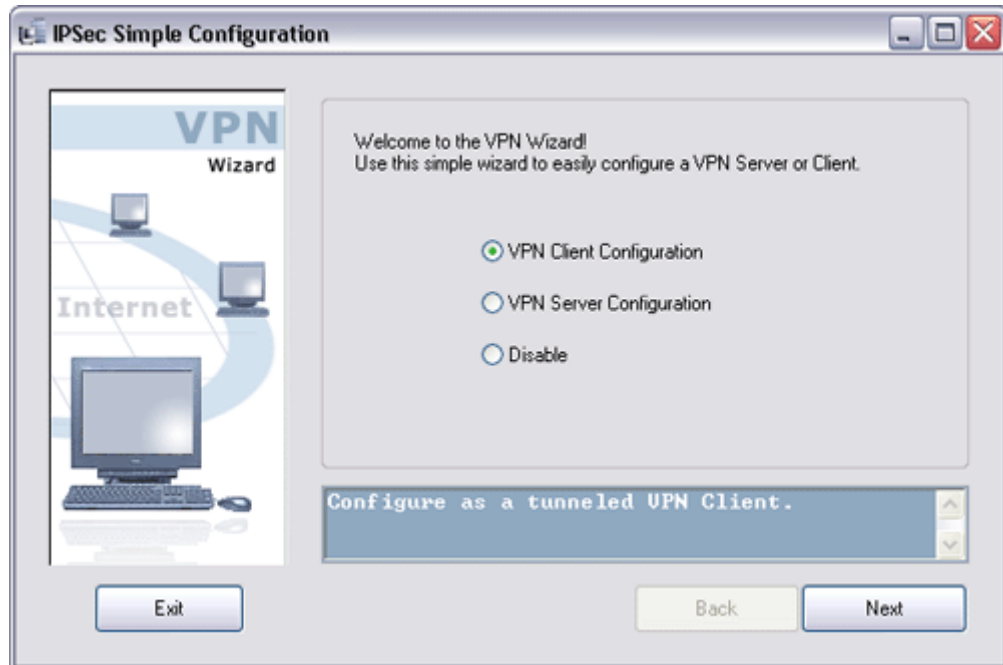
Two templates exist, one configured as a VPN Server/Concentrator and one configured as a VPN Client.

The VPN Server is configured to accept connections from both fixed and dynamic IP addresses, even through a NAT Gateway – using the NAT Traversal feature. The VPN Server authenticates VPN Clients through a combination of the Preshared Key (a VPN password) and a login based user-account (user-id and password).

You decide which encryption standard to use and specify what network is behind each IPSec endpoint. On the VPN Server you maintain the simple user database and optionally assign a Virtual IP address to the remote VPN Clients directly from the VPN Wizard.

The VPN Wizard configuration can be carried out locally or remotely, allowing the network administrator to maintain a complete VPN network from a central location.

You should use the VPN Wizard when:

- The VPN is to use a Client / Server based topology

- All VPN Clients can share the same overall IPSec/IKE policy

- Only one internal network exists behind the different VPN end-points

- Third-party interoperability requirements are minimal

The Tunnel Workshop

The Tunnel Workshop provides a comprehensive configuration option, allowing you to fully control all the individual IPSec options and security associations.
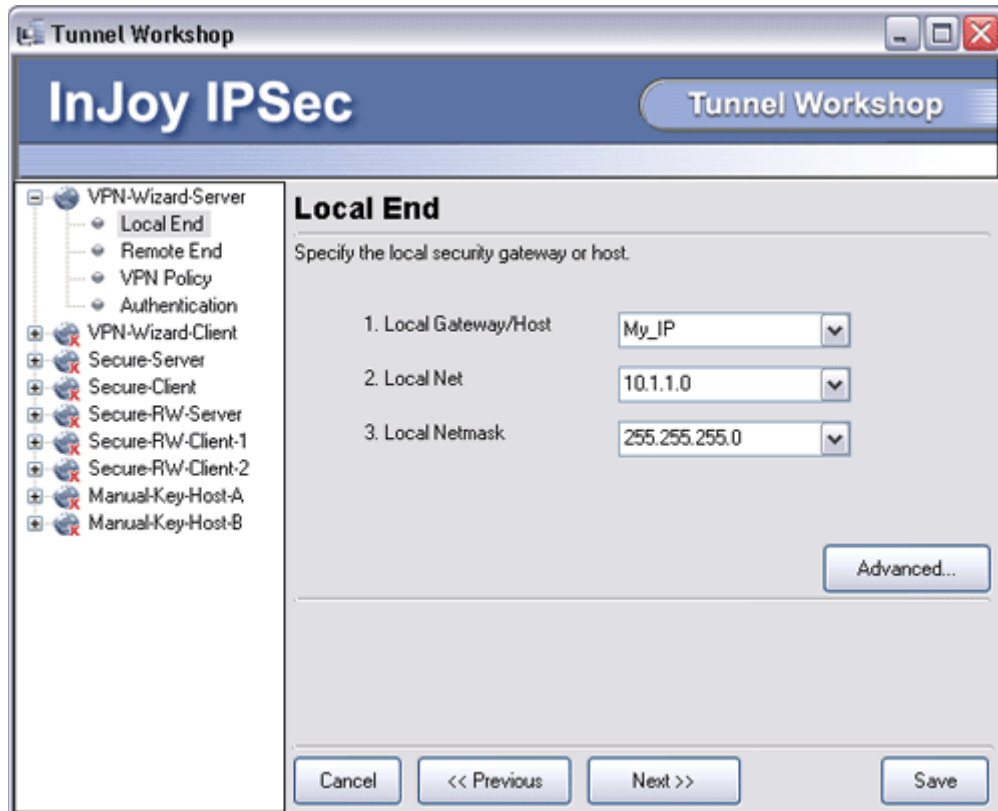
The Tunnel Workshop also allows you to edit the VPN Wizard template SAs, or add additional SAs to a VPN already configured through the VPN Wizard.

You should use the Tunnel Workshop when:

- Many different (types of) tunnels must be configured

- Third party interoperability is of key-importance

- The VPN Wizard isn't sufficient

5.3. Which IPSec Configuration Files Exist?

You can also configure InJoy security associations and other aspects of the IPSec behavior using several text-based configuration files:

- **ipsec\ipsec.cnf** – Security Associations (VPN Tunnels).

- **ipsec\options.cnf** - Global IPSec parameters.

- **ipsec\vpn-auth.cnf** – X-Auth users (server side feature).

- **pluto.secrets** or **pluto.sec** – IKE Server RSA Digital Signature file.
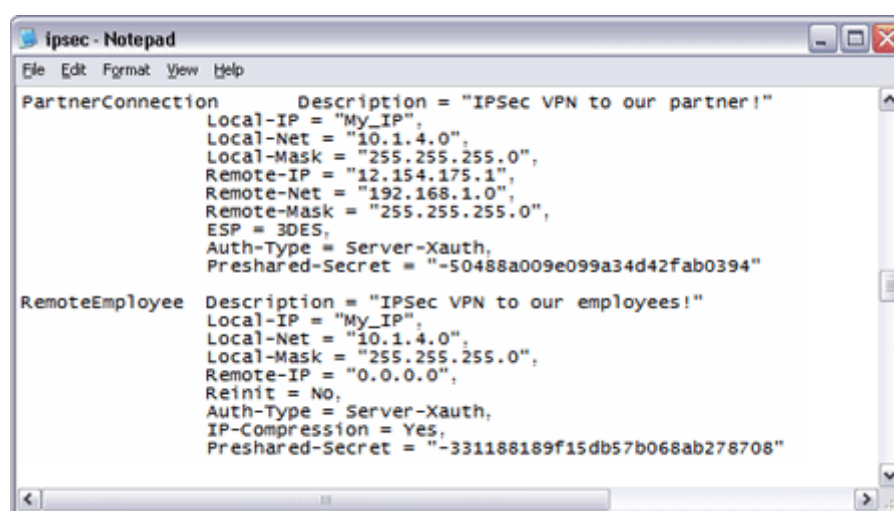
You can edit any of these files with any of the commonly available text editors, such as Notepad or Emacs.

For simple VPNs that don't make use of RSA Digital Signatures and extended authentication, the network administrator will typically only be changing the **ipsec.cnf** file.

The configuration files in the ipsec directory all follow a common format, where typically only non-default configuration attributes actually need to be specified. Default values for all configuration attributes are loaded from files of similar names in the template directory.

IPSEC.CNF

This is the file that contains the actual Security Associations, for example created or edited using the Tunnel Workshop. The file is made up of one or more configuration records, with one record for every tunnel:

Each individual SA is a comma-separated list which begins with a line containing the SA name and description. The rest of the lines in an SA contain a keyword and a value (option pairs). The keywords in the file are (mostly) similar to those you'll encounter while using the Tunnel Workshop and the values are editable by you, the network administrator.

The names of the individual records in **ipsec.cnf**, such as "PartnerConnection", are non-significant and only used for logging purposes.
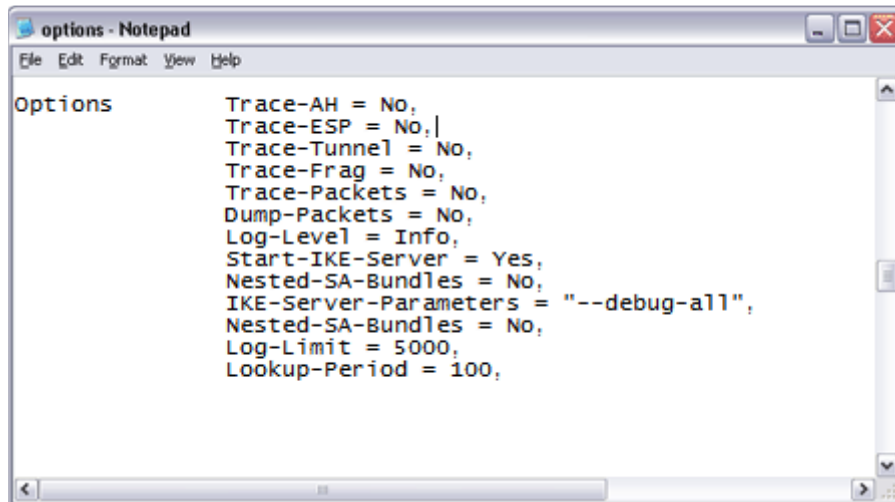
For more information about the configuration attributes and their possible values, refer to "Appendix D – Configuration Attributes".

OPTIONS.CNF

Using a single record, the general options of IPSec are configured in the file **options.cnf.** Through this file, the IPSec administrator can control auto-starting of the IKE Server, size limits of the IPSec log files, SA nesting, and verbose levels.

In the file you will see a number of configurable keyword and option pairs. Using these, you can enable the various types of IPSec tracing (including AH

or ESP header tracing), alter the IPSec logging level, or decide whether the Pluto IKE Server should be started automatically when IPSec starts or not.
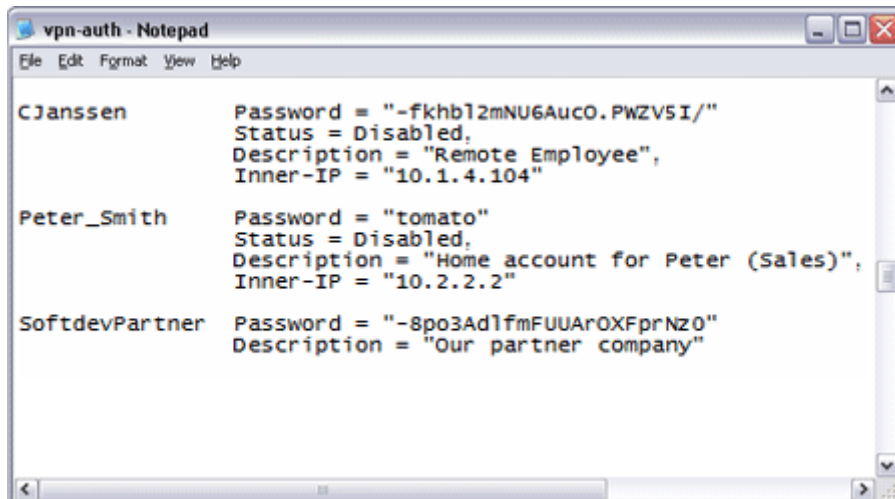
```
Options         Trace-AH = No,
                Trace-ESP = No,|
                Trace-Tunnel = No,
                Trace-Frag = No,
                Trace-Packets = No,
                Dump-Packets = No,
                Log-Level = Info,
                Start-IKE-Server = Yes,
                Nested-SA-Bundles = No,
                IKE-Server-Parameters = "--debug-all",
                Nested-SA-Bundles = No,
                Log-Limit = 5000,
                Lookup-Period = 100,
```

For more information about the configuration attributes and their possible values, refer to "Appendix D – Configuration Attributes".

VPN-AUTH.CNF

This file contains the server side user accounts (username and password pairs), used by the Authentication Module to perform Extended Authentication (Xauth).

```
CJanssen        Password = "-fkhbl2mNU6AucO.PWZV5I/"
                Status = Disabled,
                Description = "Remote Employee",
                Inner-IP = "10.1.4.104"

Peter_Smith     Password = "tomato"
                Status = Disabled,
                Description = "Home account for Peter (Sales)",
                Inner-IP = "10.2.2.2"

SoftdevPartner  Password = "-8po3AdlfmFUUArOXFprNz0"
                Description = "Our partner company"
```

You will see a number of sections related to particular users. Each section begins with a username and contains keyword and value pairs for the user's description, password, account status (enable or disable) and inner IP, when applicable.

For more information about this file, please refer to Section 16.2, "Extended Authentication (Xauth)".

PLUTO.SECRETS

This file is used only when IPSec is configured to authenticate users via the RSA Digital Signatures protocol.

The file holds a table of secrets. These secrets are used by the Internet Key Exchange Server, to authenticate other hosts.

An RSA private key is a composite of eight generally large numbers. The notation used is a brace-enclosed list of field name and value pairs. An RSA private key may be generated by rsasigkey tool.

For details on using RSA, please refer to Section 16.3, "RSA Digital Signatures."

5.4. How do I activate Configuration Changes?

Once you have made changes to the IPSec configuration, you can activate your changes using one of the methods discussed below.

Either method causes the IPSec configuration to be reloaded, IPSec tunnels to be re-negotiated and a message to appear in the product activity log (to indicate that this has taken place).

Command Line Activation

From the command line, you can trigger IPSec reloading by using the **sync** utility command with the option **-ipsec**:

GUI Activation

In the InJoy Firewall™ GUI, you may activate changes by using the pop-up menu:
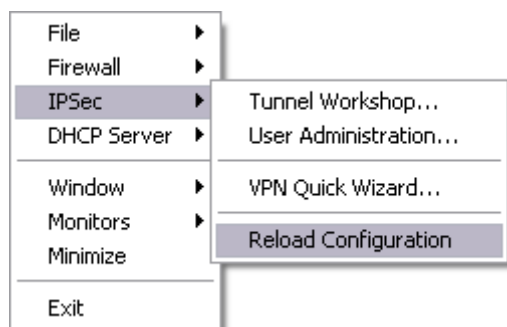
6 Using the Quick VPN Wizard

The VPN Wizard provides a simple method for configuring a secure company-wide VPN – in a matter of minutes. It makes use of an "already configured" sample VPN, as well as the built-in user-database to provide a powerful end-to-end VPN solution.

The VPN Wizard is special in the way it uses a single server-side SA to handle all the remote VPN Clients. For each individual Client that connects, a new separate tunnel is dynamically spawned.

6.1. VPN Wizard Overview

By using the VPN Wizard, you can set up a standard, well-tested and yet flexible VPN – with the following features:

- **Tunneling / Encryption**
  The VPN Wizard tunnels traffic between IPSec endpoints and their internal network. Data is scrambled by the encryption standard you choose, for example 1DES, 3DES, AES, Blowfish, or similar.

- **VPN Password Security**
  A single pre-shared secret (really just a normal encrypted password) provides the first line of defense in almost any VPN. Any IPSec Client trying to connect to your VPN must know this password.

- **Client / Server Based Authentication**
  Using the built-in VPN user-database and the extended authentication (X-Auth) feature of InJoy IPSec, the VPN Wizard sets up an authenticating VPN Server, providing user account based login for remote VPN Clients. Managing VPN users is easy and requires no additional software.

  Note: User-ID & Password of the VPN Client can be stored encrypted on their harddisk or the user can be prompted at each login. Prompting users for their login with every new connection helps you address the security vulnerabilities related to stolen laptop computers and PCs that may be left unattended.

- **Virtual IP Addresses**
  The VPN Server can optionally assign an Inner-IP address to remote VPN Clients, providing easier administration throughout the organization. The Inner-IP allows for example the dynamic IP address assigned to dial-up clients to be NAT translated into a virtual internal IP address (e.g. 10.2.2.1). The result is fixed IP addresses everywhere, making it easier to maintain company firewall policies and routing tables.

- **NAT Traversal (NAT-T)**

Allows VPN Clients situated behind NAT devices to effortlessly connect to the VPN Server on the Internet – without any additional configuration in the remote Firewall/NAT device.

When the IPSec Server runs behind a NAT device, simple Firewall redirection rules are required to forward traffic on UDP port 500 & 4500 to the internal VPN Server – and no non-standard protocols are used (when NAT-T is enabled).

- **IP Compression**
  Compression of VPN traffic provides increased network bandwidth, modestly enhances security, and saves money. The VPN Wizard enables compression by default.

- **Fixed and Dynamic IP address support**
  The VPN server supports remote VPN Clients with dynamic IP addresses, known as Road Warriors, and also VPN Clients with fixed IP address.

- **30 minute key life-time**
  Keys are generated every 30 minutes, providing powerful security, while limiting the well-known issues with "dead" VPN tunnels (e.g. dial-up users who lost the Internet connection) that clutter the monitor windows.

The features of the VPN Wizard are pre-configured, but **not** fixed. The "volatile" settings (such as passwords, encryption standard and IP addresses) can be easily changed directly in the VPN Wizard dialogs. More complicated VPN Wizard settings (such as NAT-T, IP Compression and Key life-time) can be tweaked in the tunnel workshop – retaining flexibility, while eliminating the complexity for beginners.

The VPN Wizard relies on standard features, allowing you to find more information about any of its features throughout this manual.

6.2. Starting the VPN Wizard

To start the VPN Wizard, select **"IPSec | VPN Quick Wizard"** in the pop-up menu of the InJoy Firewall GUI. You will see the simple start-up screen, prompting you to set up either a VPN Server or a VPN Client.

A third option allows you to disable the VPN Wizard functionality completely.
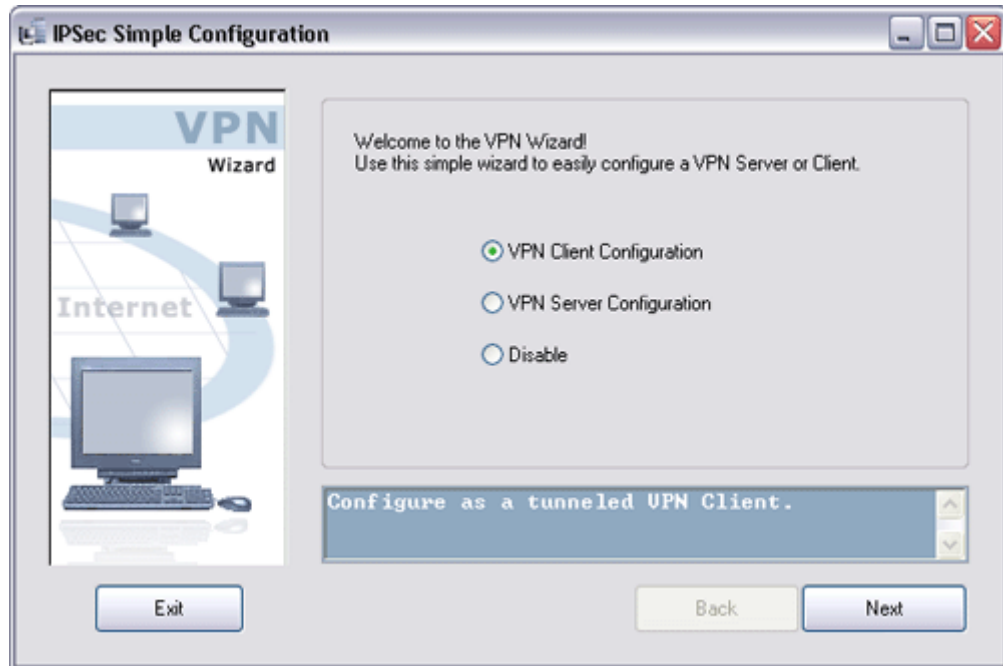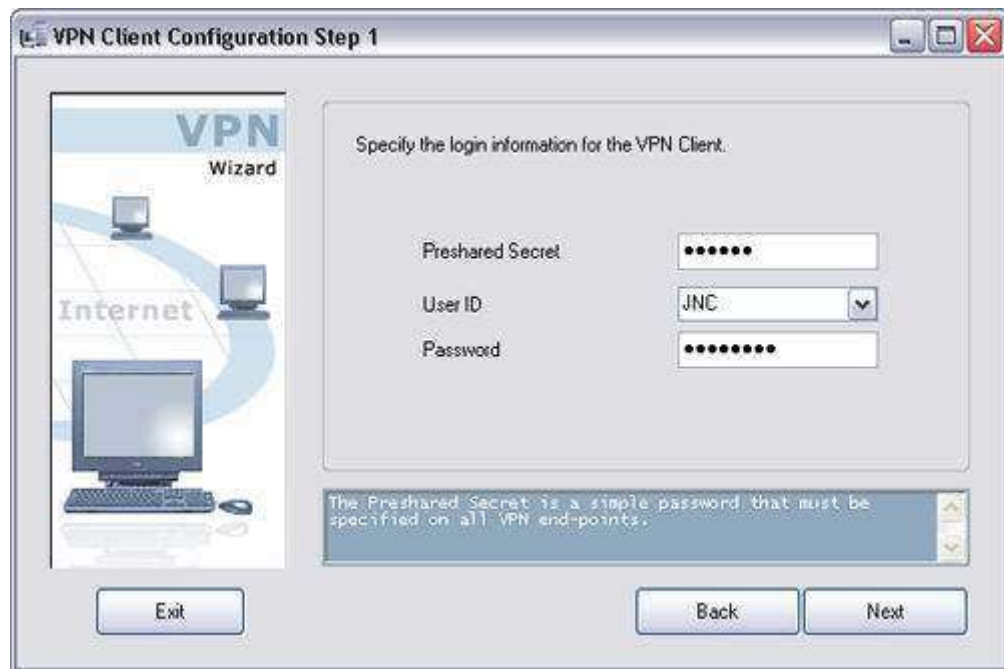
The VPN Wizard includes context sensitive on-screen hints at the bottom of each dialog, making it easy for anyone to quickly understand the questions and make the right decisions.

6.3. Setting up a VPN Server or Client

VPN Wizard - Step 1

After selecting **VPN Client** or **VPN Server**, the VPN Wizard takes you to the next screen, which allows you to specify the Extended Authentication (XAUTH) login and the pre-shared secret. For the VPN Server, you can only enter the pre-shared secret:

VPN Wizard - Step 2

Step 2 of the VPN Wizard requires that you specify the Internal Network behind the VPN end-point and the desired encryption algorithm to be used. You can select either a specific encryption standard or specify "Yes" in the "Encryption" field (on the Server side) to allow VPN Clients to select their own Encryption standard. Yes on both sides will result in 3DES encryption to be used.
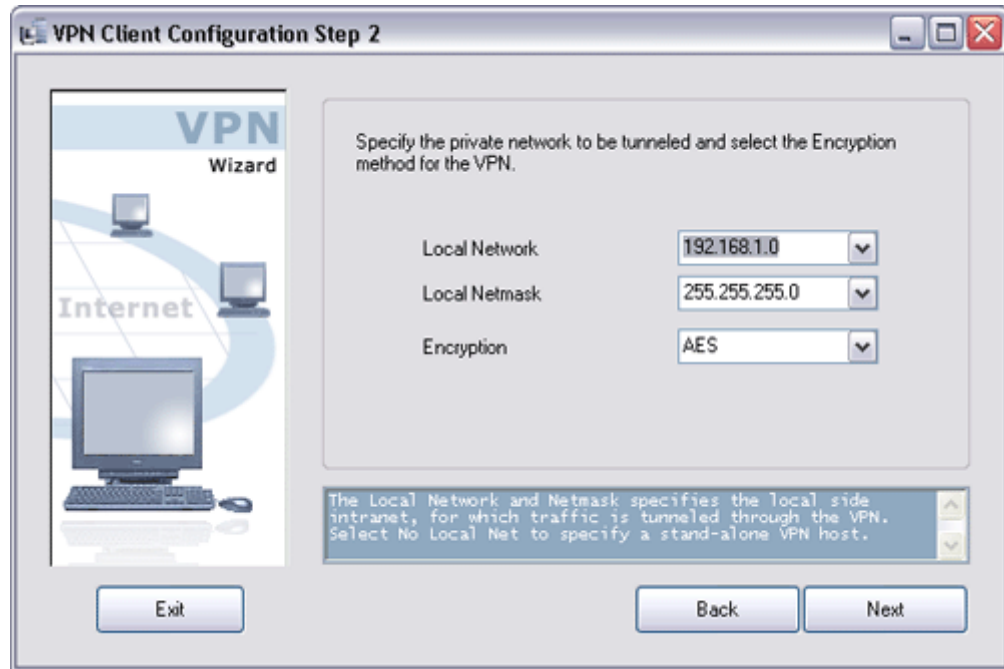
VPN Wizard - Step 3

Step 3 for the VPN Client lets you specify the IP address or a fully-qualified domain name of the VPN Wizard Server to connect to. In addition, it must be selected whether the VPN Client is using a dynamic or static IP address. In server mode, this step is skipped.
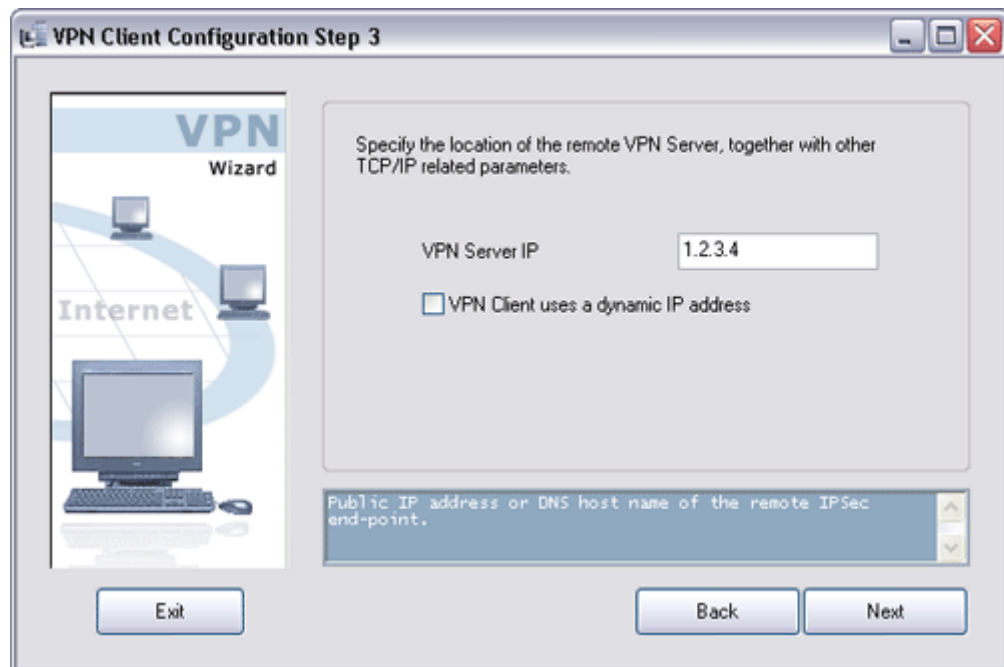
VPN Wizard - Finishing

After you have entered the desired settings, the VPN Wizard will provide a summary of options and allow you to save and activate the configuration.

Save and Activate VPN Configuration

VPN Client configuration

- SA Name   VPN-Wizard-Client
- Local Gateway   My_IP
- Local Net   192.168.1.0 / 255.255.255.0
- Remote Gateway   1.2.3.4
- Remote Net   192.168.4.0 / 255.255.255.0
- VPN Policy   No (AH) / AES (ESP)
- Initiate IKE   Yes
- Inner IP   None
- Authentication   Client-Xauth

Exit   Back   Save and Activate
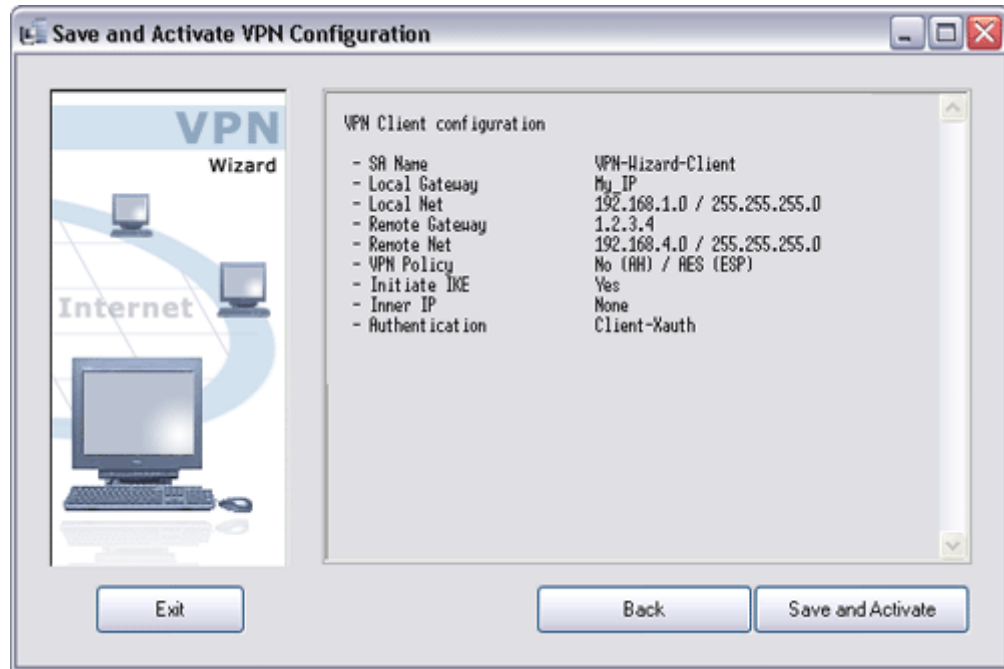
Once you click "Save and Activate", IPSec will enable your configuration and if you were setting up a VPN Client, a tunnel will be attempted set up with the remote VPN Server.

6.4. Configuring VPN Users

In step-2 of the VPN Wizard Server configuration you can click "Configure Users" to set up VPN Client user accounts. This dialog can be also be brought up at any time by selecting **"IPSec | User Administration"** in the Firewall GUI pop-up menu.
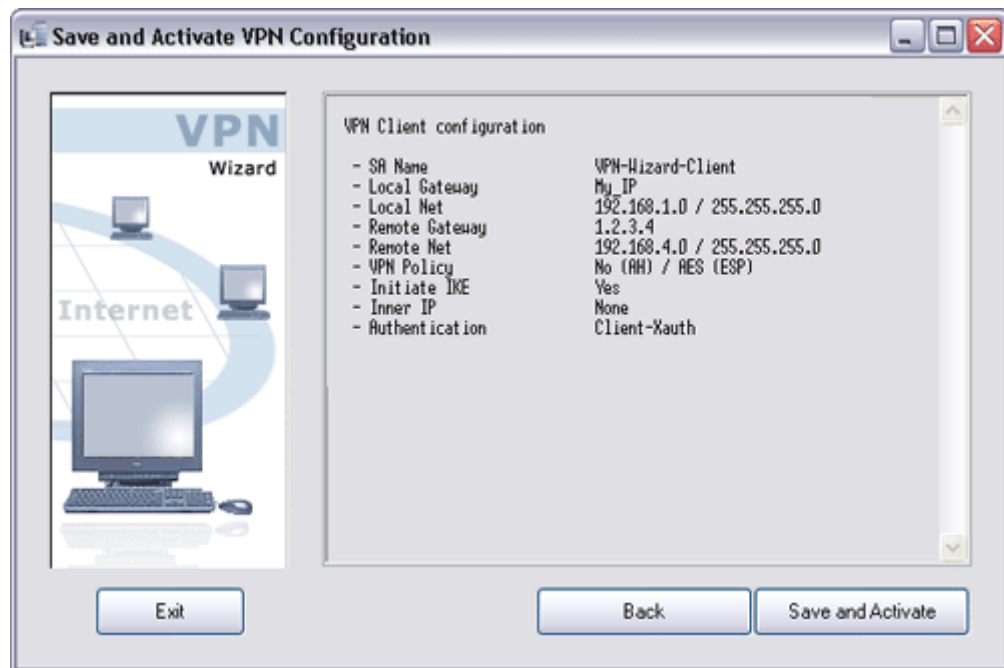
7 Using the Tunnel Workshop

You can use the Tunnel Workshop to maintain the database of IPSec tunnels. To start the InJoy Tunnel Workshop, choose **"IPSec | Tunnel Workshop"** from the firewall GUI popup menu – and wait for the following dialog to appear.
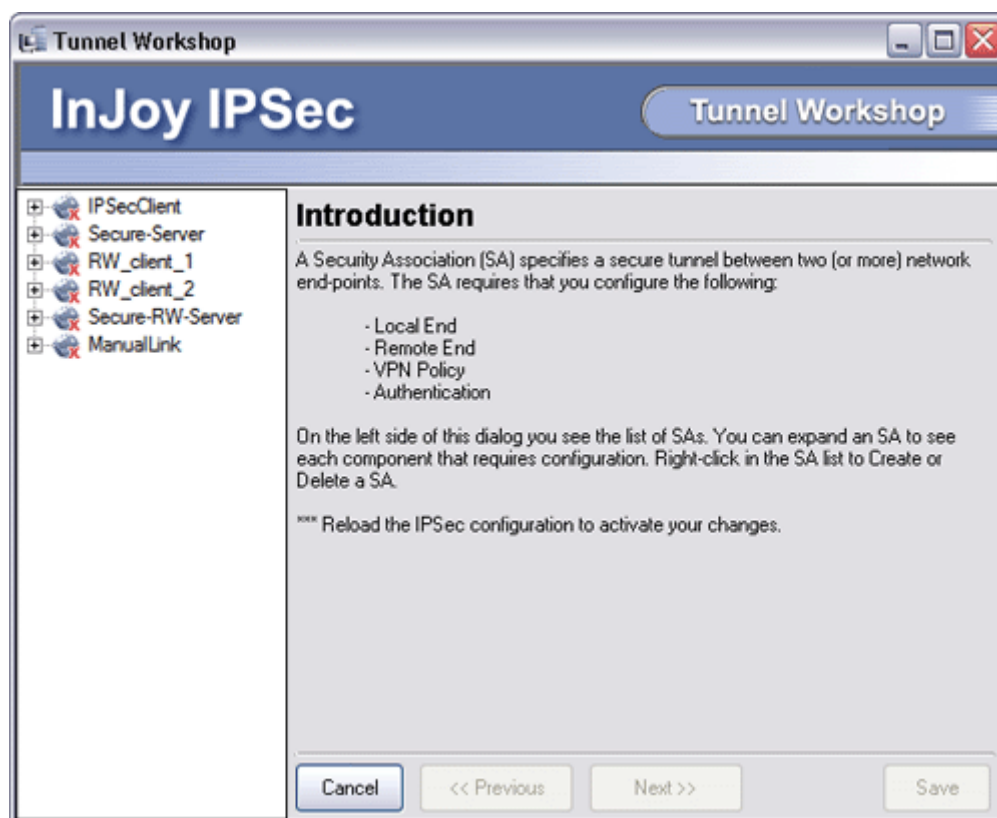
On the left side of the Tunnel Workshop, you'll see a list of the existing security associations. On the right side of the Tunnel Workshop are the settings that make up the security association.

7.1. Creating Security Associations

To create a new security association, right-click in an empty area on the left side of the Tunnel Workshop and select **Create New SA** from the popup menu; this will start a wizard that will guide you through several configuration dialogs.

In each dialog, you can click in any text field or on any drop-down list to display a concise and informative paragraph designed to guide your selection or choice. After completing all of the fields a dialog, you can click **Next** to proceed to the next dialog.

SA Setup

In the SA Setup dialog, you will choose a name and a brief description for the security association you are about to create.
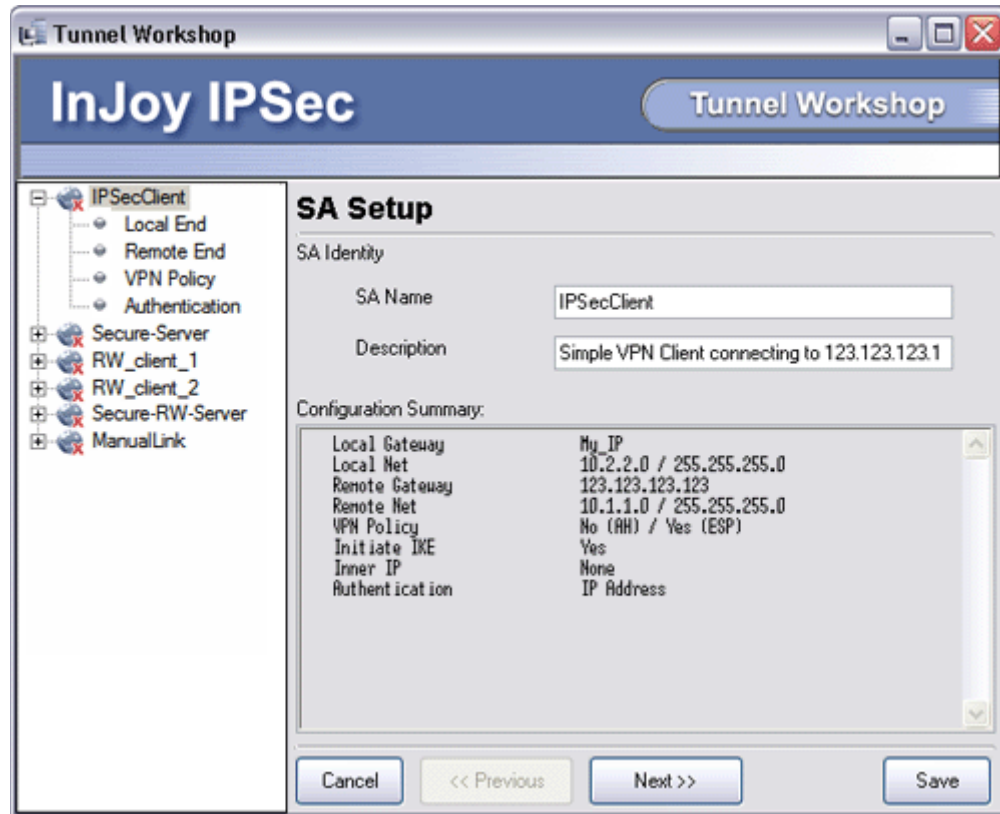
Local End

In the Local End dialog, you will configure the IPSec network interface on the local machine, including:

- The public IP address for the local interface

- A private address range for the local network (if any)

- A network mask for the local network or host

The **Advanced** button in the Local End dialog leads to a sub-dialog that offers you the choice to tunnel using the remote host's internal IP address, rather than its public IP address.
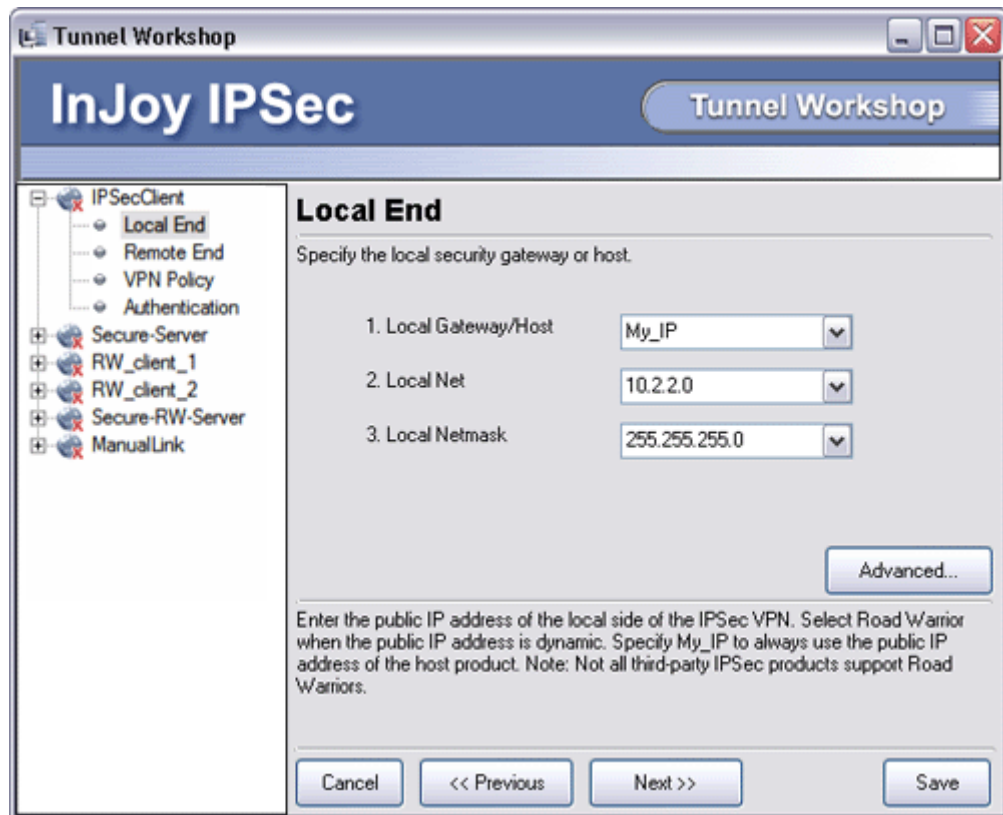
Remote End

The Remote End dialog is similar to the Local End dialog, but applies to the host on the other end of the IPSec connection. In the Remote End dialog, you will configure the IPSec network properties of the remote machine:

- The public IP address for the remote interface

- A private address range for the remote network (if any)

- A network mask for the remote network or host

The **Advanced** button in the Remote End dialog leads to a sub-dialog that offers you the choice to exclude the remote host itself from this SA and apply your configuration only to hosts within the remote network.
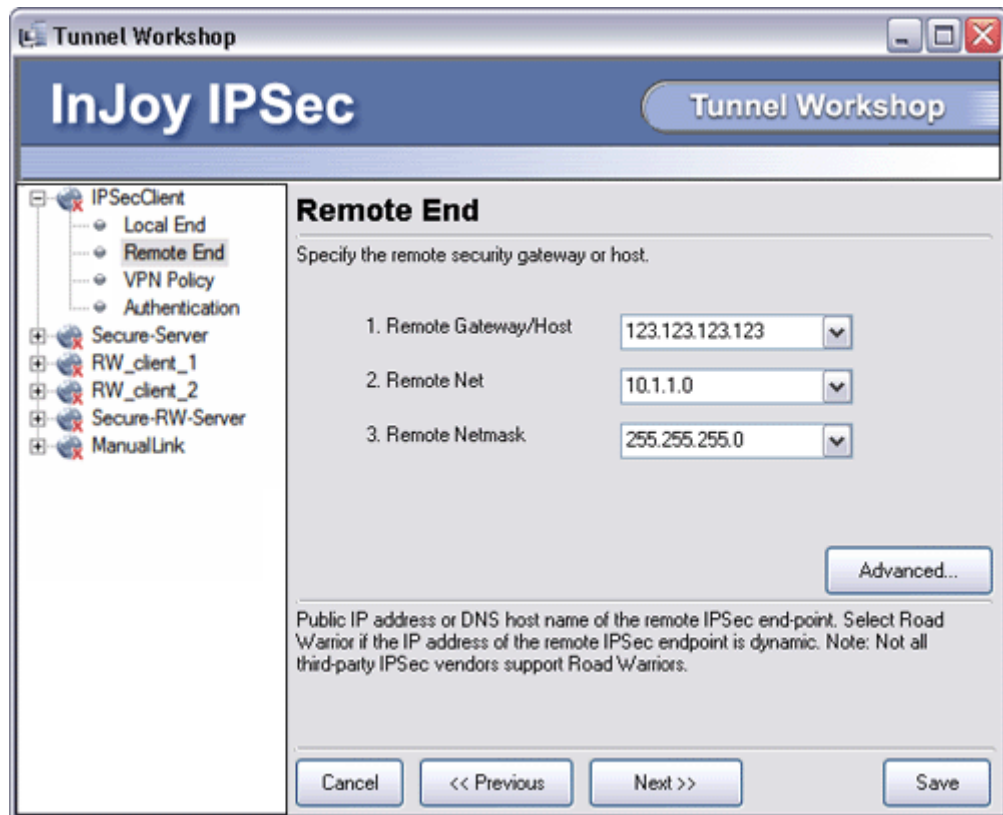
VPN Policy

The VPN Policy dialog allows you configure the IPSec properties of the connection you are creating:

- Whether to use tunnel or transport mode
- The type of host authentication to use
- The type of packet authentication (if any) to use
- The type of encryption (if any) to use

The **Advanced** button in the VPN Policy dialog leads to a sub-dialog that allows you to enable IP compression, NAT traversal features, aggressive mode negotiation, and other special-needs features.
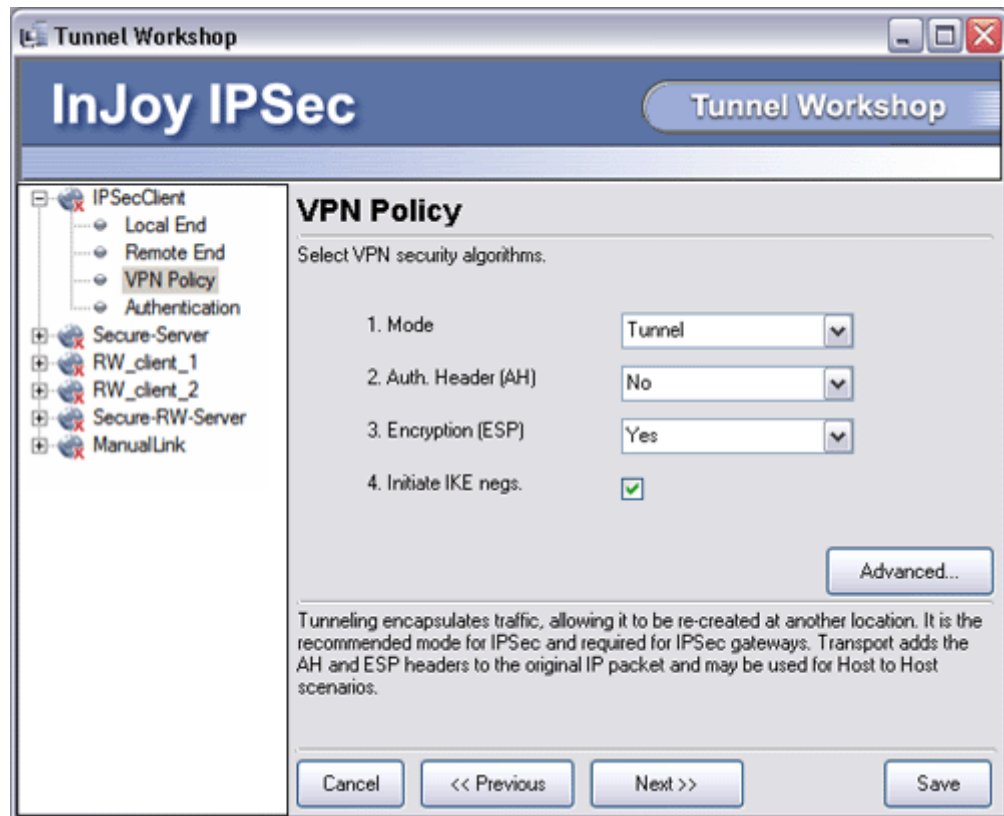
Authentication

The Authentication dialog allows you to configure the host authentication properties for this connection, including:

- The Pre-shared Key required for all other data exchange
- The type of host authentication to use
- A username and password for host authentication

The Advanced button in the Authentication dialog leads to a sub-dialog where you can enter additional information required by some types of authentication—for example, public and private keys for RSA DSS authentication.
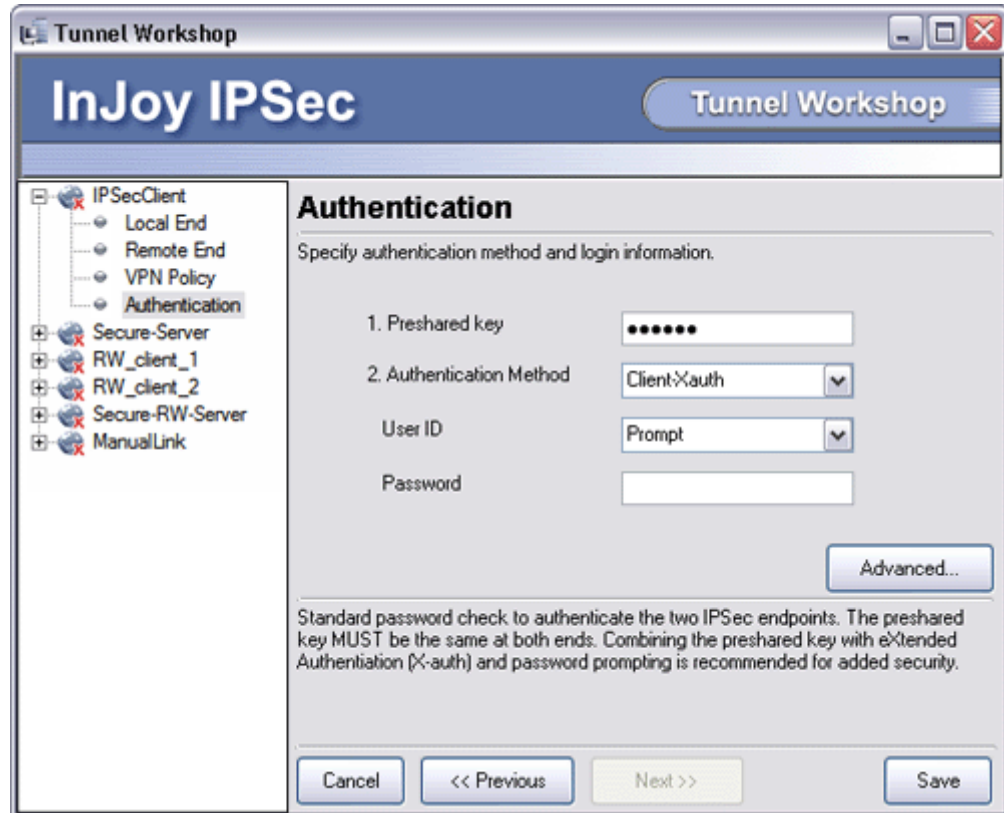
7.2. Editing Existing Security Associations

To edit existing SAs that appears on the left side of the Tunnel Workshop dialog, follow these steps:

1. Click on the small plus (+) sign next to the security association you want to edit. This will expand a dialog tree for the security association in question.

2. In the dialog tree, you will see entries for each of the four dialogs described in the previous section: Local End, Remote End, VPN Policy and Authentication. Click on the name of the dialog you'd like to view.

3. When the dialog appears on the right side of the tunnel workshop, it will already contain the values related to the security association in question. Edit them as desired.

4. Click on the **Save** button to save your changes.

7.3. Sample Security Associations

You will find that a number of sample IPSec security associations are included with the InJoy software. These appear in the Tunnel Workshop the first time you open it.

You can study and modify these sample security associations as needed in order to become familiar with the process of creating and editing security associations in the Tunnel Workshop.

Note: The red 'X' in the icons next to sample SAs, indicates that these samples are disabled. To enable such an SA, right click it and choose enable in the pop-up menu.

8 Using InJoy IPSec

This section is designed to help you to become familiar with the operational details related to InJoy IPSec, including:

- The basic architecture of the InJoy IPSec implementation

- GUI tools for monitoring IPSec

- IPSec logging and tracing

- Functionality limitations related to InJoy IPSec

For real-world examples that use the information presented in this section, please refer to Section 10, "A VPN Case Study."

8.1. Basic Architecture

The InJoy IPSec implementation is divided into a number of software components, each of which carries out certain IPSec-related tasks.

Pluto IKE Server
The Pluto IKE Server (Pluto) is an open-source implementation of the Internet Security Association and Key Management Protocol (ISAKMP). It performs end-point authentication and automatic negotiation for key exchange and other properties related to the VPN policy.

IPSec Module
The IPSec Module (IPSec) performs the actual packet transformations in the VPN. This includes Authentication Header (AH) transformations, which are used for data authentication and integrity checking, and Encapsulating Security Payload (ESP) transformations, which are used for data encryption.

IPSec Interface Module
The IPSec Interface Module (IIDLL) allows the InJoy IPSec Modules to communicate with the open-source Pluto IKE Server in order to coordinate security policy and key exchange.

Authentication Module
The Authentication Module (FXauth) cooperates with the Pluto IKE Server to perform the types of client host authentication described in Section 3.3, "Authentication Methods."

8.2. Monitoring Users and Tunnels
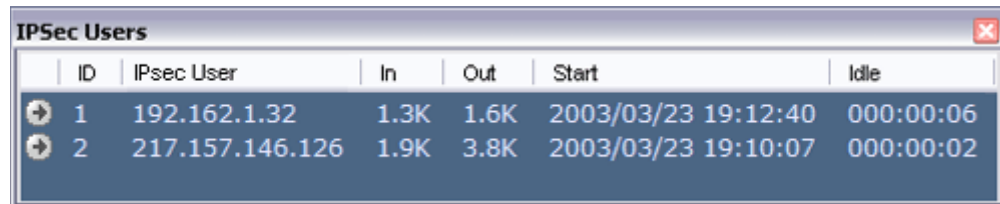
IPSec enables you to monitor the status of your IPSec connections, and internal users VPN utilization.

InJoy Firewall™ IPSec Monitors

When IPSec is enabled, the InJoy Firewall™ GUI offers a number of monitors that you can use to observe the operation of IPSec. A monitor is toggled on and off by selecting it from the **Monitors** sub-menu in the InJoy Firewall™ GUI popup menu.

IPSec Users Monitor
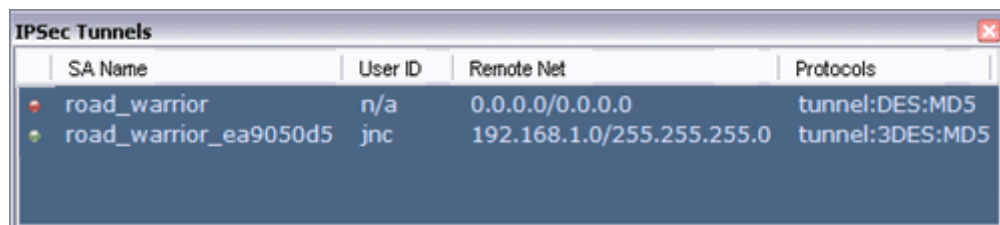
For each internal user (behind the IPSec VPN gateway), the "IPSec Users Monitor" shows the IP address, the volume of traffic which has been received (In) and sent (Out) over the IPSec VPNs.

	ID	IPsec User	In	Out	Start	Idle
	1	192.162.1.32	1.3K	1.6K	2003/03/23 19:12:40	000:00:06
	2	217.157.146.126	1.9K	3.8K	2003/03/23 19:10:07	000:00:02

IPSec Tunnels Monitor

For each active tunnel (SA), the IPSec Tunnels Monitor shows the security association name, the user-id (if any), the local IP address and network, the remote IP address and network, the protocols in use, the time at which the connection was opened and the volume of traffic which has been received and sent.
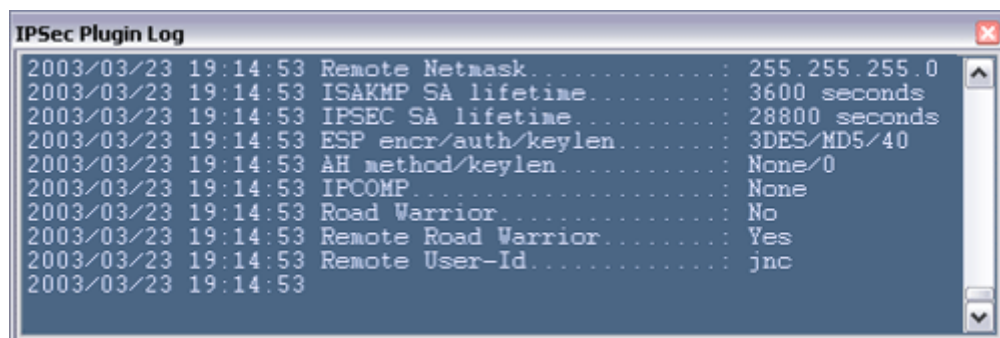
	SA Name	User ID	Remote Net	Protocols
	road_warrior	n/a	0.0.0.0/0.0.0.0	tunnel:DES:MD5
	road_warrior_ea9050d5	jnc	192.168.1.0/255.255.255.0	tunnel:3DES:MD5

IPSec Log Monitors

The IPSec Plugin Log Monitor, the IPSec IKE Log Monitor, and the IPSec Auth. Log Monitor each display entries made to the logs described in Section 8.3, "Logging and Trace Files."

The Plugin Log Monitor watches **logs\ipsec.log**; the IKE Server Log Monitor watches **logs\pluto.log**; the Auth. Log Monitor watches **logs\vpn-auth.log**.

```
IPSec Plugin Log
2003/03/23 19:14:53 Remote Netmask..............: 255.255.255.0
2003/03/23 19:14:53 ISAKMP SA lifetime.........: 3600 seconds
2003/03/23 19:14:53 IPSEC SA lifetime..........: 28800 seconds
2003/03/23 19:14:53 ESP encr/auth/keylen.......: 3DES/MD5/40
2003/03/23 19:14:53 AH method/keylen...........: None/0
2003/03/23 19:14:53 IPCOMP.....................: None
2003/03/23 19:14:53 Road Warrior...............: No
2003/03/23 19:14:53 Remote Road Warrior........: Yes
2003/03/23 19:14:53 Remote User-Id.............: jnc
2003/03/23 19:14:53
```

The IPSec Plugin Log, showing a new SA being installed for a remote user.

8.3. Logging and Trace Files

The activity of IPSec can be monitored by means of plain-text log files.

IPSec is a sophisticated VPN solution, which makes use of a wide variety of technologies to provide an extremely flexible, reliable and transparent security layer. The IPSec logs mirror the complexity and they can be challenging to comprehend at first.

IPSec Logging

The **logs\ipsec.log** file contains messages related to the IPSec engine and the various IPSec-related modules described in detail in Section 8.1, "Basic Architecture."

Each one-line entry in the **logs\ipsec.log** file contains a number of fields:

- Time of event or time of entry.

- Category, one of Warning, Error or Internal Error. When no category is given, the entry is simply a status or informational message.

- Name of related security association.

- Logged event or message.

The **logs\pluto.log** file contains messages related to the Pluto IKE Server and Internet Key Exchange, as well as aspects of authentication which are managed by the IKE Server. For details on the format of this file, please refer to the Pluto IKE Server documentation.

The **logs\vpn-auth.log** file contains messages related to Extended Authentication (Xauth) attempts from remote hosts.

Controlling the Log File Sizes

Log files are not wrapped around and new log lines are appended to the end of the file. The log files are only open while they are actually being updated.

The amount of information logged and the maximum file size can be specified in the settings of the **ipsec\options.cnf** configuration file. As illustrated below:

```
Options         Trace-AH = No,
                Trace-ESP = No,
                Trace-Tunnel = No,
                Trace-Frag = No,
                Trace-Packets = No,
                Dump-Packets = No,
                Log-Level = Info,
                Log-Limit = 20000,                    ← 20MBytes Log Files
                Start-IKE-Server = Daemon,
                Nested-SA-Bundles = No,
```

The default log file size is 10Mbytes.

When log files reach their maximum size, they are deleted.

8.4. Fail-over and Fall-Back

When the IKE Server fails to successfully connect to the remote IPSec end-point within its timeout period (typically between 1-3 minutes), it offers fail-over capability to a second IP address.

If the second IP address is also unreachable, IPSec falls back to the primary IP address.

IPSec will continue to switch between addresses until it manages to successfully connect. Once a successful connection is obtained, IPSec will continue using the successful IP number for as long as it works.

The two IP addresses are specified in the SA configuration as "Remote-IP" and "Remote-IP-2". If Remote-IP-2 is not set to any value, the feature is disabled.

8.5. Transform Order Control

The IKE server allows advanced users to specify the combinations of encryption and authentication algorithms to be proposed during tunnel negotiation with the remote endpoint.

A transform is a combination of a encryption and authentication algorithms that IKE servers offer to each other, during negotiation. A collection of transforms (i.e. all possible encryption / authentication combinations) is called a proposal.

A typical transform may look like this:

- Encryption: AES-256

- Data Authentication: SHA1

- Client Authentication: Pre-shared key

- Diffie-Hellmann Group: 1536 bits (Group 5)

In InJoy IPSec, the order is controlled by means of the Transform-Order attribute in **ipsec.cnf**. This attribute consists of transforms separated by spaces:

```
Transform-Order = "AES-256;SHA1;PK;DH5 AES-192;SHA1;PK;DH5 AES;SHA1;PK;DH5",
```

A transform is composed from 4 values separated by semicolons. Possible values include:

- Encryption: AES-256 (256-bit AES), AES-192 (192-bit AES), AES (128-bit AES), BF (BlowFish), 3DES (Triple DES), DES.

- Data Authentication: MD5, SHA1.

- Client Authentication: PK (Pre-shared keys), RSA (RSA Signatures).

- Diffie-Hellmann Group: DH1 (768 bits), DH2 (1024 bits), DH5 (1536 bits).

8.6. Perfect Forward Secrecy (PFS)

Perfect Forward Secrecy (abbreviated PFS) is an additional method to get more security for the tunnels: the essence of PFS is that no encryption keys should be derived from earlier negotiated encryption keys. By avoiding this, even if one of the keys is compromised, the attacker will be unable to determine the next encryption key that is negotiated.

By default, PFS is off. To turn it on, set the "PFS" attribute in to **Yes** in **ipsec.cnf** or the InJoy Firewall GUI.

8.7. Selectively Bypassing the Tunnel

At times, it is necessary to create exceptions in the way that IPSec unconditionally processes ALL traffic, which is covered by an SA.

As a typical scenario where this feature is of use, is when e.g. a DNS or print server must be available at all times - even when the secure tunnel isn't active.

For example, if the entire 10.*.*.* network is covered by an IPSec tunnel, but certain parts of the 10.*.*:* network isn't to be routed via the secure IPSec tunnel, then this can be achieved by using the attribute **Direct-Nets** in **ipsec.cnf**: just fill it with the networks that you would like to pass unencrypted.

```
Direct-Nets = "10.11.30.* 10.11.31.* 10.11.3.*",
```

Notice the notation: Networks are separated by spaces and wildcards can be used in the network addresses (only the '*' wildcard is supported).

As a security consideration, be sure to use this feature carefully, as the traffic matching the Direct-Nets is sent in the clear.

8.8. Path MTU Discovery

When an IPSec endpoint has large amount of data to transfer to another IPSec endpoint, the data is then split to several datagrams and sent one by one. Afterwards, at the destination IPSec endpoint, the datagrams are collected and assembled into one big datagram. This way, no application is limited in amounts of data to transfer.

However, at times, the datagrams splitted by TCP/IP stack may be too large for intermediary path nodes to transfer. If such a situation happens, the intermediary router bounces the erroneous datagram to originating IPSec endpoint and it splits then big amounts of data into smaller datagrams. The maximum size of the datagram that intermediary router can forward, is stored inside the error message.

The smallest datagram size in the path is called **Path MTU** and the process of identifying the Path MTU is called **Path MTU Discovery**.

Because InJoy IPSec is built on top of TCP/IP and not inside the kernel, InJoy IPSec requires additional setup for Path MTU Discovery to work as intended. By default, Path MTU Discovery is turned on in the **options.cnf**, and its operation is transparent to TCP/IP stack and user.

8.9. Heartbeats and Tunnel Liveliness

It's possible that the situation goes wrong at times and the tunnel stability is less than usual, meaning for the user unstable operation, disconnects and other undesired effects. This situation can be recovered in some time automatically by IKE Server, however, a method exists to recover it faster by using Heartbeats feature of IPSec.

Essentially, heartbeats are simple ICMP pings sent to a remote endpoint over the tunnel. Thus, if it does not respond, it usually means that there is a problem with IKE negotiations, the tunnel itself, or network instability. In this case, InJoy IPSec tries to re-establish the tunnel until it has a good connection.

For this method to work, the remote endpoint must be ping-able, i.e. the ICMP pings must not be firewalled.

For more details about the feature and its setup, refer to **options.cnf** description.

8.10. Limitations

The InJoy IPSec implementation is a powerful, flexible software product designed to protect your network and keep your data private. There are, however, several limitations to be aware of as you use InJoy IPSec:

- Applications which require broadcast or multicast support will not work with IPSec.

- IPSec works only with TCP/IP traffic; other types of network traffic can not be secured using IPSec.

- A maximum of 1000 security associations (tunnels) are supported. More tunnels are possible with specially compiled versions.

- Only one instance of InJoy IPSec can operate on any one computer, regardless of the number of network interfaces.

Part III
Setting up a VPN

9   IPSec Deployment Planning

To configure the most secure and optimal solution possible for your own unique needs, you should plan ahead carefully before deploying your VPN. Follow these steps:

Step 1:	Step 2:	Step 3:	Step 4:
Preliminary planning and decision-making.	Identifying VPN participants (endpoints and networks).	Choosing specific IKE and IPSec policies.	Choosing which IPSec extensions to use.

This section is intended to provide you with background information about IPSec options, along with answers to common IPSec deployment questions. After reading this section, you should be able to make good decisions about the IPSec-related security architecture of your network.

If you are setting up a small-business VPN or a simply need to hook up two PCs securely, the information in this section will be helpful, but it should not be considered required reading. For an easy start, instead refer to section 6, "Using the Quick VPN Wizard".

9.1. The IPSec Planning Workshop

Welcome to the IPSec Planning Workshop. This part of the text is designed to help you ask and answer a number of important questions about your VPN:

- How will it be structured?

- What policies and features will you need and use?

- What problems might you have?

Rather than presenting you with long lists of features, keywords and caveats, the following sections pose questions that you should ask yourself. Then, they provide answers to these questions.

What data should my VPN protect? Who should it include?

As you read in earlier sections, IPSec is the premier solution for the confidentiality, security and reliability needs of your company Intranet.

At the same time, IPSec imposes a measurable amount of overhead on computing and network resources. IPSec can also add complexity to network administration tasks and create interoperability or firewall management

headaches when used with third-party products or large numbers of independent remote users.

For these reasons, the first step in building your VPN is make two simple lists:

- A list of the resources and communications in your business which must be protected, and related networks or hosts

- A list of users which must be given secure access to these resources and communications, and related networks or hosts

Under most circumstances, you will deploy your VPN with respect to these two lists—thereby avoiding, if appropriate, the use of extra processing, bandwidth, and administration resources on hosts where IPSec just isn't needed.

How should my VPN be structured?

Once you decide who and what to protect with IPSec, you must make a number of decisions related to the organization and behavior of your VPN.

Use Peer-to-peer or VPN Client / Server?
Hosts that communicate using IPSec can pass data to each other in two basic ways:

- Indirectly, with each connecting using IPSec to a central IPSec/VPN Server that has network-wide routing responsibilities

- Directly, by opening a dedicated peer-to-peer IPSec tunnel between hosts

Peer-to-peer tunnels are the obvious solution when the number of connecting hosts is small and when having additional hosts in the communication path would provide little or no benefit.

This is the case, for example, when gateway machines need to communicate with one another across the Internet in order to link two corporate networks together. In this scenario, the remote subnets can be clearly defined; this allows the gateway machines on either end to easily determine which packets should be routed through the IPSec connection.

Notice in the figure below, how all 3 VPN gateways are directly connected and traffic can be routed directly.

Head Office
(VPN Gateway)

Internet

Direct Communications

VPN Node   **VPN Node**
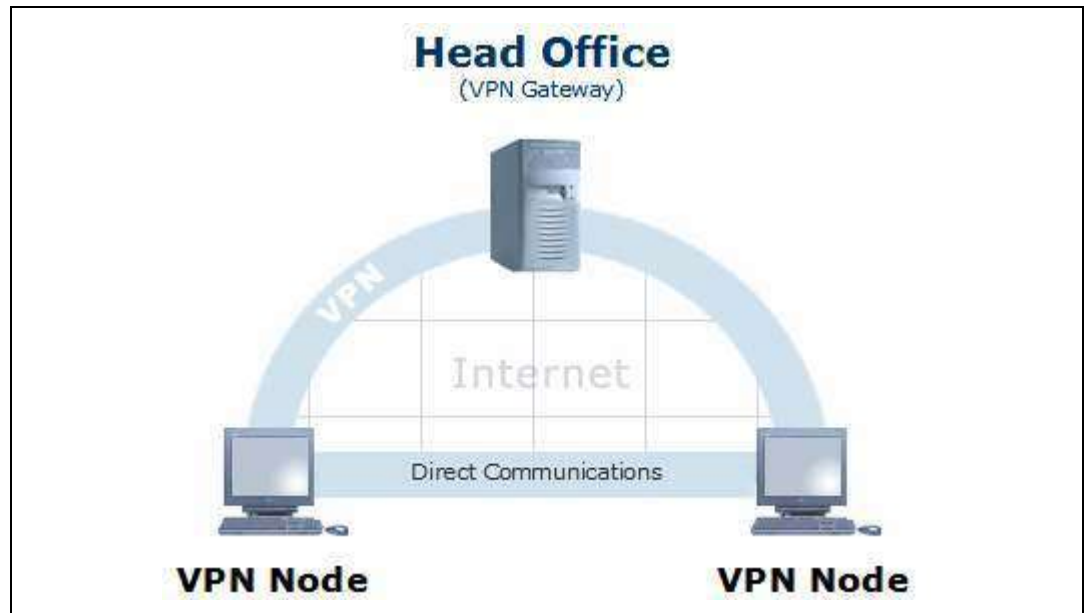
VPN Client / Server tunnels are the obvious solution when remote VPN clients need central authentication, if there are many VPN hosts, and when some remote IPSec hosts use dynamic IP addresses (i.e. IP addresses are only known by the central VPN Gateway, to which the VPN clients login).

To communicate with other hosts, client machines in the private network first connect to the VPN Gateway, then route traffic through it. This allows a large number of SAs —often a mix of hosts, networks, and users with a variety of access policies—to be managed from a single, central location.

Furthermore, rather than needing to know the address of every client or user that requires a secure connection, all a client needs to know is the address of the VPN server responsible for routing secure traffic.

Notice in the figure below, how remote VPN nodes can only communicate through the VPN Gateway.

Head Office
(VPN Gateway)

Internet

NOT CONNECTED

VPN Node   **VPN Node**
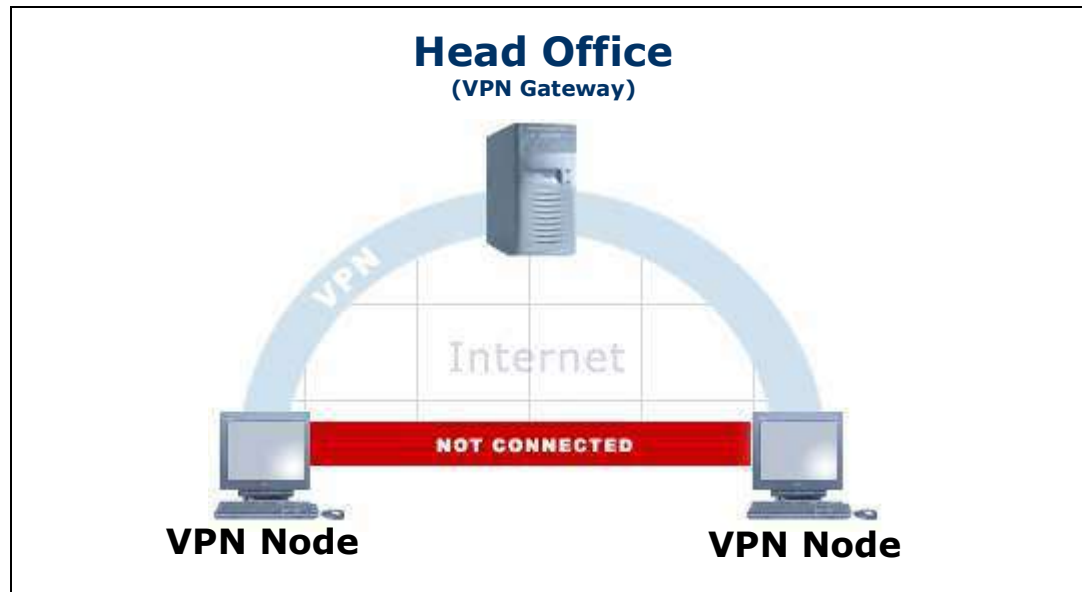
Note: IPSec is a flexible protocol and even if you initially decide to use a VPN Server, then you can easily add peer-to-peer security associations at a later stage – effortlessly mixing the two technologies.

How is IPSec tunnel negotiation handled?
In general, whether your network is organized primarily as a peer-to-peer network or as a centrally served network, any host will be able to open a secure IPSec connection whenever it needs to communicate to another host.

The important exception to this rule is the Road Warrior, or any IPSec host that uses dynamic IP addresses. When dynamic addresses are used, only the dynamic client is able to make an initial connection request.

IKE negotiation is automatically triggered at a number of different times:

- As the host is booted, if you have configured IPSec to remain connected on a permanent basis.

- Whenever tunnel synchronization problems occur.

- When the firewall GUI or command line tools like ipsec.exe or sync.exe are used to restart various InJoy components.

- When a client is assigned a new IP address via DHCP.

- When the remote host IP resolves to a new address (if a DNS name is specified).

Can I avoid routing Internet traffic through my VPN server?
In practice, most network clients will communicate both with your proprietary resources (which should be secured using IPSec) and with the public Internet, for things like Web browsing or streaming media. Communication with the public Internet obviously does not need to be routed through an IPSec server; in fact, doing so can unnecessarily tax computing and network resources.

To eliminate this waste, you can use split tunneling to connect the host to the network—traffic that is destined for other hosts on your internal network will

be routed through your VPN server, while traffic destined for the public
Internet is routed through your standard Internet gateway.

Split Tunnelling

Internet

VPN

VPN end-point　　　　　**VPN end-point**

Internet Host
(Web server)
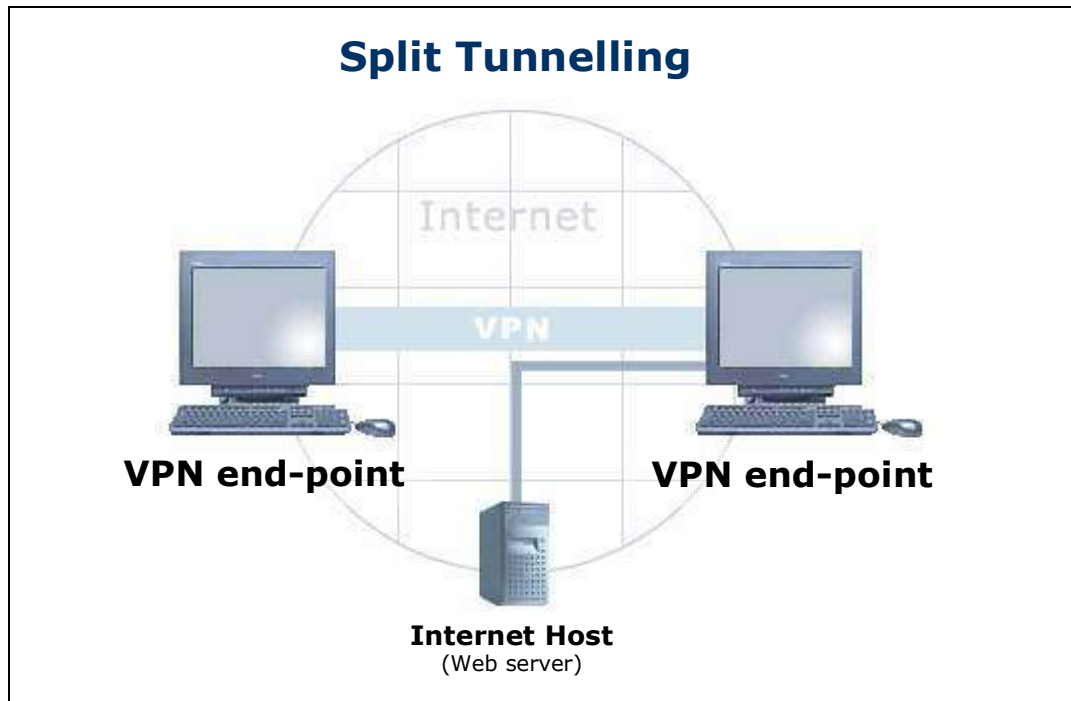
There are cons to using split tunneling as well, however. Clients who route all
network traffic through a VPN server are less susceptible to attack. The VPN
server also provides a forum for logging clients' traffic and connection
requests, a feature which can be very useful in business contexts.

What IPSec security measures should I use?

Most of the authentication and security features in IPSec can be enabled or
disabled at will, with some exceptions. Only you can know your own security
needs, but the following sections give pros and cons for each type of security
or authentication measure, in order to help you to make an informed decision.

What are the pros and cons of Pre-shared Keys?
A Pre-shared Key is simply a word known to both parties when a new
connection is opened. If the connecting host doesn't know the word, it will not
be authenticated. Pre-shared Keys are a relatively insecure form of
authentication. However, because they are so fundamental, they must be
used as a prerequisite for any IPSec connection.

Pre-shared Keys should always be encrypted when used with InJoy IPSec
because unencrypted Pre-shared Keys can be stolen by anyone who can see
them—on a monitor, on a sheet of paper, or anywhere else. Pre-shared keys
are encrypted when saved from the InJoy GUI components, but for
convenience it is also possible to fill in plain-text secrets directly in the IPSec
configuration files.

A significant consideration when relying only on pre-shared keys is the fact
that remote Road Warriors all have to use the same pre-shared key. This is
because they all belong to the same server side SA definition and hence

setups with many Road Warriors should pay extra attention to the choice of the authentication method.

In general, you should use one of the other authentication methods in addition to mandatory Pre-shared Keys to ensure a reasonable level of trust between authenticated hosts.

What are the pros and cons of RSA signatures?
RSA signatures provide one of the most secure forms of host authentication that can be used with IPSec. This form of authentication uses two keys, a public key and a private key, to verify the identity of a host. Data is encrypted with a host's freely available public key; in order to be trusted, the host must then be able to decrypt the data (which can only be done with the host's private key) and return it intact.

RSA signatures also offer another benefit—the ability to uniquely identify a host based on their public key. This allows security associations to be build for hosts whose IP addresses are not known or are subject to change, as is often the case with Road Warriors.

Unfortunately, while RSA signatures are very secure, they are also somewhat difficult to manage, requiring:

- Use of an additional command (rsasigkey.exe) to generate public and private key pairs for each host

- Manual editing of the IKE Server's pluto.secrets file

- A trusted system for distributing public and private keys to hosts before they can be authenticated

What are the pros and cons of X.509 certificates?
X.509 is the most technology advanced and safe private-public key-pair authentication and encryption model. It is widely available in most IPSec solutions and by using signed certificates rather than just the key-pairs themselves, it represents a significant improvement over RSA signatures. Further, in addition to RSA signatures, X.509 certificates can be easily revoked on the certificate revocation list, thus making the authentication server reject a user with that certificate.

X.509 certificates make it easier for administrators to uniquely identify users, because an ID and other relevant information about the user are embedded in every certificate.

X.509 certificates can be managed more flexibly than RSA signatures, since they reside in individual files and their generation/management can be fully automated with certificate management software. The flexibility comes at a cost though. X.509 in medium sized installations might introduce more overhead and administration than the added security might justify.

What are the pros and cons of extended authentication?
Extended authentication employs a user ID and password to authenticate each new IPSec connection. Because it requires two identifying pieces of information, rather than just one, extended authentication is somewhat more secure than Pre-shared Key authentication.

You can also configure extended authentication to prompt for username and password information when new VPN connections are negotiated; this reduces the possibility that a stolen or lost laptop could compromise your VPN. In addition, X-authentication provides the additional benefit of login audit logs and as it integrates well with the IPSec configuration-mode protocol, it also allows the central VPN Gateway to assign internal IP addresses to remote VPN clients. These are features that help ease the day-to-day management of the VPN, as well as administration of the corporate network services.

Overall, Extended Authentication provides an excellent balance between high security authentication and easy manageability. For the same reason it is recommended for the common VPNs that have strict, but not very high-grade authentication needs.

What are the pros and cons of data encryption?
Data encryption provides security beyond mere host authentication. The Internet is a packet-switched network; traffic between two hosts often passes invisibly through any number of other hosts on its way. Even when you know that a remote host can be trusted, you can rarely be sure that every host between you and that remote host can be trusted as well. Data encryption solves this potential security problem by making the data being sent unreadable to everyone but the sender and receiver.

IPSec supports several types of encryption, refer to Section 3.2, "Encryption Methods" for details.

Can I use these together for maximum trust and security?
In short, yes. When you need maximum security and a high level of trust between hosts, you can pick freely among the authentication methods and the encryption standards.

For authentication, these combinations are supported:

- Pre-shared Keys (by itself)

- RSA signatures (by itself)

- X.509 certificates (by itself)

- Pre-shared Keys + Extended Authentication

- RSA signatures + Extended Authentication

- X.509 certificates + Extended Authentication

Any of the available encryption standards (e.g. 3DES) can then be used in conjunction with the chosen authentication method.

How can I evaluate and minimize risks?

People often think about IPSec in terms of disabling—preventing untrusted users from interfering with your communication. At its core, however, IPSec is essentially an enabling technology—it allows you to bring remote users into your private network.

What precautions can I take against problem VPN users?

When you give IPSec tunnel access to a remote user, it is as though you have given them access to a workstation in your LAN. You should therefore apply any security checks or considerations to IPSec connections that you would to users or employees who work onsite.

Because the InJoy Firewall™ includes a number of rules designed specifically to interoperate with InJoy IPSec, you can and should use the InJoy Firewall™ to control access to your VPN server or other hosts in your VPN.

Should I limit access to my VPN?

You should avoid giving anyone access to your internal network who does not absolutely need to access it; this includes users who connect via IPSec. IPSec excels at preventing attacks and data theft from unknown parties; IPSec can not, however, protect you from users to whom you've explicitly given access.

What interoperability issues might I encounter?

IPSec is a standards-based solution. Because of this, many different versions of IPSec produced by a variety of software makers are in widespread use today. InJoy IPSec is designed to be interoperable with a wide range of IPSec implementations, but compatibility issues can still arise.

Will particular features cause interoperability problems?

Common or mainstream IPSec features are likely to be interoperable across many IPSec implementations, including InJoy IPSec. However, less common or less often used features may need to be disabled in order to use InJoy IPSec with less capable IPSec implementations. Refer to the following table for details on which features are likely to cause interoperability issues.

Common (High interoperability)	Less Common (Low interoperability)
Pre-shared Key authentication	Extended authentication (Xauth)
RSA signature authentication	X.509 public key exchange
Main mode IKE negotiation	Aggressive mode IKE negotiation
Road Warrior (dynamic IP)	DES encryption (deemed insecure)
3DES encryption	IP compression
Manual keying (no IKE Server)	NAT traversal
MD5, SHA Authentication Header	AES, Blowfish encryption

How can I access all of InJoy's features across platforms?

Because the InJoy software represents a truly multiplatform solution, including versions for OS/2, Windows 2003/2000/XP, Linux and FreeBSD, you can use InJoy IPSec to access all of these features across multiple platforms, bridging legacy and future systems under a single administrative and compatibility umbrella.

What technical obstacles might I encounter?

IPSec is designed to be transparent to you as you use it. As with any complex technologies, however, there are some technical difficulties that you should be aware of.

Is Network Address Translation a problem for IPSec?
Because IPSec encapsulates the very packet header information that Network Address Translation modifies, the use of NAT with IPSec is often problematic. InJoy IPSec includes IPSec NAT traversal features designed specifically to address this issue.

In cases where NAT traversal presents deployment or other problems, the NAT difficulty in some situations can be overcome by sending all traffic on ports 50, 51 and 500 directly to the IPSec endpoint (a task of the Firewall administrator).

For details on these two techniques, please refer to Section 13, "Using IPSec behind NAT."

Can I implement both IPSec and firewall security?
Firewall security presents similar problems to IPSec users, for similar reasons; firewalls are often not able to use the encapsulated headers found in IPSec communication.

Because the InJoy Firewall™ includes rules specifically designed to act on InJoy IPSec traffic and also process traffic after the IPSec layer has de-encrypted traffic, the InJoy Firewall™ doesn't suffer from this limitation. Because of this, we recommend using the InJoy Firewall™ with InJoy IPSec products.

9.2. Identifying Your IPSec Endpoints

Before you deploy your VPN, you must identify your IPSec endpoints (hosts) and make decisions about the ways that they will be treated.

IP Numbers that Impact IPSec

The IP numbers that IPSec uses to identify endpoints and hosts during configuration are the real-world IP addresses of hosts as they are known on the public Internet.

When constructing a VPN, use hosts with static IP addresses whenever possible. Static IP addresses provide an intrinsic layer of security because they are only susceptible to complex man-in-the-middle attacks that require the ability to completely compromise a host.

By avoiding the additional details related to Road Warriors or dynamic IP addresses, static IP addresses can also simplify administrative tasks on your VPN.

Issues Related to Road Warriors (Dynamic IP Addresses)

The use of dynamic IP addresses with IPSec does moderately impact security, because in most cases dynamic IP addresses are owned by ISPs or other organizations not directly connected with your own; as a result, at any given time these addresses may be assigned to your employee, or they may be assigned to completely unrelated, untrusted parties.

In fact, precisely identifying Road Warriors at connect time can be a challenge. For this reason, extra monitoring and logging of IPSec hosts connected through dynamic IP addresses is recommended.

Road Warriors might also at times negotiate connection types through the IKE Server that other hosts don't commonly use, such as IP compression. Since many of the connection options handled by IKE negotiation can negatively impact processor or network resources or security, careful configuration for Road Warriors is important.

Road Warriors often connect through hosts that do not belong directly to your network. A connected IP address that does not belong to any of your IP address ranges can pose an administrative or security problem. For this reason, InJoy IPSec allows the use of Inner-IP addresses—virtual addresses within the address range of your network. These are assigned to dynamic IP hosts and can then be used instead of the hosts' real-world IP addresses.

For additional details on Road Warriors, please refer to Section 11, "Using Road Warrior Support." For additional details on using Inner-IP addresses, please refer to Section 12, "Using Inner-IP Support."

9.3. Defining Your IKE Negotiation Policies

The IKE Server is the heart and soul of the IPSec standard, implementing the protocol layer for all of the following features:

- Host authentication in order to establish a VPN connection

- Negotiation of per-packet authentication properties

- Negotiation of per-packet encryption properties

- Type of per-packet authentication keying and amount of time between key exchange/refresh

Because the IKE Server is so central to the functioning of your IPSec VPN and the level of security it implements, you should take time to carefully plan the IKE Server negotiation choices you'll make.

IKE Server Negotiation Mode

Most PC-based or other computing-platform-based IPSec implementations support both Main Mode and Aggressive Mode negotiation. Main Mode negotiation is more secure and imposes a minor (if any) throughput penalty under most circumstances and should therefore be used whenever possible.

Many network appliance platforms support only Aggressive Mode negotiation. If you will be using devices of this type with InJoy IPSec, you may need to opt for Aggressive Mode instead.

Note also that in noisy or low-grade physical networks environments, excessive IPSec connection re-establishment or renegotiation can occur. Under these circumstances, Aggressive Mode may provide measurably better performance.

IKE Server Authentication Type

By default, the InJoy IPSec will suggest Pre-shared Keys to authenticate connecting hosts. Now is the time to also decide one or both of the other options as well:

- RSA signature authentication or X.509 certificates if you are willing to manage a public/private key (certificates) infrastructure for your networks.

- Extended Authentication (Xauth) if you would like to authenticate and associate connections with particular usernames and passwords (recommended).

IKE Server Encapsulation Mode

If you need a secure gateway-to-gateway connection across the public Internet, need to support Road Warriors, or have a number of disparate IP address ranges assigned to you, tunnel mode is your likeliest encapsulation choice (recommended).

If you need to minimize network and processing overhead, have a large network with a unified assigned address range, and do not need gateway-to-gateway, through-NAT connections, or extra security and integration for Road Warriors, you may wish to consider transport mode instead.

IKE Server Keying Policies

Under almost all circumstances, you should opt to let the IKE Server manage a regular, automatic key exchange for you. This ensures the integrity of the packet-level authentication process and occurs transparently, causing no administrative overhead.

If for some reason you are unable to use an IKE Server with IPSec on one of your connection endpoints, you may need to enable manual keying, which will allow you to use the same pre-shared packet-level authentication key indefinitely—at the expense of security.

9.4. Defining Your Encryption and Hashing Policies

In nearly every instance, you'll want to protect the actual contents of the network packets (data) traveling through your IPSec connections. IPSec uses two tools to protect your data in transit:

- Packet-level encryption (using DES, 3DES, AES, etc ciphers)

- Packet-level authentication (using MD5 or SHA hashes)

Choosing an Encapsulating Security Payload (ESP)

After reading through earlier sections, you're likely already familiar with the Data Encryption Standard (DES), 3DES, AES, and Blowfish ciphers. It's now time for you to decide which you'll use, and for which machines. Use DES only when:

- Minimal security is acceptable

- Traffic encrypted this way will be on an internal network, behind a firewall

- You have unavoidable limits on processing power versus the amount of bandwidth available to your

In nearly all other instances, you should use AES, Blowfish or 3DES for packet-level encryption because of the additional security it offers. AES is the cipher used by the U.S. government, but it's not as widely available as 3DES.

In the rare instances in which you need only packet-level authentication and integrity checking, but not privacy, you may also opt to use Null-ESP encryption—which imposes no processing overhead whatsoever, but also offers no data confidentiality at all.

Choosing an Authentication Header (AH)

IPSec offers two hash types for guaranteeing the integrity of packets which arrive at the receiving end of an IPSec connection: MD5 and SHA.

The MD5 hash algorithm is somewhat faster and is more familiar to many users, but offers slightly less protection against undetected anomalies. The SHA hash algorithm is a little bit slower than MD5, but provides slightly better protection against undetected anomalies.

Some IPSec implementations may support only MD5 or only SHA hashing; if you must use InJoy IPSec in conjunction with other IPSec implementations on your endpoints or hosts, you should consider their capabilities as well when choosing.

In practice, the Authentication Header is often turned off when deploying modern IPSec networks. This is due to its incompatibility with NAT and because the Encapsulating Security Payload includes its own packet hash.

9.5. Using IPSec Extensions

In your final planning step, you should take a moment to consider the extensions offered by InJoy IPSec and to decide which of these you plan to use.

- Plan to offer NAT Traversal if you need to provide access to IPSec connections through a firewall/router.

- Plan to offer IP Compression if you have the available processing resources and you will be accepting connections from Road Warriors, who might be forced to use dial-up lines.

- Plan to offer Inner-IP if you need to unify a network of disparate or dynamic IP addresses under a single private sub-network's umbrella.

Keep Interoperability in Mind

As you take these final planning steps, you should also take time to ensure that you are aware of the features and limitations of the IPSec implementations which will be used at each of your endpoints or on each of your hosts (including Road Warriors).

Because features such as NAT Traversal, IP Compression and Inner-IP are all extensions to the IPSec standard, you may find uneven support among the hosts on your VPN, especially if many of them are using non-InJoy products.

In some cases, you may even need to disable these features if they create problems during negotiation phases with IKE Servers that don't support them.

10 A VPN Case Study

In this section you'll read about a fictional company called Softdev.com. Softdev.com has decided to build and use a VPN, and you'll be able to observe as they step through the process of creating it.

Step 1:	Step 2:	Step 3:	Step 4:
Initial VPN planning and considerations.	Head Office VPN Server configuration.	VPN Client configuration.	Verify VPN operation.

10.1. Softdev.Com: VPN Planning

Softdev.com is a small-but-growing software development company who needs to maintain business communication over the public Internet. In particular, Softdev.com needs to be able to privately, safely communicate with:

- A remote partner company that has its own internal network and an IPSec-enabled gateway

- Two remote software engineers—employees who need to be able to access the company network—running Linux

All of the hosts involved are already installed and can already communicate with one another across the Internet, and all of them are using the InJoy Firewall™ to protect themselves from unwanted network traffic.

Road Warrior
(Dynamic IP)
Inner-ip: 10.1.4.104

Softdev.com

Partner Office

10.1.4.10

Softdev.com

192.168.1.10

12.154.175.1

10.1.4.11

192.168.1.11

Road Warrior
(Dynamic IP)
Inner-IP: 10.1.4.105
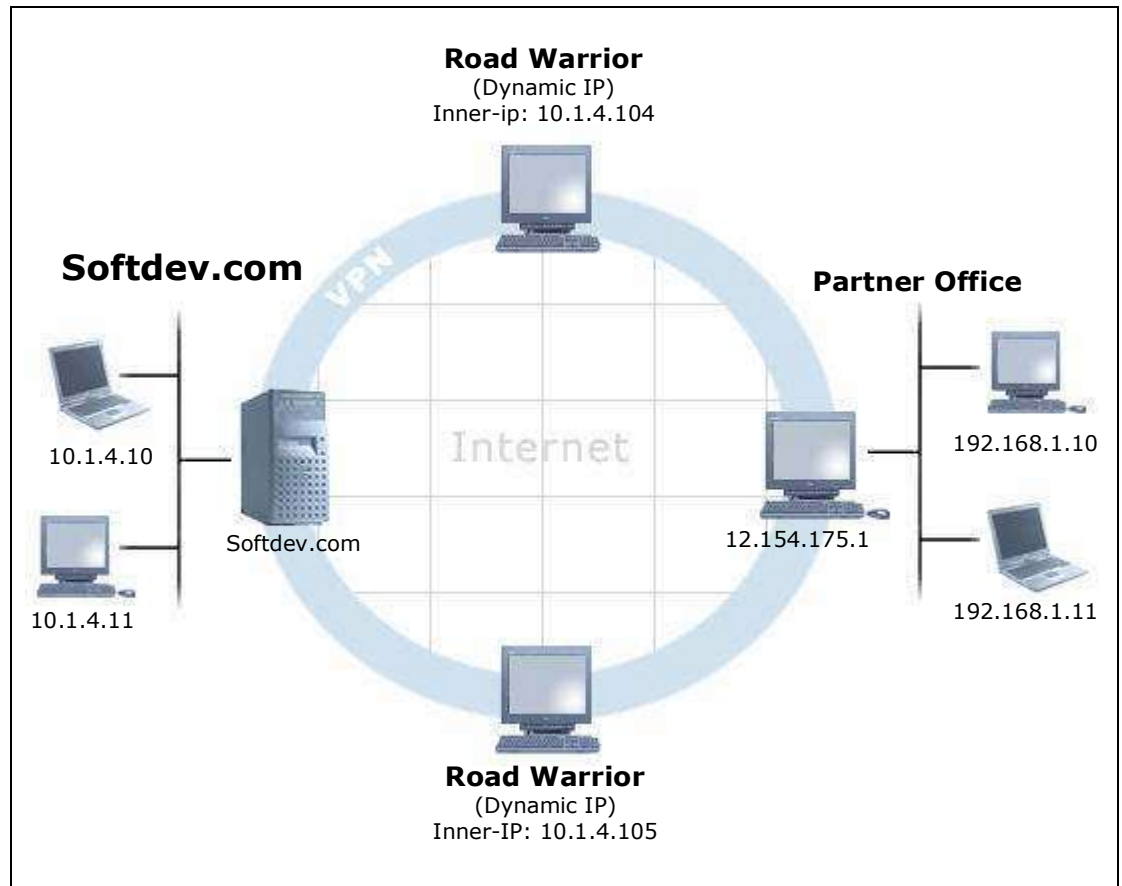
VPN Policy Decisions

Because Softdev.com is a small company, yet still needs maximum confidentiality and reasonably strong authentication, it decides on the following set of IPSec policies:

	Partner	Employees
Authentication	Pre-shared Key + Xauth	Pre-shared Key + Xauth
Encapsulation	Tunnel	Tunnel
Inner-IP	No	Yes
Encryption	3DES	3DES
IP-Comp	No	Yes
Negotiation	Main Mode	Main Mode

To easily monitor VPN use on a per user basis, Softdev.com opts for Extended Authentication (Xauth) for all remote hosts. All connections will be tunneled connections; the remote employees will be assigned an Inner-IP and will use IP compression as well.

10.2. Head Office VPN Server Configuration

Softdev.com begins by configuring its VPN server. The network administrator starts the Tunnel Workshop and begins to configure the machine for use as a VPN server. To do this, he opens the SA Setup dialog. (Refer to Section 5, "Configuration" for an overview of the Tunnel Workshop.)

Partner Company SA

The network administrator steps through each dialog, creating a security association for the partner company's gateway. After completing each dialog, he clicks **Next** to continue to the next one.

Softdev.com's network administrator begins by entering an easy-to-remember name and a description to the security association he's about to create.
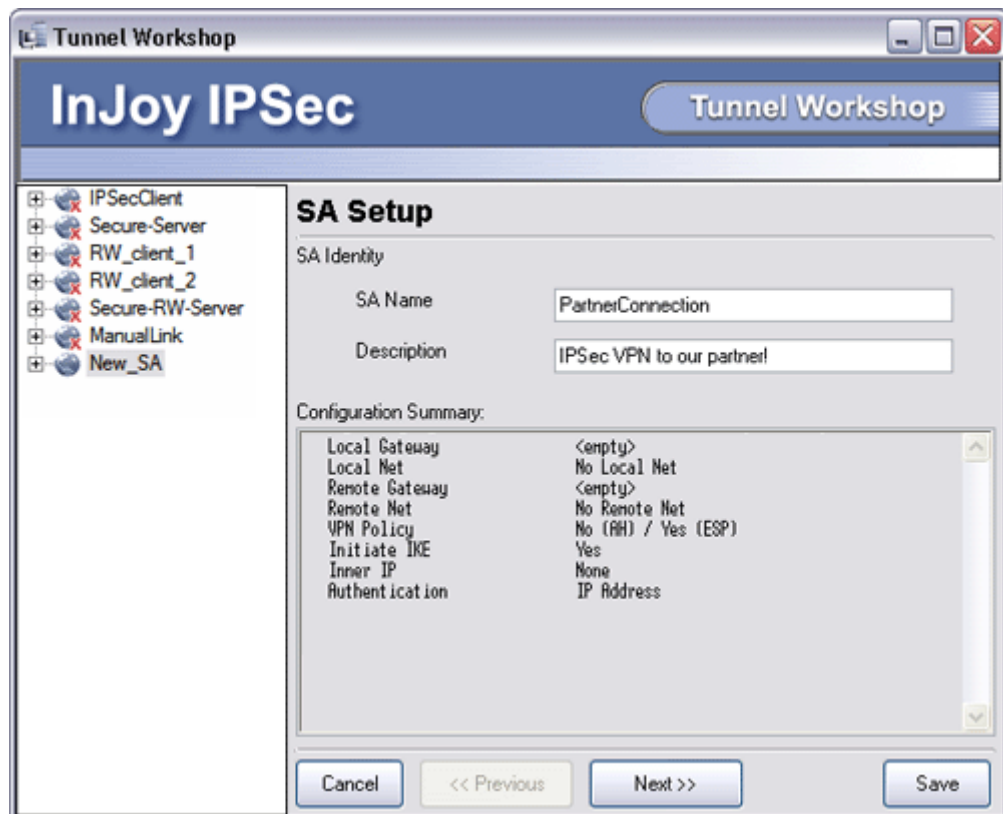
Having entered the SA Name and the Description, he clicks **Next** to proceed to the Local End configuration.
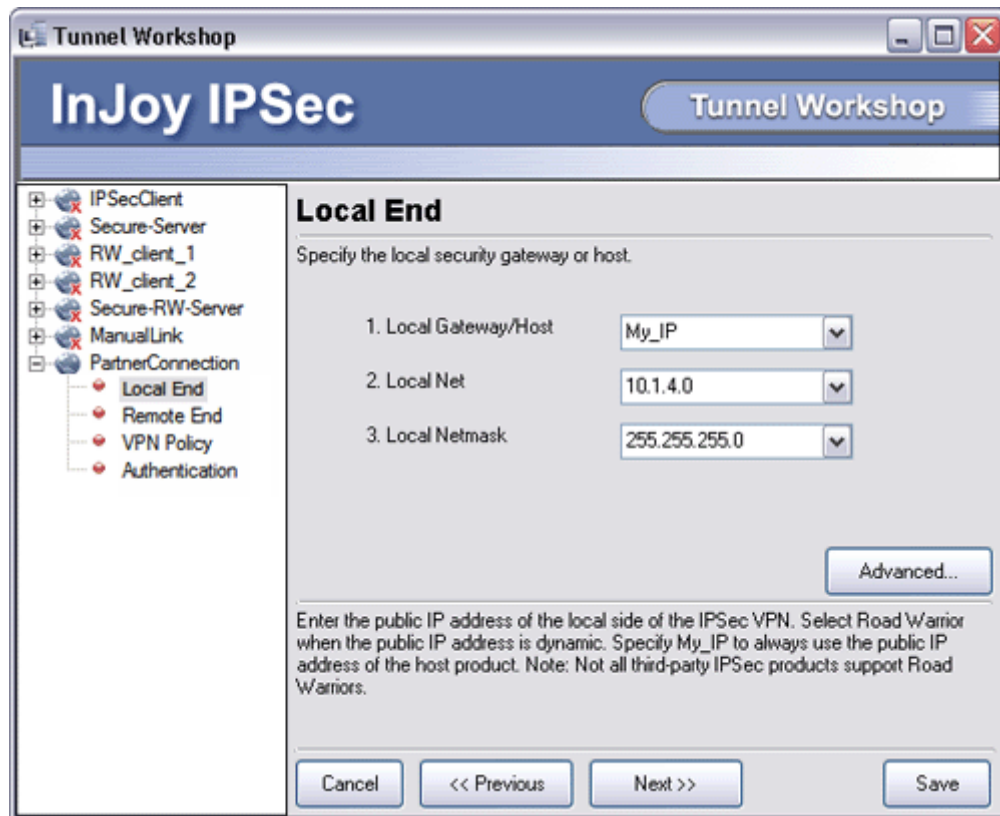
In the Local End dialog, the network administrator selects "My_IP" as the local host, 10.1.4.0 as the local network, and 255.255.255.0 as the local netmask.
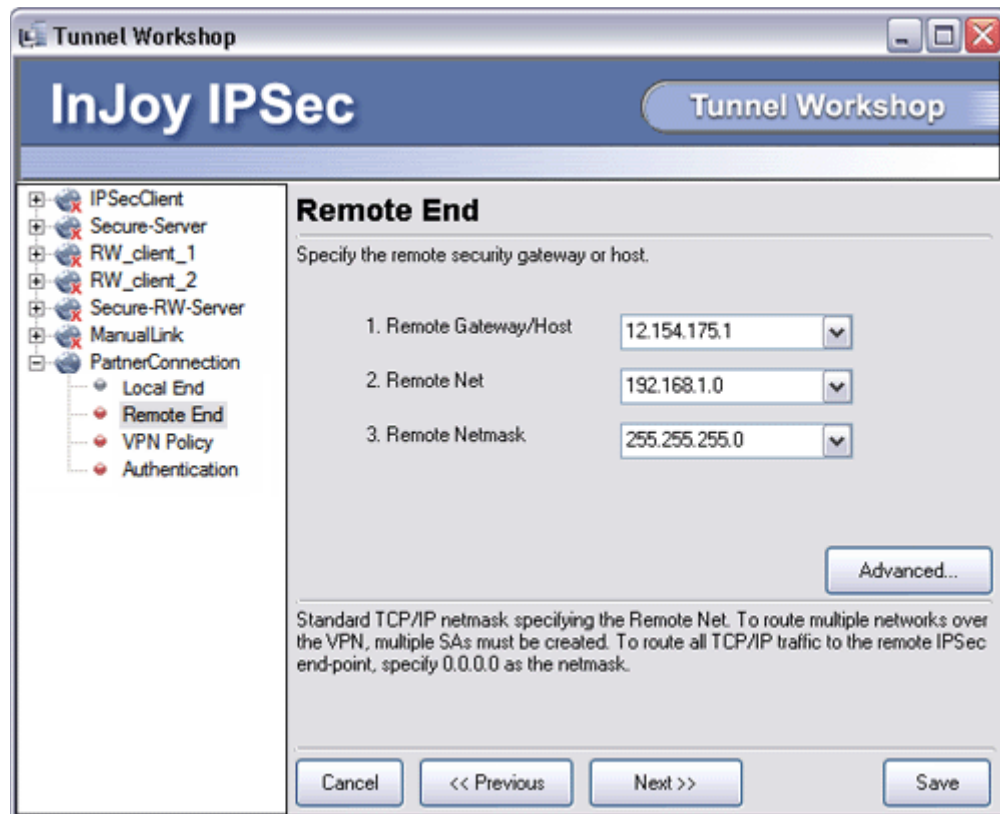
In the Remote End dialog, the network administrator enters the host address of the partner company's gateway, 12.154.175.1, as the remote host. The remote net and remote netmask are then set to 192.168.1.0 and 255.255.255.0, accordingly.
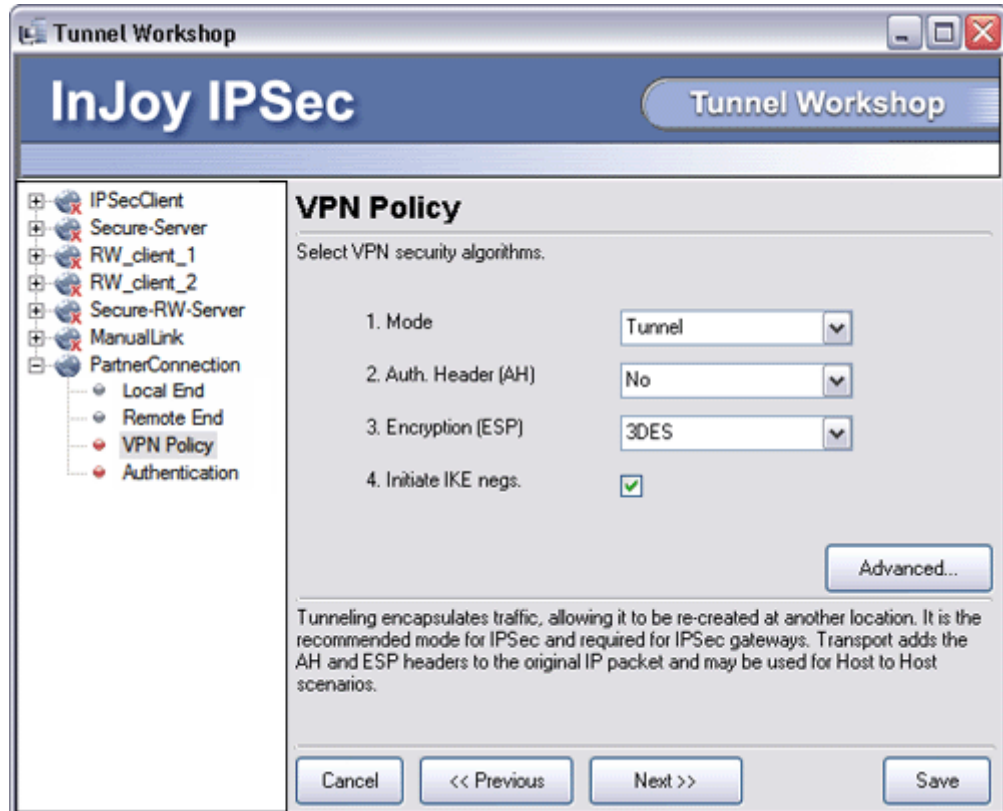
In the VPN Policy dialog, he configures for tunnel mode, without authenticated headers, using the more secure 3DES cipher and he selects this end-point can initiate IKE-negotiations.
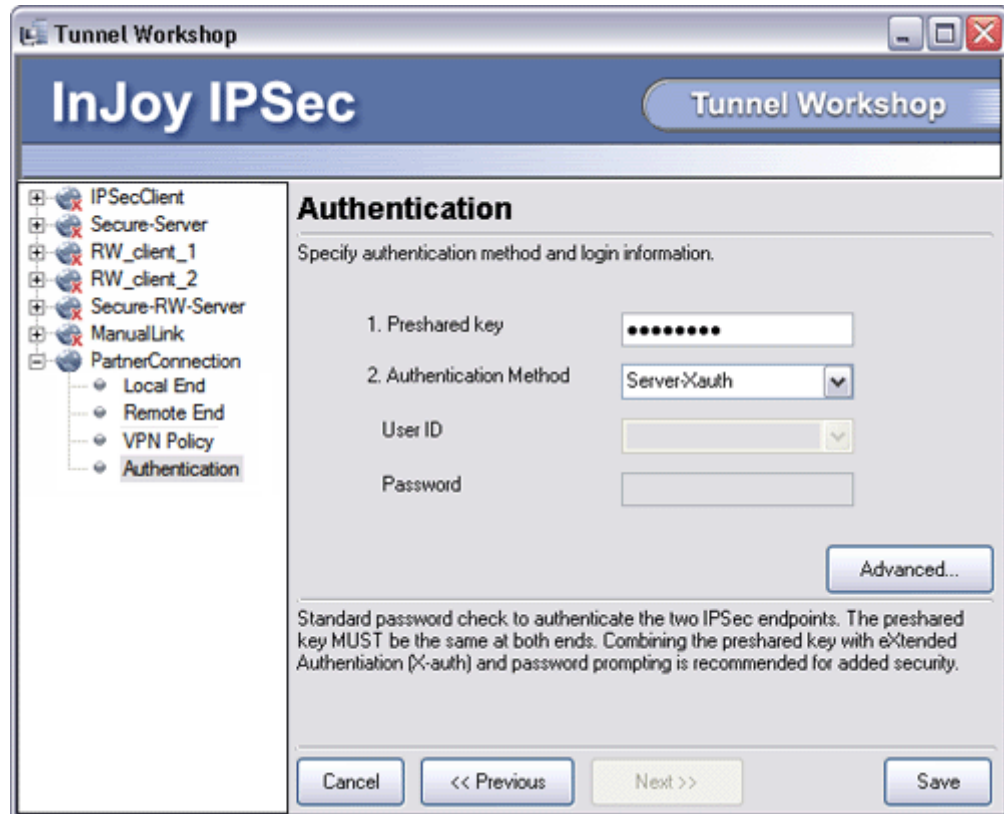
Finally, in the Authentication dialog, the network administrator enters "P4rtnerP4ssWD" as the Pre-shared Key; the partner company's gateway must also know this word in order to connect. He also enables Xauth (server side) as an additional authentication method.

Creating a User Account for the Partner

Sofdev.com's network administrator has now finished creating the SA that applies to traffic to and from the partner company's IPSec VPN gateway.

Because Xauth was selected as an authentication method on the corporate VPN Server, the network administrator proceeds to add the username "SoftdevPartner" and the password "2ndP4ssWD" to the **User Administration** IPSec dialog:
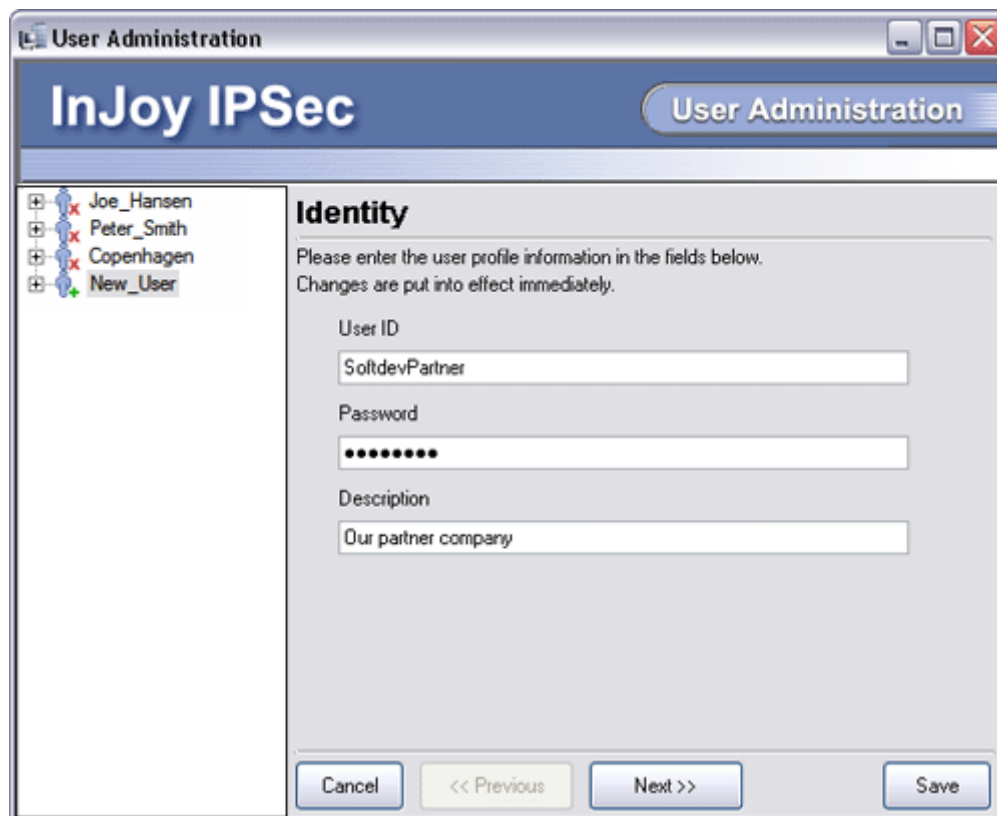
By entering and saving these values, the Softdev network administrator creates an Xauth login for the partner company without assigning it an Inner-IP address. The security association for the partner company's gateway is now ready to be used on the VPN server.

To limit the number of screen-shots, additional user accounts created throughout this chapter will be shown only as sections from the respective configuration file: **ipsec\vpn-auth.cnf**.

Configuration File Details
The Tunnel Workshop configuration steps described above resulted in the following new section in the SA configuration file: **ipsec\ipsec.cnf**:

```
PartnerConnection        Description = "IPSec VPN to our partner!"
                         Local-IP = "My_IP",
                         Local-Net = "10.1.4.0",
                         Local-Mask = "255.255.255.0",
                         Remote-IP = "12.154.175.1",
                         Remote-Net = "192.168.1.0",
                         Remote-Mask = "255.255.255.0",
                         ESP = 3DES,
                         Auth-Type = Server-Xauth,
                         Preshared-Secret = "-50488a009e099a34d42fab0394"
```

Note that the Pre-shared Key doesn't appear as it was entered because it has been encrypted. Also note that only values that differ from the default values found in **template\ipsec.cnf** are actually stored in this file.

The User account information is stored in the file **ipsec\vpn-auth.cnf** and the user account entered in the above section resulted in this new section:

```
SoftdevPartner             Password = "-8po3AdlfmFUUArOXFprNz0"
                           Description = "Our partner company"
```

Employees Security Association

After completing the SA for the partner company, the Softdev network administrator begins work on the SA for the remote employees, following the same steps he followed for the partner company's SA.
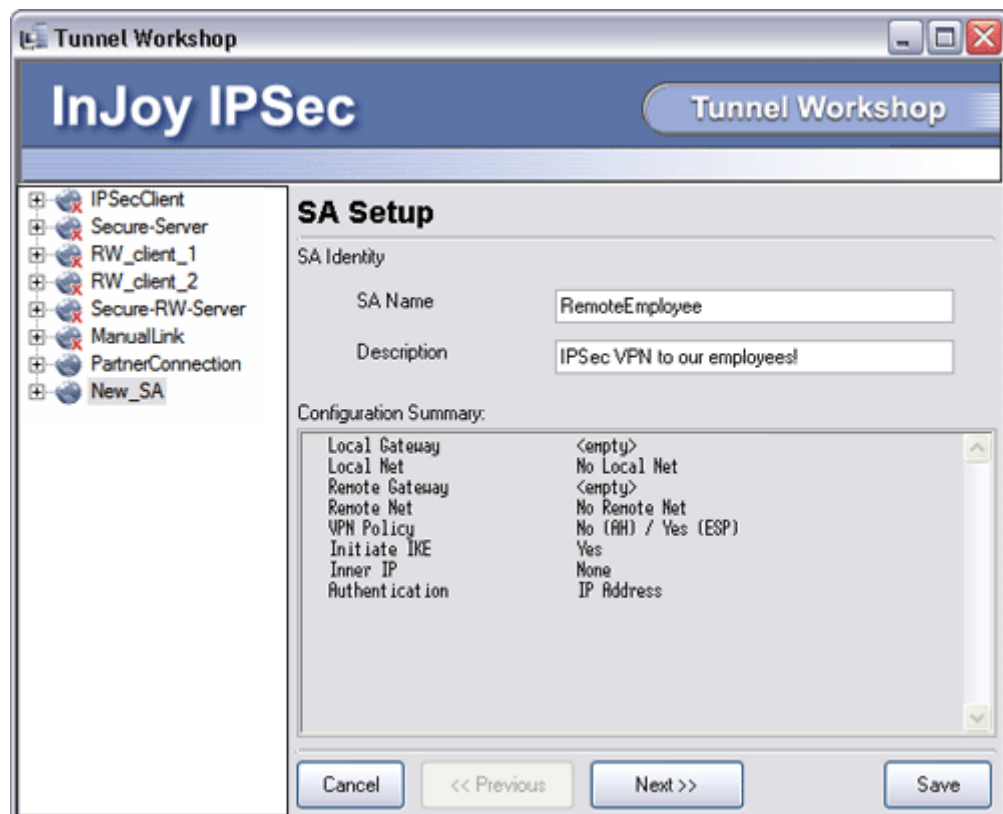
Because the administrator completes the "local end" dialog using precisely the same information he used when creating the partner company SA, the screen shot is omitted here.

In the Remote End dialog, things have changed.
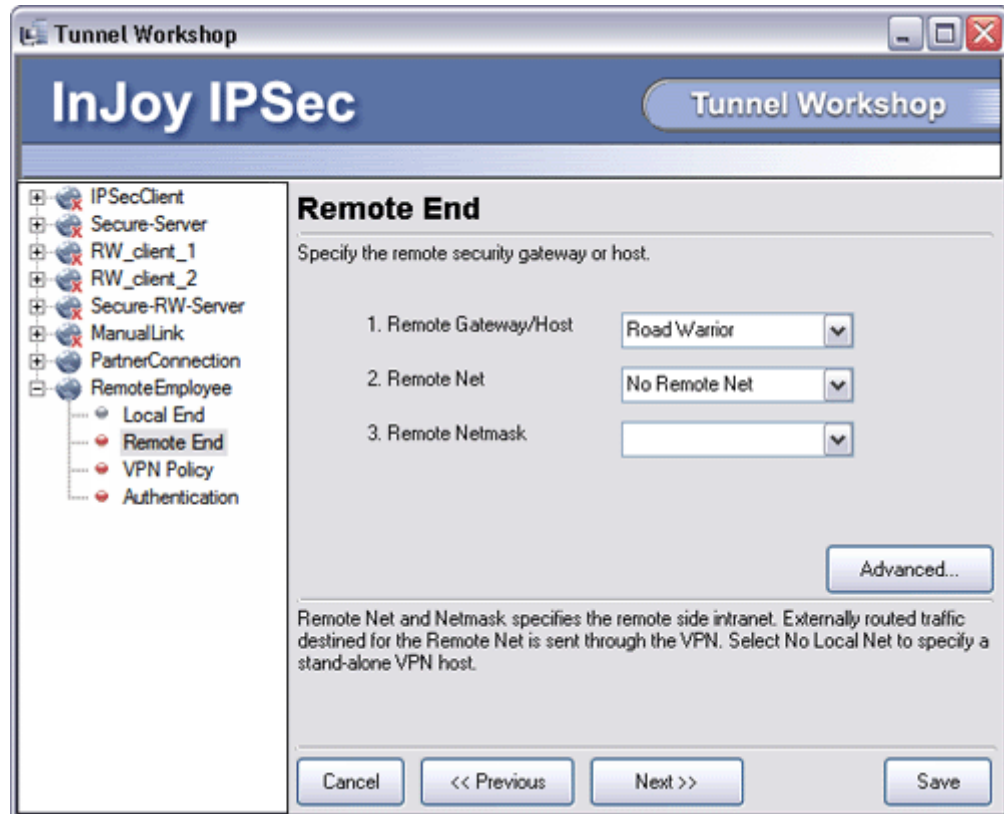
The remote employees are Road Warriors—they are dial-up ISP users and their single-host IP addresses are likely to change on a regular basis. Because of this, Softdev.com's network administrator selects the "Road Warrior" and "No Remote Net".
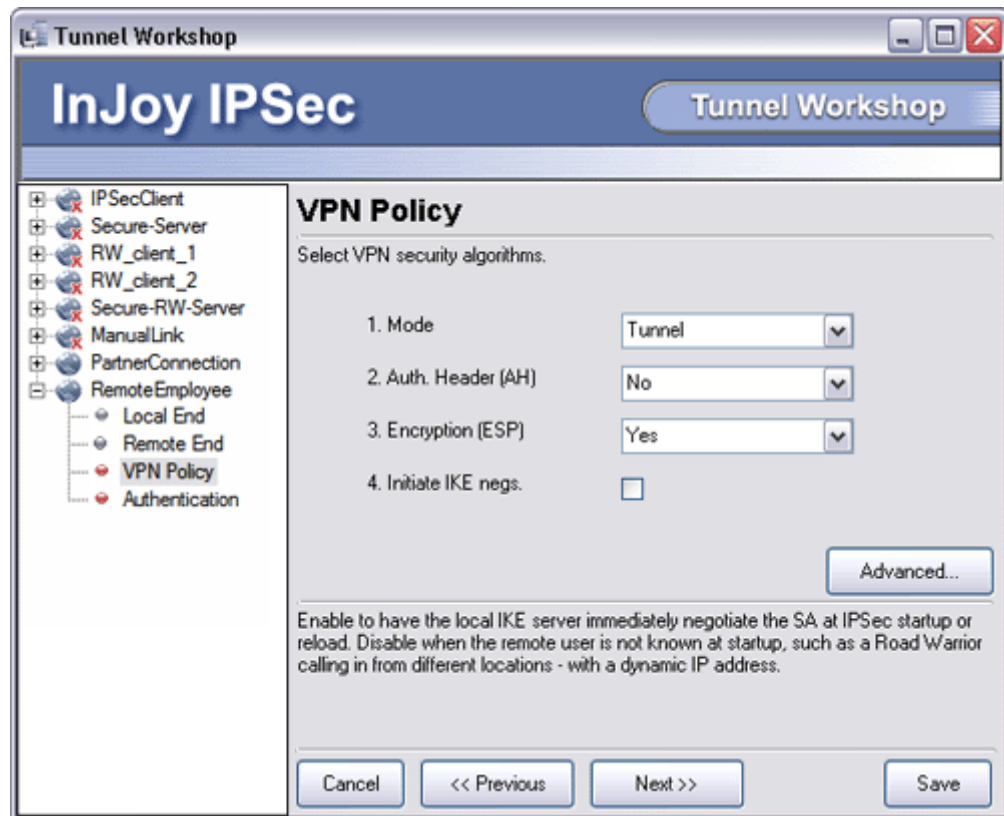
The VPN policy for the employees is similar to the policy used for the partner company connection, with one exception: because Road Warriors IP addresses change and their connections go up and down, the VPN server can't initiate IKE negotiation. This feature is disabled (unchecked) by the network administrator.

The network administrator now clicks on the **Advanced** button in the VPN Policy dialog, in order to access advanced VPN policy options.
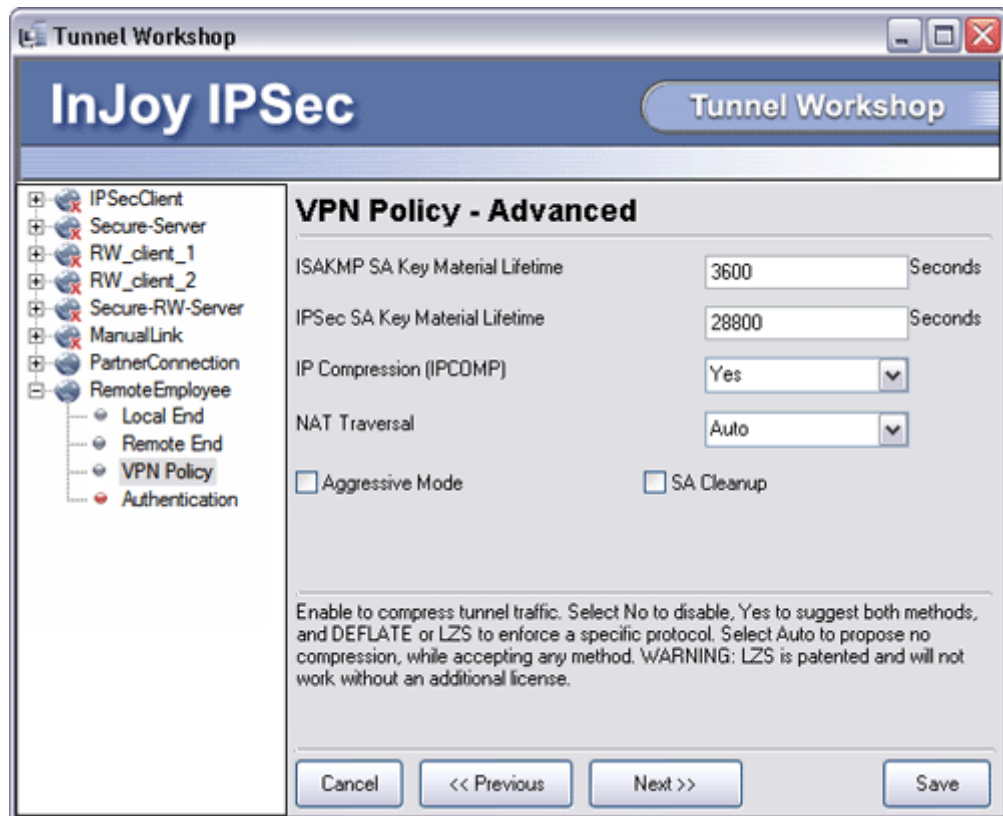
Because the employees use slow dial-up lines, it is helpful to mitigate the overhead imposed by IPSec. This is done using IP compression, which can at times significantly increase throughput. Aggressive mode is left disabled for this connection, since the employees will be using main mode negotiation.
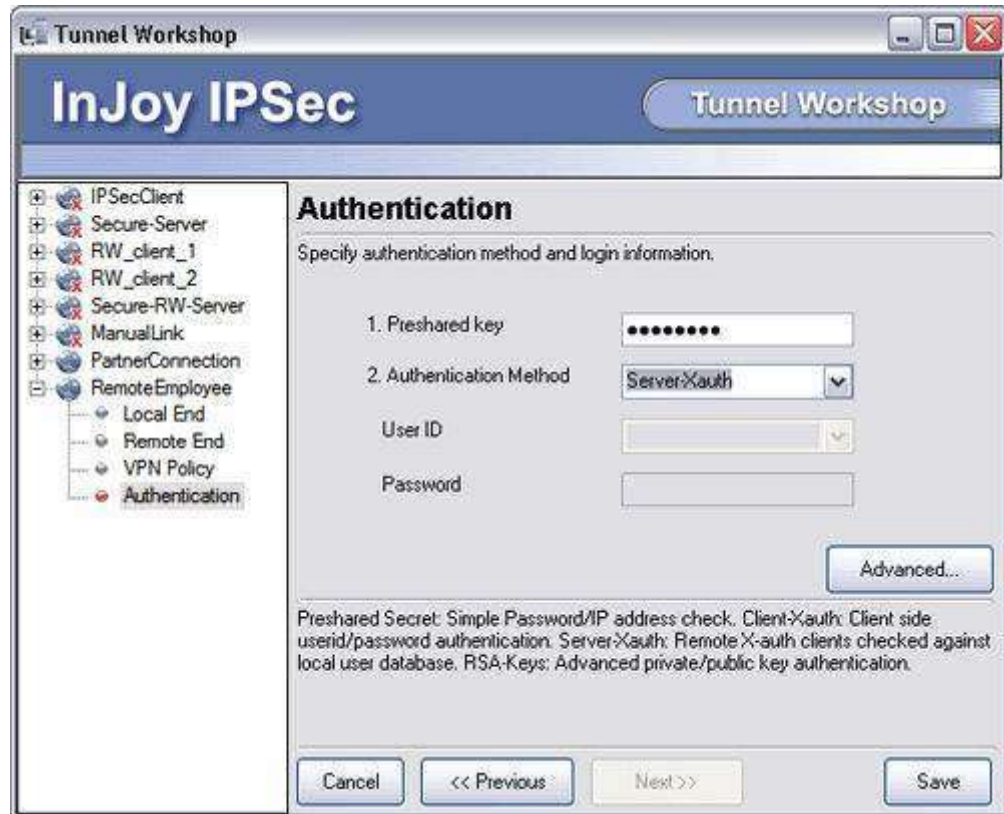
The network administrator chooses a slightly different Pre-shared Key ("3mploy33P4ssWD") for the employee connections.

Creating User Accounts for the Remote Employees

To allow the remote employees to login via the Extended Authentication method, the network administrator now adds entries to the **ipsec\vpn-auth.cnf** file for two employees:

```
CJanssen        Password = "0ranger0lls",
                Inner-IP="10.1.4.104"


KPeterson       Password = "2bgBust0ut",
                Inner-IP="10.1.4.105"
```

Note that each of these employees is also assigned a virtual inner IP number in the VPN server's local network, to avoid the need to manage the employees' dynamic IP addresses.

Configuration File Details
The Tunnel Workshop configuration steps described above add the following lines to the SA configuration file **ipsec\ipsec.cnf**:

```
RemoteEmployee              Description = "IPSec VPN to our employees!"
                            Local-IP = "My_IP",
                            Local-Net = "10.1.4.0",
                            Local-Mask = "255.255.255.0",
                            Remote-IP = "0.0.0.0",
                            Reinit = No,
```

```
                              Auth-Type = Server-Xauth,
                              IP-Compression = Yes,
                              Preshared-Secret = "-331188189f15db57b068ab278708"
```

Once again, the Pre-shared Key doesn't appear as it was entered because it has been encrypted.

10.3.Partner Company Configuration

While Softdev's network administrator was busy creating SAs, the network administrator at the partner company was creating an SA on his gateway machine. Note: if the Softdev network administrator would have had remote Firewall access to the partner company Firewall, all SAs could have been centrally managed.

To limit the number of screen shots, the input to the majority of the dialogs is shown as text - below:

SA Setup Dialog

SA Name:	"SoftdevComVPN"
SA Description:	"IPSec to Softdev.com!"

Local End Dialog

Local Gateway/Host	My_IP
Local Net	192.168.1.0
Local Netmask	255.255.255.0

The partner network administrator selects "My_IP" as the gateway's host address and enters the local network address range, 192.168.1.0 with a netmask of 255.255.255.0, as the local net. This will allow his internal work-stations in the 192.168.1.x address space to transparently access work-stations at the corporate headquarter and work-stations within the headquarter can transparently access the work-stations behind the partner's VPN gateway.

Remote End Dialog

Remote Gateway/Host	softdev.com
Remote Net	10.1.4.0
Remote Netmask	255.255.255.0

In the Remote End dialog, the partner company's network administrator enters "softdev.com" as the Softdev.com VPN server. This assumes that softdev.com will correctly resolve to the appropriate public IP address of Softdev (resolving softdev.com to an IP address happens automatically at run-time).

VPN Policy Dialog

Mode	Tunnel
Auth. Header	No
Encryption	3DES
Initiate IKE negotiations	Yes (checked)

Initiate IKE is enabled to allow both ends of the IPSec connection to initiate VPN negotiations.

VPN Policy Dialog (Advanced)

ISAKMP Key Lifetime	3600
IPSec Key Lifetime	28800
IP Compression	No
NAT Traversal	Auto
Aggressive Mode	No (unchecked)
SA Cleanup	No (unchecked)

The VPN policy and advanced VPN policy options chosen by the partner company's network administrator match the options chosen by Softdev.com's network administrator for this connection. This ensures that the Softdev.com VPN server and partner company gateway will be able to connect using IPSec.

Authentication Dialog

Pre-shared Key	"P4rtnerP4ssWD"
Authentication method	Client-Xauth
User ID	"SoftdevPartner"
Password	"2ndP4ssWD"

The partner company's network administrator enters the Pre-shared Key assigned by the two companies for this connection. He selects Xauth (in client mode) as an additional authentication method, since Softdev.com requires it, and enters the user ID and password assigned by Softdev.com for this connection.
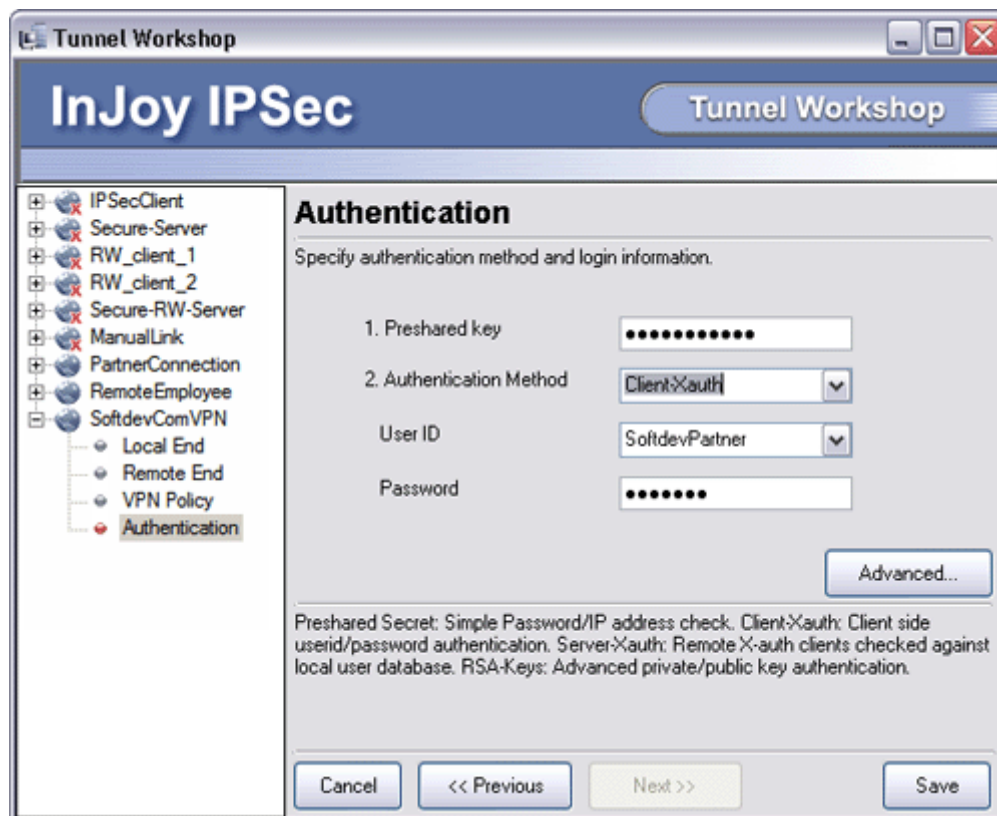
The partner company's security association for Softdev.com's VPN server is now complete and the connection is ready to be used!

Configuration File Details

The Tunnel Workshop configuration steps described above add the following lines to the security associations file **ipsec\ipsec.cnf**:

```
SoftdevComVPN Description = "IPSec VPN to Softdev.com!"
                    Local-IP = "My_IP",
                    Local-Net = "192.168.1.0",
                    Local-Mask = "255.255.255.0",
                    Remote-IP = "Softdev.com",
                    Remote-Net = "10.1.4.0",
                    Remote-Mask = "255.255.255.0",
                    ESP = 3DES,
                    Auth-Type = Client-Xauth,
                    User-Id = "SoftdevPartner",
                    Password = "-32129c24c41f9b33a4",
                    Preshared-Secret = "-50488a009e099a34d42fab0394"
```

10.4. Remote Employees Configuration

Softdev.com has given its remote employees the InJoy software in order to remotely connect to the VPN server and the Softdev.com network.

SA Setup Dialog

SA Name	"SoftdevComVPN"

SA Description	"IPSec to Softdev.com!"

Local End Dialog

Local Gateway/Host	Road Warrior
Local Net	No Local Net

In the Local End dialog, the employees configure the local host as "Road Warrior" and the local net as "No Local Net" since each of them is only a single dial-up host.

Remote End Dialog

Remote Gateway/Host	softdev.com
Remote Net	10.1.4.0
Remote Netmask	255.255.255.0

The employees configure the Remote End dialog using the same values used by the partner company's network administrator: softdev.com as the host name of the Softdev.com VPN server, and Softdev.com's network information, 10.1.4.0 with a netmask of 255.255.0.0.

VPN Policy Dialog

Mode	Tunnel
Auth. Header	No
Encryption	3DES
Initiate IKE negotiations	Yes (checked)

VPN Policy Dialog (Advanced)

ISAKMP Key Lifetime	3600
IPSec Key Lifetime	28800
IP Compression	Yes
NAT Traversal	Auto
Aggressive Mode	No (unchecked)
SA Cleanup	No (unchecked)

The VPN policy and advanced VPN policy options used by the employees match the options chosen by Softdev.com's network administrator for the Road Warrior connections, with one exception: for the employees' configurations, the Initiate IKE negotiations box is checked, since these are Road Warrior client machines.

Authentication Dialog

User: CJanssen	Pre-shared Key:	"3mploy33P4ssWD"
	Authentication method:	Client-Xauth
	User ID:	"CJanssen"

	Password:	"0ranger0lls"
User: KPeterson	Pre-shared Key:	"3mploy33P4ssWD"
	Authentication method:	Client-Xauth
	User ID:	"KPeterson"
	Password:	"2bgBust0ut"

The Pre-shared Key used by the employees is the one assigned by Softdev.com. The employees both choose to use Extended Authentication (Xauth) as clients, each one entering his own user ID and password in the related boxes.

Once each of the employees has finished filling out the dialogs in the Tunnel Workshop, the Road Warrior IPSec connections are ready to go.

Configuration File Details

The Tunnel Workshop configuration steps described above add the following lines to the SA configuration file **ipsec\ipsec.cnf**:

In User CJanssen's File

```
PartnerConnection          Description = "IPSec to Softdev.com!"
                           Local-IP = "0.0.0.0",
                           Remote-IP = "softdev.com",
                           Remote-Net = "10.1.4.0",
                           Remote-Mask = "255.255.255.0",
                           AH = No,
                           ESP = 3DES,
                           Auth-Type = Client-Xauth,
                           User-Id = "CJanssen",
                           Password = "-300e991a97099a548c3ab",
                           IP-Compression = Yes,
                           Preshared-Secret = "-331188189f15db57b068ab278708"
```

In User KPeterson's File

```
PartnerConnection          Description = "IPSec to Softdev.com!"
                           Local-IP = "0.0.0.0",
                           Remote-IP = "vpn.softdev.com",
                           Remote-Net = "10.1.4.0",
                           Remote-Mask = "255.255.0.0",
                           AH = No,
                           ESP = 3DES,
                           Auth-Type = Client-Xauth,
                           User-Id = "KPeterson",
                           Password = "-321e9f36851f9c549528",
                           IP-Compression = Yes,
                           Preshared-Secret = "-331188189f15db57b068ab278708"
```

10.5. Establishing the Tunnel

Once all of the configuration steps have been carried out, Softdev.com, its partner company, and its employees are ready to connect to one another. They follow these steps to bring the VPN online:

1 Because the most common cause of VPN problems is a simple loss of connectivity, Softdev.com, its partner and its employees all perform one

final step before taking their VPN live—they test their connections to one another, ensuring that they can ping one another in both directions and transfer sizable amounts of data without errors.

2 Once this has been done, Softdev.com's network administrator activates the IPSec configuration, by reloading the IPSec configuration.

3 The partner company's network administrator and the remote employees activate their IPSec configuration.

The IPSec link between Softdev.com and its partner company is now live. The employees, too, can now connect at will; each time either of them connects to the Internet with the InJoy software running, they will have a route through a secure IPSec link to the hosts in Softdev.com's internal network.

10.6.Monitoring and Maintenance

As the Softdev.com VPN carries out its day-to-day responsibilities, securely passing network traffic between the VPN hosts across the public Internet, the network administrator at Softdev.com takes care to monitor the VPN and its performance. This is primarily done in two ways:

- The network administrator routinely runs the InJoy Firewall™ GUI and inspects the IPSec monitors.

- He also regularly searches the **logs\ipsec.log**, **logs\vpn-auth.log** and **logs\pluto.log** files for unusual activity. Once a week, he archives these logs to long-term storage, in case he should need to refer to them in the future.

For details on using the IPSec monitors in the InJoy Firewall™ GUI, please refer to Section 8.2, "Monitoring Users and Tunnels." For details on using the InJoy IPSec logs, please refer to Section 8.3, "Logging and Trace Files."

Part IV
Advanced Features Guide

11 Using Road Warrior Support

This section is intended to give you a more in-depth description of the benefits, limitations, and operational details of InJoy's support for Road Warrior IPSec connections.

Additional instructions and examples for enabling and using Road Warrior support can be found in Section 10, "A VPN Case Study."

11.1.Introduction to Road Warriors

Road Warriors are VPN clients that have dynamic IP addresses. Most often, Road Warriors are remote or telecommuting employees that use dial-up or other consumer-class Internet connections. The Road Warrior feature however doesn't need to be limited to handling dynamic IP addresses. InJoy's Road Warrior support is useful whenever you need a security association that does not refer to a particular IP address.

Key Benefits

Road Warrior support provides a number of key benefits to network administrators.

Dynamic IP Address Support
InJoy's Road Warrior support allows you to create SAs for use with any IP address that is not known in advance, whether dynamic or static. By combining Road Warrior support with Extended Authentication, you can associate IPSec connections with particular users, rather than particular public IP addresses.

Static Inner-IP Support (Virtual IP)
InJoy's Road Warrior support also provides Inner-IP capability, which allows you to assign connecting hosts a virtual IP address inside the local network, greatly simplifying administration tasks.

Blanket Security Association Support
A single Road Warrior SA can manage connections from multiple public IP addresses. This provides the ability to create one "blanket" SA that can be used with all client connections to a given VPN server.

Using Road Warrior Support

Road Warrior support can be enabled in the Tunnel Workshop in either the Local End (for InJoy dial-up clients) or Remote End (for InJoy VPN servers) dialogs.

1. Remote Gateway/Host	Road Warrior ▾

To enable Road Warrior support using the **ipsec\ipsec.cnf** configuration file, set either the Local-IP or Remote-IP keyword to a value of 0.0.0.0. Recall the VPN server SA for remote employees from Section 11, for example:

```
RemoteEmployee          Description = "IPSec VPN to our employees!"
                        Local-IP = "My_IP",
                        Local-Net = "10.1.4.0",
                        Local-Mask = "255.255.255.0",
                        Remote-IP = "0.0.0.0",       << Specifies Road Warrior
                        Reinit = No,
                        Auth-Type = Server-Xauth,
                        IP-Compression = Yes,
                        Preshared-Secret = "-331188189f15db57b068ab278708"
```

11.2. Road Warrior Limitations

InJoy's Road Warrior support is a powerful tool for network administrators. However, by its nature it is subject to a number of functional limitations.

One Security Association

Because the Road Warrior SA is associated with all dynamic IP hosts, only one Road Warrior SA can exist for each InJoy VPN server. All incoming Road Warrior connections will be managed by this single Road Warrior SA (causing the configuration flexibility to be somewhat weakened).

Client-only Initiation

Because a Road Warrior's public IP address is not known in advance and because many Road Warriors have intermittent physical connections, the VPN server cannot initiate Road Warrior IKE negotiations. All Road Warrior IKE negotiations must be initiated by Road Warrior clients.

Connectivity/Compatibility Limitations

In order for a Road Warrior connection to be established, the VPN server must support Road Warrior SAs. Therefore, an InJoy Road Warrior client can connect to a given VPN server only if the server in question supports Road Warriors.

Note that an InJoy VPN server can accept connections from IPSec clients on dynamic IP addresses even if the client software doesn't explicitly support Road Warriors, so long as the client is careful to reflect the ISP assigned IP address at all times (i.e. the client must update its own configuration with

every ISP reconnect). For further details about compatibility with specific third-party vendors, please refer to the "IPSec Interoperability Guide" documents.

11.3. Operational Details

Before using InJoy's Road Warrior support, you should familiarize yourself with a number of Road Warrior operational details.

Road Warrior IKE Negotiation

Because the Road Warrior SA is used as a template from which "real" SAs are created, each connecting Road Warrior client can negotiate independently for unique IPSec connection properties, including connection-specific:

- Packet-level authentication (SHA vs. MD5)

- Packet-level encryption (3DES, DES, or NULL)

- IP Compression (enabled vs. disabled)

Road Warriors with Subnets

InJoy IPSec supports remote Road Warriors with internal subnets behind.
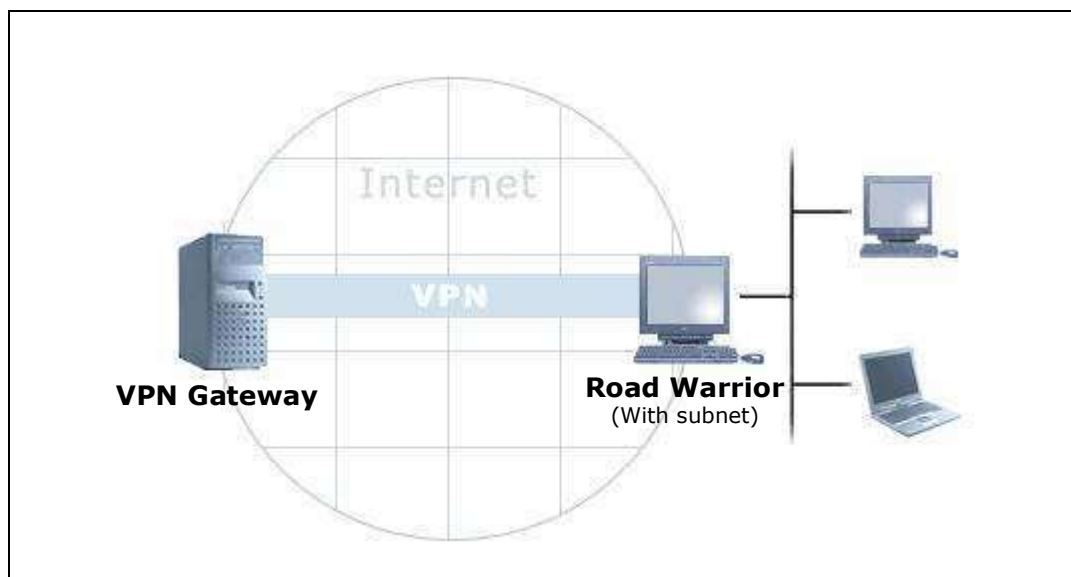
To properly utilize this support, the VPN Gateway **must** be configured with a common Road Warrior SA that defines a Remote Network of "appropriate proportions"; i.e. with enough addresses to allow the remote Road Warriors to define their own subnets within this "master network".

Because traffic destined for a particular RW subnet will be routed only to the first IPSec run-time tunnel associated with the subnet, Road Warriors should use unique subnets.

For an example of how Road Warrior clients with subnets can be configured, refer to the Road Warrior Scenario "Road Warriors with Subnet".

Authentication and Identification

Extended Authentication or RSA signature authentication are the preferred authentication methods for Road Warrior clients. These options give the VPN network administrator the ability to assign authentication secrets (and Inner-IP addresses) on a per-user basis, thereby enhancing network security and simplifying administration.

Security Implications

The use of Road Warrior connections in conjunction with NAT traversal is associated with a set of additional security implications which must be considered.

For details on the use of Road Warrior connections with NAT traversal, please refer to Section 13, "Using IPSec behind NAT."

11.4. Sample Road Warrior Scenarios

Road Warrior support can be the ideal IPSec solution for a variety of different needs and scenarios. Some common scenarios are outlined below.

Stand-alone Road Warrior hosts

In this simple scenario, one or more stand-alone (i.e. no internal network behind the RW) Road Warrior connects to a VPN server, which then assigns the Road Warrior an Inner-IP address. Authentication can be either a simple Pre-shared Key or something more secure, like Extended Authentication or RSA key exchange.

For a detailed walkthrough of a similar scenario, refer to the discussion in Section 10, "A VPN Case Study."

Road Warriors with Subnet

In this scenario, a VPN Gateway hooks up with two remote Road Warriors, each with subnets behind them.

The example below shows the VPN Gateway allocating the Class-A net 10.0.0.0, while two separate RW based subnets each allocate a Class C subnet from that VPN Gateway specified Class-A network.

The VPN Gateway SA specifies the Remote Net as 10.0.0.0 / 255.0.0.0, while RWs clients allocate portions of that network. For example 10.0.1.0 / 255.255.255.0.

VPN Gateway
217.39.43.22

10.0.1.2

10.0.1.3

217.212.45.55
Inner-IP: 10.0.1.1

Internet

217.212.45.67
Inner-IP: 10.0.2.1

10.0.2.2

10.0.2.3
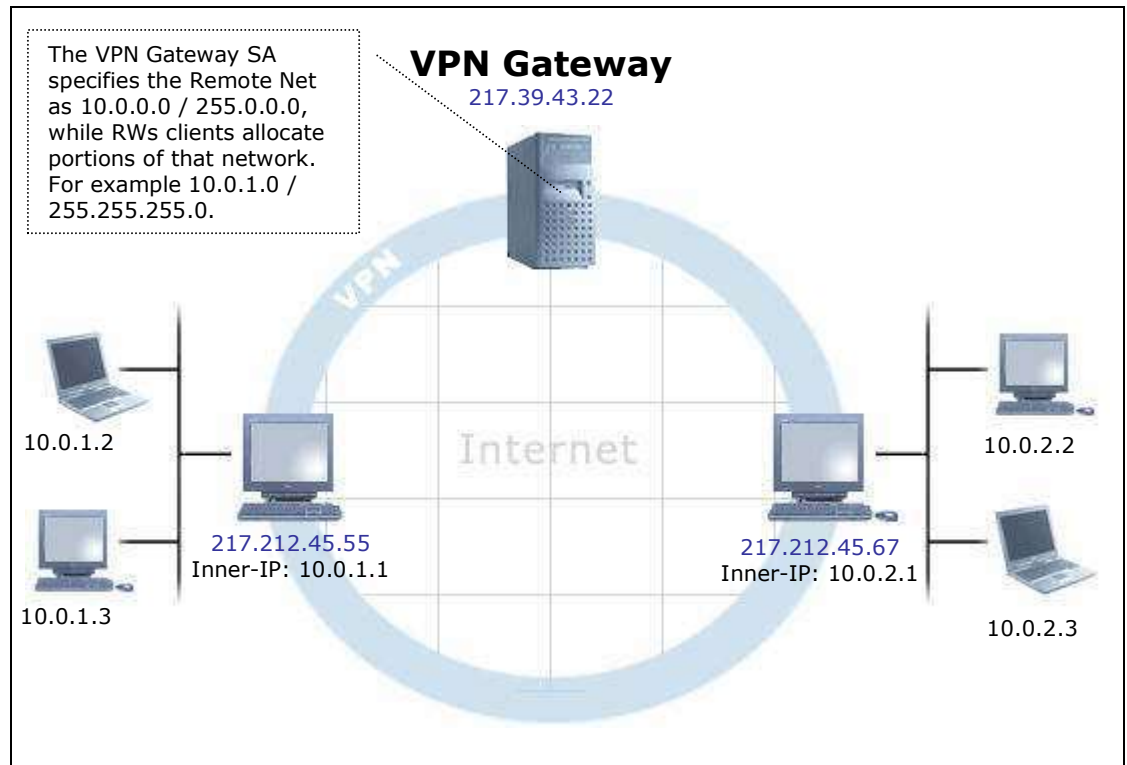
Multiple Telecommuting Road Warriors

Another common scenario involves a pool of telecommuting employees that connect to a central VPN server; each employee may be using dial-up or some other connectivity technology.

Because it is usually desirable to implement per-user security, and to assign each user his or her own IP address on the private network, Extended Authentication is usually used to identify users. RSA signatures may be used as well, to create a "must-know secret" for the telecommuting employees as a class.

For details on using RSA signatures for an entire class of connecting clients, please refer to Section 16.3, "RSA Digital Signatures."

IPSec between Two dynamic-IP Hosts

Sometimes it is necessary to create a confidential, secure connection between two dynamic-IP hosts. Some coordination between parties is required in this case to ensure that before either member initiates a connection, the other member is already running the InJoy software with an SA for remote Road Warrior support enabled.

To make possible a setup where both involved IPSec end-points use dynamic IP addresses, the remote IP address in the SA of both sides should be set to either:

- A DNS host name that always reflects the current IP address of the other IPSec end-point.

- An IP address that is exchanged manually among the involved parties (i.e. the dynamic IP address).

The second approach is acceptable when the dynamic IP address change very rarely, which is the case with certain Internet Service Providers.

12   Using Inner-IP Support

This section is intended to give you an overview of the benefits, limitations, and operational details of InJoy's support for Inner-IP address assignment.

Additional instructions and examples for enabling Inner-IP in the InJoy software can be found in Section 5, "Configuration" and Section 10, "A VPN Case Study."

12.1.Introduction to Inner-IP

Inner-IP is a feature that allows the VPN administrator to assign a virtual IP address (a "Red Node IP") to any connecting IPSec client. This address:

- Must be assigned at connect time (using the SA configuration for the connection in question)

- Can be used to refer to the host instead of the host's external public IP (or "Black Node IP") address

Key Benefits

The use of Inner-IP addresses provides several key benefits to IPSec network administrators.

Ease of Administration
Connected IPSec hosts which have been assigned an Inner-IP can be managed using the Inner-IP. This greatly simplifies the administration tasks faced by network administrators—all addresses on the VPN can fall within a single, unified internal address range.

Increased Security
Because Inner-IP can be assigned on a per-user basis using Extended Authentication, security policies can also be created and enforced on a per-user basis.

Because the blanket use of Inner-IP addresses creates a unified internal address range for all remote hosts, network administrators can create a simpler firewall policy - based on IP addresses. Policies involving a single address range are both less prone to error and more robust than security policies or rulesets involving multiple disparate address ranges which may pass across or through any number of networks or gateways.

Inner-IP can be enabled either from the VPN client side or gateway side, depending on the needs of the network administrator (and to some extent the specific IPSec protocols in use – be ware of third party equipment limitations).

Client-Side Inner-IP
To enable and set the Inner-IP from the VPN client side, include the Inner-IP keyword in the related SA in **ipsec\ipsec.cnf** on the client side. Refer to the Inner-IP scenarios section below, for more information.

Server-Side Inner-IP
To enable Inner-IP support for a particular user from the server side, edit the **ipsec\vpn-auth.cnf** configuration file, setting a value for the Inner-IP keyword for the user in question. Recall the VPN server entry for the users CJanssen and KPeterson, for example:

CJanssen   Password = "0ranger0lls",
                Inner-IP="10.1.4.104"

KPeterson   Password = "2bgBust0ut",
                Inner-IP="10.1.4.105"

Inner-IP Routing Implications
Because the Inner-IP feature uses NAT to change the public IP address of a remote VPN host, it is important to consider the possible routing implications.

To illustrate the most common routing pitfall, envision a head office network where all traffic destined to public IP addresses is "default routed" to the designated VPN Gateway PC. The VPN Gateway PC then handles the traffic according to the IPSec SA database and ensures IP packets are correctly distributed to the various remote VPN Clients. This is a very common setup and a normal routing configuration.

In contrast, now envision a scenario where Inner-IP is used. Traffic from a remote VPN host is tunneled into the head office network with a local source IP address of e.g. 10.0.0.1 (the Inner-IP). When reply traffic is generated to 10.0.0.1, it may appear local and accordingly won't be routed to the head office VPN Gateway. The result could be routing havoc and the remote VPN Client would never see its reply traffic.

The solution to this problem is trivial. As the IPSec network administrator, you must generally consider whether reply packets to incoming VPN traffic will appropriately be routed back to the VPN Gateway, and thereby be given the opportunity to be routed back out to the remote VPN hosts. An important notice in this regard, is to carefully plan in advance, which IP address segments that are to be used in the different VPN locations. By simply avoiding conflict in the VPN subnets and by routing all non-local traffic to the VPN Gateway, routing problems are generally avoided.

12.2. Inner-IP Limitations

There are a number of limitations to be aware of when using the Inner-IP features of the InJoy software.

Inner IP Address Must Be Unique

Each VPN host or client on a particular VPN network must have a unique IP address. For this reason, you can not assign the same Inner-IP address to multiple hosts. If this occurs, only the client that connected first will be able to communicate and due to the nature of the IP protocol, such misconfiguration may lead to serious network disruption or undefined behavior.

Server-side Inner-IP Assignment Only Possible Through X-auth

The InJoy software only permits server-side Inner-IP assignment using Xauth and the **ipsec\vpn-auth.cnf** configuration file. Other assignment methods outlined in the draft standard (including RADIUS or directory services) are not implemented.

Inner-IP Disables the Use of the Public IP

Because Inner-IP uses NAT to change the address of the related host in the packet headers themselves, you cannot assign an Inner-IP address and then use both the Inner-IP and "real" address of the host in question in VPN network transactions or network configuration. Only one address (Inner-IP or the public address) can be valid at a time.

12.3. Operational Details

Inner-IP uses Network Address Translation (NAT) on IPSec packets before transformation by AH or ESP. NAT works "backward" in this case when compared to traditional NAT in that addresses are being translated from public to private for routing, rather than vice-versa, in order to provide a more convenient IP address for administrative and security tasks.

Related Documents

InJoy's Inner-IP implementation is based on the following IETF draft documents:

draft-ietf-ipsec-isakmp-xauth-02.txt
draft-ietf-ipsec-isakmp-mode-cfg-05.txt

12.4. Sample Inner-IP Scenarios

Inner-IP can be used to simplify network administration for a variety of different needs and scenarios.

Pool of Road Warriors

Most Road Warriors use dial-up Internet access and therefore have dynamic IP addresses. Network administration involving a large pool of telecommuting employees can create a blizzard of un-maintainable firewall and service rules based on IP addresses that are subject to change at any time.

Inner-IP is commonly used together with Extended Authentication under such circumstances to associate each telecommuting user with a particular static IP

address on the internal network, greatly simplifying network and security administration.
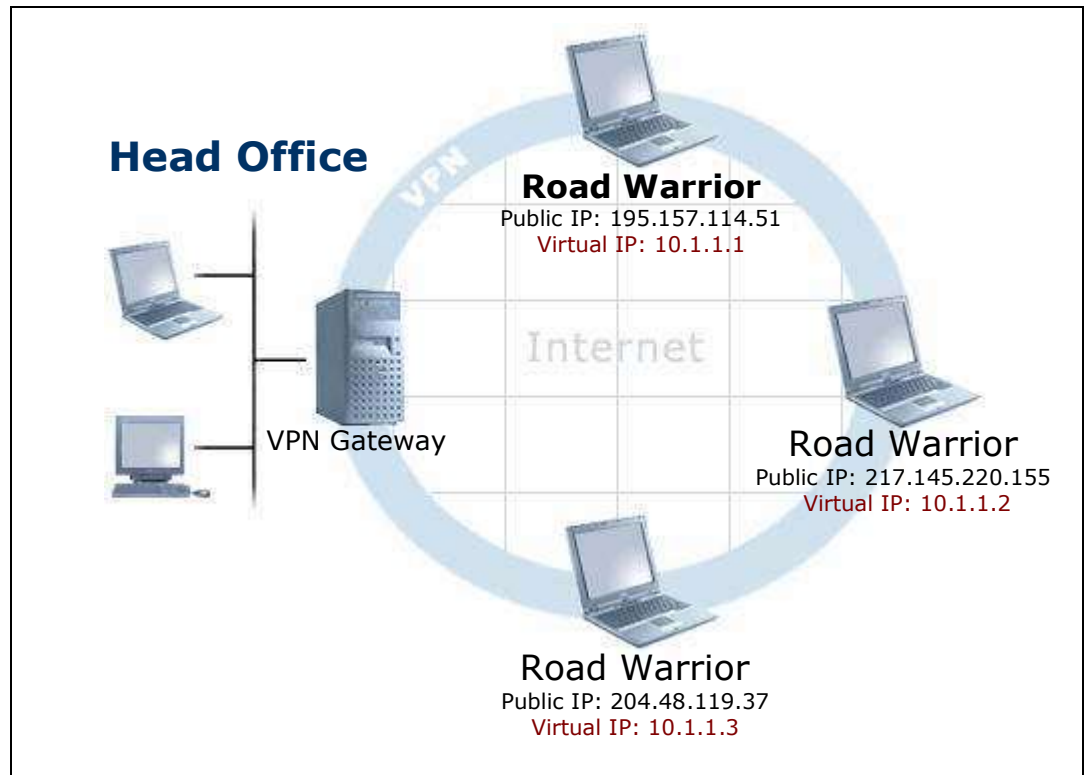
13   Using IPSec behind NAT

This advanced section is intended to give you an overview of the tools available for using IPSec through gateways or hosts that employ Network Address Translation (NAT) to connect private sub-networks to the public Internet.

13.1. Introduction to NAT Traversal

In IPSec tunnels, encapsulated or authenticated packet headers can not be changed during data transmission. These types of changes are explicitly forbidden by IPSec. By design, IPSec takes strong measures—MD5 or SHA authentication, or DES or 3DES encryption—to prevent such changes.

Because of this, NAT poses a natural limit to the functioning of normal IPSec connections; by its very nature, NAT must rewrite packet headers before forwarding them to the next host in the transmission path.

NAT traversal (NAT-T) is a feature that allows IPSec connections between two endpoints to tunnel through NAT gateways or hosts which would otherwise be impenetrable to IPSec.

Key Benefits

InJoy's NAT-T support provides two key benefits to IPSec network administrators.

IPSec through NAT
NAT-T allows IPSec to function through NAT gateways or hosts, removing the necessity for network reorganization of any kind before strong security can be added.

Administrative Transparency
NAT-T requires little configuration or monitoring on the part of the network administrator. When enabled, NAT-T will automatically detect NAT devices in the IPSec transmission path and adjust for them accordingly, making NAT-T a nearly transparent feature from use or administration perspectives.

Using NAT-T Support

NAT-T is enabled (set to "Auto") by default when you create security associations using the Tunnel Workshop. If you wish to prevent NAT-T, you can change this setting in the advanced section of the VPN Policy dialog.
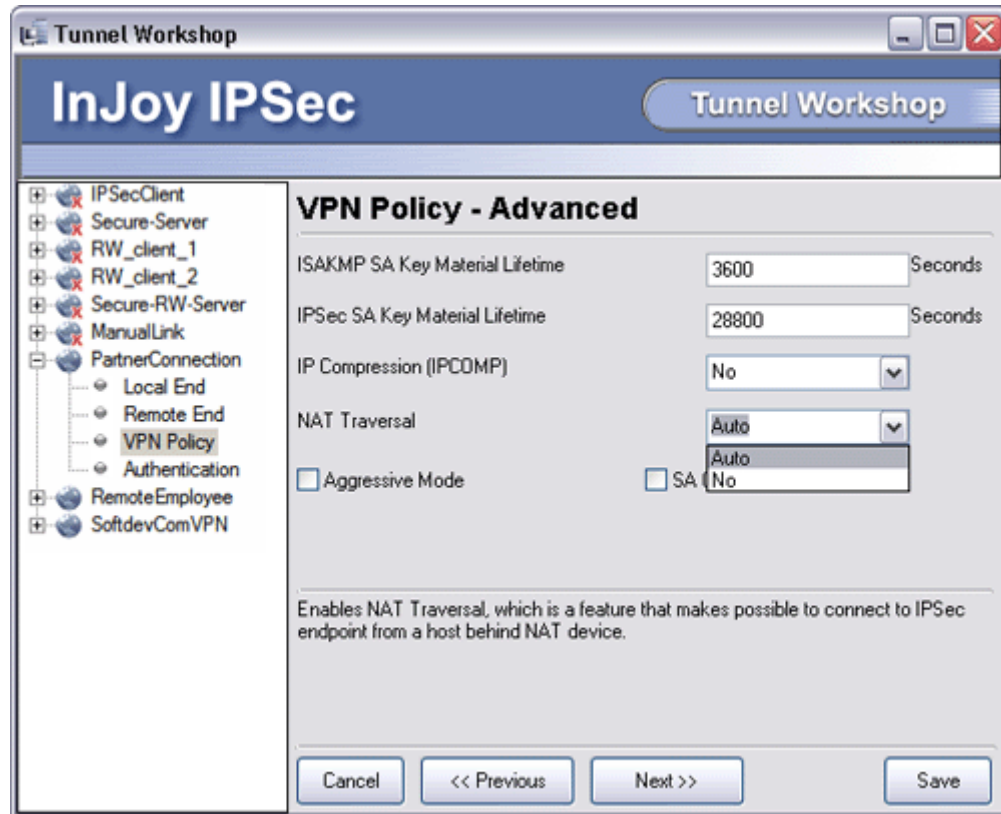
In the configuration files, the absence of a NAT-Traversal keyword for a particular security association indicates that NAT-T is enabled and will function as necessary (automatically). To explicitly disable NAT-T for a particular Security Association, set the value of the NAT-Traversal keyword to "No" as shown below:

```
NAT-Traversal = No
```

13.2. NAT-T Operation Details

NAT-T initiation and communication involves a number of distinct steps, all of which occur transparently and without user intervention:

1 During the IKE negotiation process, both endpoints agree to support NAT-T over the connection in question.

2 Endpoints detect the absence or the presence of NAT devices by calculating and comparing hashes if the IP addresses sent and the IP addresses received in the packet header.

3 If NAT devices are present, the IPSec packets are encapsulated yet again using a UDP header and an additional NAT-T header to supply the extra address information.

4   Packets routed over the network between endpoints.

5   The NAT-T-aware receiving endpoint detects NAT-T traffic, de-encapsulates it, then processes it accordingly at the application layer.

NAT-T relies on the IKE port (500) for actual transmission of NAT-T encapsulated packets.

13.3.NAT-T Limitations

There are several limitations and issues to be aware of when using NAT-T on your VPN.

Processing and Traffic Overhead

NAT-T imposes approximately 200 bytes of overhead during IKE negotiation and about 20 bytes of additional overhead for each packet. Depending on the amount of available bandwidth and processing power, the difference in throughput may in some instances be measurable.

Reduced Security

Because Authentication Header (AH) transforms actually authenticate packet headers as well as packet payloads, and because NAT-T provides a mechanism by which packet headers can be modified in transit, AH and NAT-T do not function together; NAT-T operates only on ESP-transformed packets.

Because of this authentication deficiency, the trust level between hosts using NAT-T is greatly reduced; NAT-T should not be used when the greatest level of host-based authentication is required.

Security Association Details

Because the original IP address is changed by intermediate NAT devices, it is impossible for the NAT-T Gateway to determine the original internal IP address of the NAT-T client. In consequence, you need to explicitly include the IP address of the Remote NAT-T endpoint in the Remote-Net/Remote Mask fields on the VPN Server. For instance, if the remote NAT-T client has a local network behind it, then the VPN Gateway can specify a wider netmask, such as 255.255.255.0:

```
…
Remote-Net = "192.168.1.0",
Remote-Mask = "255.255.255.0",
…
```

The **important** rule of thumb in this regard is to ensure that the IP address of the NAT-T client is included in the Remote-Net on the VPN Gateway and included in the Local-Net on NAT-T client.

Another variant of letting the VPN server to define original internal IP address of the NAT-T client is defining generic net/mask on the client and afterwards assigning Inner IP addresses through Extended Authentication (XAUTH). This would require specifying Inner-IP attribute in vpn-auth.cnf on the server side:

```
…
Password = "super",
Inner-IP = "10.11.2.45",
…
```

Detecting Dead Tunnels

Any given NAT device cannot be assumed to maintain constant port mapping, hence a NAT-T aware IPSec implementation must be able to quickly detect tunnels that cease to be operational.

InJoy IPSec uses the NAT-T heartbeat mechanism (part of the NAT-T standard draft) to detect dead tunnels. Because NAT-T heartbeat isn't supported by all vendors, it is recommended that you set the key material lifetimes low (e.g. 30 minutes), as a second measure to quickly detect dead tunnels.

The NAT-T heartbeat mechanism is turned on automatically when IPSec detects a NAT device between endpoints.

14 Using IP Compression

This advanced section is intended to give you an overview of the InJoy software's IP compression feature and details on how to enable it in your security associations.

14.1. Introduction to IP Compression

IP compression (IPComp) is a protocol used to apply data compression to IP packet payloads. This provides network users with two main benefits:

- Reduced network bandwidth utilization
- Increased overall network throughput

These benefits come at the expense of the increased CPU processing overhead required to apply common compression algorithms to a stream of network data.

Two compression algorithms, Deflate and LZS, are supported by InJoy IPSec. While the Deflate algorithm is able to achieve better compression than LZS by a typical 10-20% margin, Deflate is considerably more computation-intensive, requiring two passes to compress each data set.

Standards Documents

IPComp is defined and detailed in the following Request for Comment (RFC) documents:

- RFC 3173—IP Payload Compression Protocol (IPComp)—Proposed Standard (2001)
- RFC 2393—IP Payload Compression Protocol (IPComp)—Proposed Standard (1998)
- RFC 2394—IP Payload Compression Using DEFLATE—Informational
- RFC 2395—IP Payload Compression Using LZS—Informational
- RFC 3051—IP Payload Compression Using ITU-T V.44 Packet Method—Informational

LZS licensing

The company Hi/fn, Inc. holds several patents related to the LZS compression algorithm. For this reason, potential users of LZS must purchase a valid license from Hi/fn, Inc. before deploying solutions which make use of the LZS compression method.

Because of this licensing requirement, shipped versions of the InJoy software do not include the LZS compression code. Contact F/X Communications if you wish to obtain a license to use LZS with the InJoy software.

14.2.IP Compression Configuration

When configuring IPComp for a particular security association, you can select one of four IPComp negotiation options:

- No (decline requests to use IPComp)

- Yes (request or agree to use IPComp, either mode)

- Deflate (request or agree to use IPComp, deflate mode only)

- LZS (request or agree to use IPComp, LZS mode only)

- Auto (agree to requests to use IPComp)

Note that for IPComp to be used for a negotiated connections, one endpoint must explicitly request it. When both endpoints are set to Auto, neither host will request IPComp and IPComp will not be used for the connection; therefore, at least one endpoint should be set to Yes, Deflate, or LZS if IPComp connections are desired.

Enabling IPComp

IPComp can be enabled for a particular security association using the VPN Policy—Advanced dialog in the Tunnel Workshop. This dialog can be reached by clicking the advanced button in the VPN Policy dialog.
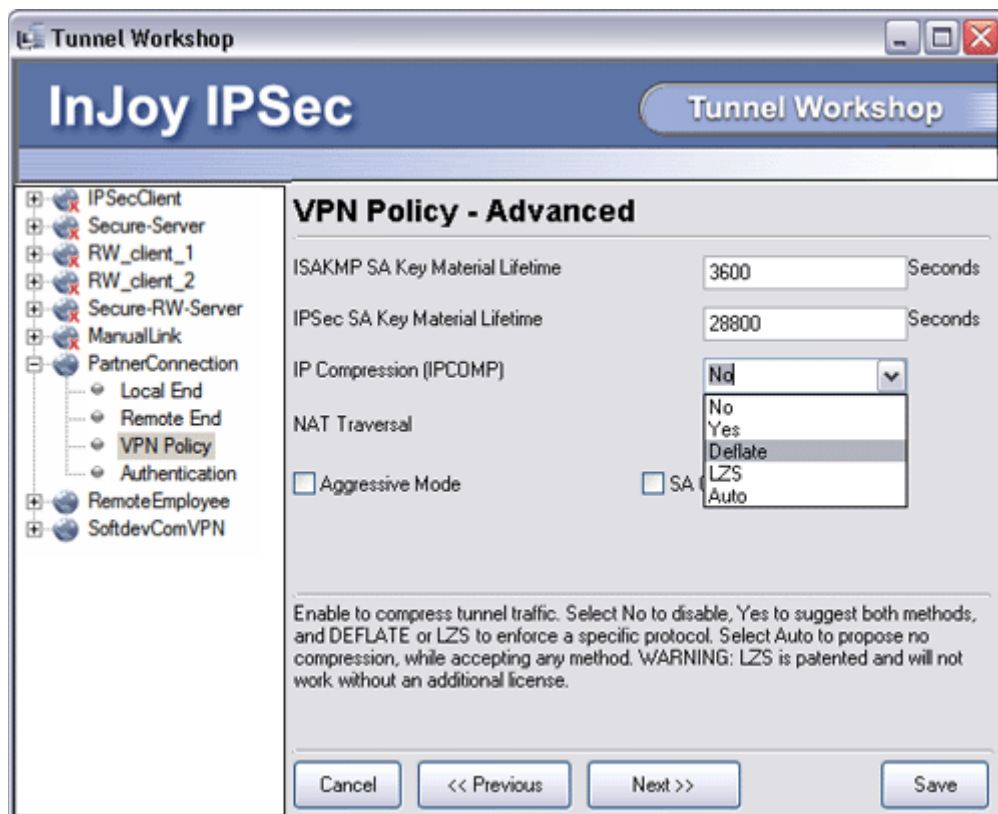
For details on using the Tunnel Workshop to configure security associations, please refer to Section Section 5, "Configuration".

IPComp can also be enabled in the **ipsec\ipsec.cnf** file by adding the IP-Compression keyword to a security association, along with one of Yes, No, Deflate, LZS or Auto as a value:

```
IP-Compression = Deflate
```

15   Using Manual Keying

This advanced section is intended to give you an overview of the InJoy software's allowance for manual keying, including details on how to enable it in your security associations.

15.1. Introduction to Manual Keying

At times, it can be inconvenient or impossible for one reason or another to run the Pluto IKE Server normally responsible for packet-level key management.

When this is the case, the InJoy software provides you with the ability to supply a set of keys manually, in advance, which will be then used for authentication and/or encryption of all packets that pass over a particular IPSec connection, thus eliminating the need for key exchange management or an Internet Key Exchange server like Pluto.

Note: You can not use manual keying if the local or the remote side is a Road Warrior. This is because IPSec must obtain the RW IP address through the IKE negotiations and with manual keying, no such negotiation takes place.

15.2. Manual Keying Drawbacks

Manual keying is relatively insecure—though it is still more secure than not using IPSec at all!

When using manual keying, the key used for encryption and authentication remains static, leading to two fundamental security issues:

- Potential thieves or malicious third parties have an indefinite amount of time (until keys are manually changed) to try to discover these keys using data analysis
- Once manual keys are compromised, thieves or malicious third parties can steal data for an indefinite amount of time (until keys are manually changed) with no additional effort

For these reasons, manual keying is not recommended under normal circumstances; instead, it should only be used for one-time communications or when there is no possibility of running an IKE Server.

15.3. Using Manual Keying

In order to use manual keying, additional configuration options must be included in the security associations in **ipsec\ipsec.cnf** on the two hosts involved.

Keys and SPI Values

If AH (authentication) is being used for the connection in question, the following configuration options must be added to the SAs in question:

- **AH-Key**, which specifies the local authentication header key.

- **Remote-AH-Key**, which specifies the remote authentication header key.

- **AH-Receive-SPI**, an arbitrary identification number for AH receiving for this security association.

- **AH-Transmit-SPI**, an arbitrary identification number for AH sending for this security association.

If ESP (encryption) is being used for the connection in question, the following configuration options must be added to the security associations in question:

- **ESP-Key**, which specifies the local encryption key.

- **Remote-ESP-Key**, which specifies the remote encryption key.

- **ESP-Receive-SPI**, an arbitrary identification number for ESP receiving for this security association.

- **ESP-Transmit-SPI**, an arbitrary identification number for ESP sending for this security association.

Note that the values of these options must mirror each other on the two IPSec endpoints, as is illustrated in this table:

Value on Host A	Matching Value on Host B
AH-Key	Remote-AH-Key
Remote-AH-Key	AH-Key
AH-Receive-SPI	AH-Transmit-SPI
AH-Transmit-SPI	AH-Receive-SPI
ESP-Key	Remote-ESP-Key
Remote-ESP-Key	ESP-Key
ESP-Receive-SPI	ESP-Transmit-SPI
ESP-Transmit-SPI	ESP-Receive-SPI

Mandatory Key Lengths

Depending on the type of AH hash (MD5 or SHA) or ESP encryption (DES or 3DES) used, manual keys in an SA must be of particular lengths. If both AH and ESP is being applied to connection packets, the lengths are:

- 16 characters (or 32 hex digits) for MD5 authentication

- 20 characters (or 40 hex digits) for SHA authentication

- 8 characters (or 16 hex digits) for DES encryption

- 24 characters (or 48 hex digits) for 3DES encryption

If only ESP is being applied, the lengths include both encryption and authentication keys in unified strings:

- 24 characters (or 48 hex digits) for DES + MD5 ESP

- 28 characters (or 56 hex digits) for DES + SHA ESP

- 40 characters (or 80 hex digits) for 3DES + MD5 ESP

- 44 characters (or 88 hex digits) for 3DES + SHA ESP

Keys must be enclosed in quotes and unbroken on a single line (i.e. without embedded carriage returns).

Sample Security Associations

To understand the use of manual keys, it might be helpful for you to study the deployment example in section 17.1, "Simple VPN Using Manual Keying".

16 Authentication Methods

This advanced section is designed to provide a more detailed introduction to the range of host authentication options and tools offered to InJoy IPSec users, including:

- Pre-shared Key authentication
- Extended Authentication (Xauth)
- RSA Signature authentication
- Group authentication

For a briefer, general introduction to common authentication methods used with IPSec, please refer to Section 3.3, "Authentication Methods."

16.1. Pre-shared Keys

Pre-shared Key authentication is the most fundamental type of host authentication, and is supported by every IPSec implementation.

Introduction

Pre-shared Key authentication is simply the secure exchange and verification of a text secret, or password, which has been shared in advance and is known to both hosts. When each host has verified that the other host knows the secret, both hosts are considered to be authenticated to one another and the negotiation for the IPSec connection proceeds.

Other authentication methods, with the exception of RSA Signatures, may be used in addition to Pre-shared Key authentication, in order to achieve more robust authentication.

Limitations

Pre-shared Key authentication is much less secure than other types of authentication used with IPSec for several reasons:

- Only a single, generally brief text secret is needed for authentication; such a secret is easily remembered and communicated through any number of channels.
- The secret is in no way related to the host's IP address, users, or any other identifying features; any host that can provide the secret can be authenticated.
- In many IPSec implementations (though not in InJoy IPSec), the text secret is stored in cleartext in configuration files, easily retrievable by users under some circumstances.

- In order for both endpoints to know the secret in advance of IPSec communication, the secret must at some point be communicated from endpoint to endpoint, either electronically or by human contact. Often, this exchange is insecure or can be intercepted by third parties.

- Because all Road Warriors associated with a VPN server use a single template security association, all Road Warriors must also use the same Pre-shared Key, meaning that a Road Warrior key may be known by a large number of people.

- Pre-Shared Keys are limited to 30 bytes in length.

Because of this laundry list of security issues, the use of Pre-shared Key authentication as the primary (i.e. only) host authentication method is only recommended when authentication security is of low priority.

Configuration

The Pre-shared Key for a particular SA can be entered using the Authentication dialog in the Tunnel Workshop. This is the preferred method for entering Pre-shared Keys, since the Tunnel Workshop will automatically encrypt the key before storing it in the configuration file for security.

For general information on using the Tunnel Workshop to configure authentication for SA, please refer to Section 5, "Configuration".

Pre-shared Keys can be configured in the **ipsec\ipsec.cnf** file using the Preshared-Secret keyword:

```
IPSecClient   Description = "VPN client using Pre-shared Key"
              Local-IP = "0.0.0.0,
              Remote-IP = "vpn.mycompany.com",
              ESP = 3DES,
              Preshared-Secret = "4pplesN0r4nges"
```

16.2. Extended Authentication (Xauth)

Extended Authentication (Xauth) is an IETF draft standard extension to IPSec which supports user-based authentication.

Introduction

Xauth requires that the VPN client supply two pieces of identifying material—a username and a password—to the VPN server in order to be authenticated. This allows multiple users to be associated to a single host, each one supplying unique authentication material in order to connect.

Xauth also allows user-based authentication from a pool of Road Warriors. By assigning Inner-IP addresses to such connections, it also provides the ability to create user-specific firewall rules and perform other types of security filtering based on internal IP addresses.
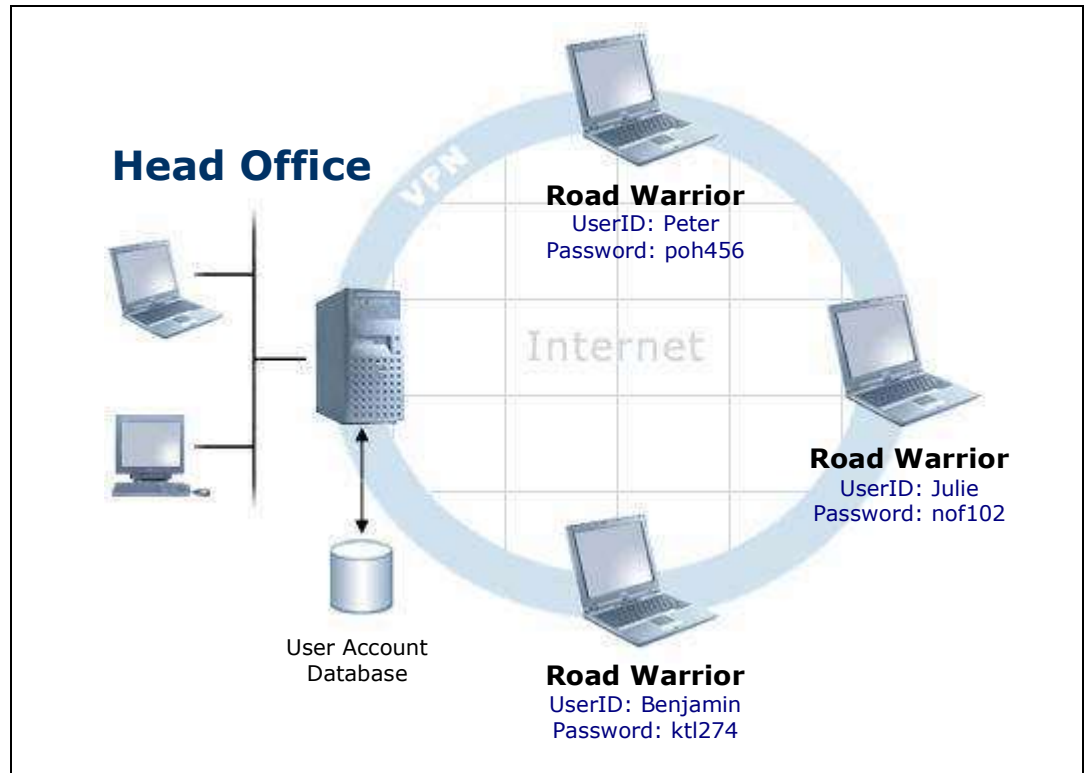
Limitations

The Pluto IKE Server implements only version 2 of the IETF draft standard. Only the Inner-IP capability of Configuration Mode is supported. Aspects of Configuration Mode not supported include:

- Domain Name Service (DNS) assignment

- WINS resolution assignment

Note also that support for Xauth among vendors is somewhat limited because of Xauth's status as a draft-only standard.

Configuration

The Xauth authentication method for either client-side Xauth or server-side Xauth can be selected in the Authentication dialog in the Tunnel Workshop. If client-side Xauth is selected, the Authentication dialog also accepts the Xauth username and password information for the security association.

This Authentication dialog is the preferred method for entering client-side Xauth username and password information, since the Tunnel Workshop will automatically encrypt the password before storing it for security.
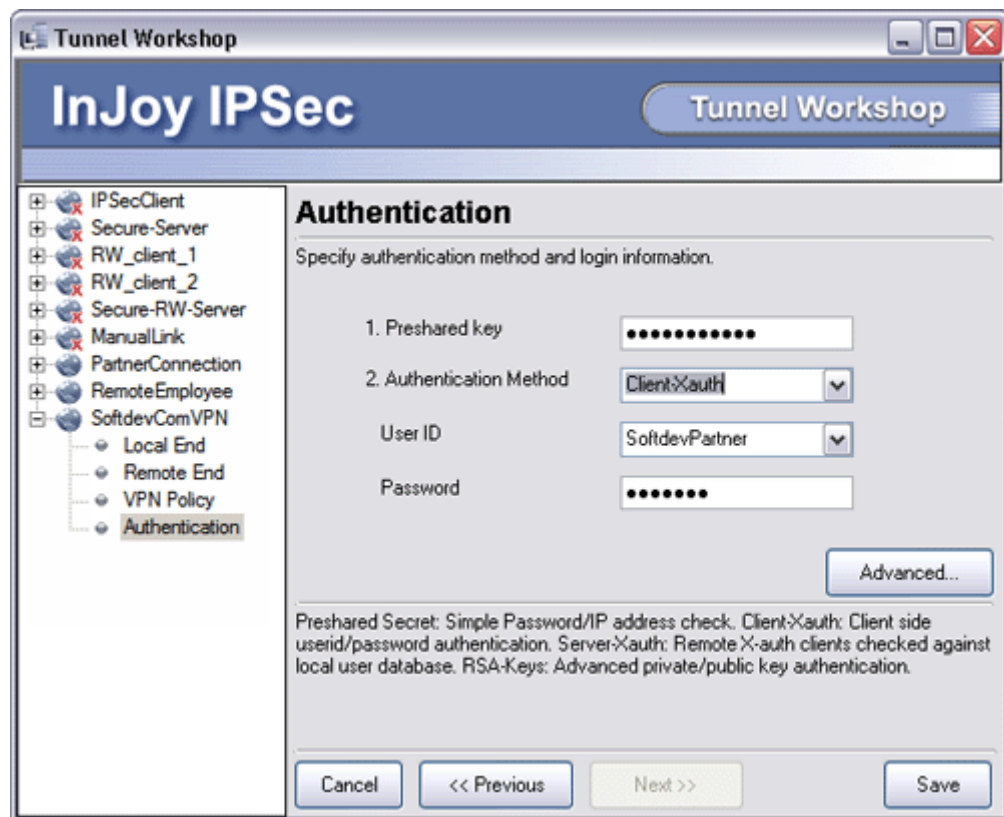
For general information on starting and using the Tunnel Workshop to configure Xauth for security associations, please refer to Section 5, "Configuration".

Xauth can also be configured directly in the **ipsec\ipsec.cnf** file using the Auth-Type keyword and specifying Client-Xauth for client-side Xauth or Server-Xauth for server-side Xauth, respectively. Username and password pairs for client-side Xauth are stored in the **ipsec\ipsec.cnf** configuration file as well:

```
IPSecClient    Description = "Road Warrior Xauth VPN client"
               Local-IP = "0.0.0.0,
               Remote-IP = "vpn.mycompany.com",
               ESP = 3DES,
               Auth-Type = Client-Xauth,
               User-Id = "User",
               Password = "P4ssword",
               Preshared-Secret = "-331188189f15db57b068ab278708"
```

For server-side Xauth, username and password pairs do not appear in the **ipsec\ipsec.cnf** SA. Instead, enter valid Xauth username and password pairs in the **ipsec\vpn-auth.cnf** file using the following format:

```
User           Password = " P4ssword",
               Inner-IP="inner.ip.address.here"
```

In each entry, replace User with the username and P4ssword with the password for the user in question. If applicable, supply an Inner-IP address for the user using the Inner-IP keyword, or use empty quotes ("") to indicate that no Inner-IP address should be assigned.

If the User-ID a Client-Xauth security association is set to "prompt", the user of the host will be prompted to supply this information at connect time.
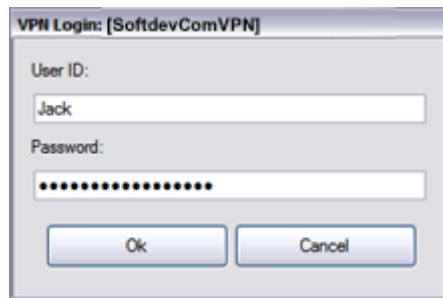
This is the recommended configuration in high-security situations, since it removes the need to store User-ID or password information in text files.

Note: the User-ID and password prompting is a feature specific to InJoy and it will not work with third-party IPSec solutions.

Encrypting Xauth Passwords for Storage
Though Xauth passwords can be stored in **ipsec\ipsec.cnf** or **ipsec\vpn-auth.cnf** in cleartext, as is shown in the example above, it is strongly recommended that you encrypt Xauth passwords after saving them in configuration files. This can be done using the ipsec utility program, supplying -password as an option, followed by the username, the desired password, and the name of the configuration file as arguments:

        ipsec.exe -password User P4ssword vpn-auth.cnf

The command will replace the associated cleartext password in the configuration file with an encrypted version of the same password for more secure storage.

To generate a password to be shown on the screen, use these arguments:

        ipsec.exe -password "" P4ssword

16.3. RSA Digital Signatures

The Rivest Shamir Adleman Digital Signature Standard (RSA DSS) authenticates hosts using public and private encryption key pairs.

Introduction

In order to authenticate itself using RSA DSS, a host must be able to decrypt (using its private encryption key) a message which has been encrypted (using its public encryption key) and then sent by the other endpoint.

In order for a connection to be negotiated, both hosts must be able to authenticate themselves to the other endpoint, meaning that each host must have the other host's public key.

This authentication technique is more secure than Pre-shared Key or Xauth authentication methods because the secret information a host must use to authenticate itself (its private encryption key) never needs to be communicated to any other host or individual.

Unlike the "secrets" used in Pre-shared Keys or Xauth, a host's private encryption key is indeed a secret—it is known only to the host itself. A host's public encryption key, on the other hand, can be circulated to any other necessary endpoint without needing to rely on the propriety of the endpoint for maintaining security.

Because of its flexible key-based architecture, RSA DSS authentication is better able to support multiple network identities than other authentication methods.

Limitations

A few IPSec vendors may not support RSA DSS authentication.

Configuration

Configuration for the use of RSA DSS for authentication between two hosts involves the following steps:

1 Generating public and private keys for each host.

2 Exchanging public keys between hosts.

3 Inserting each remote host's public key into the SA in **ipsec\ipsec.cnf**.

4 Inserting each host's private key into the IKE Server's PLUTO.SECRETS file.

Because these steps are somewhat involved, they are documented in detail in the following sections.

Generating Key Pairs

On each host, a public and private RSA key pair must be generated. This is done using the rsasigkey command, which accepts the number of signature bits as an argument. A full discussion of RSA DSS technology is beyond the scope of this document; 1024 represents a good starting value for keys of this type.

To generate a 1024-bit signature and save it to a file called rsakey.txt, issue the following command:

        rsasigkey.exe 1024 > rsakey.txt

This file will contain a number of comment lines and a large amount of data, and will be formatted as follows:

```
# RSA 1024 bits   localhost   Tue Mar  4 12:24:34 2003
# for signatures only, UNSAFE FOR ENCRYPTION
#pubkey=0sAq0JftKCCixNBok62JPKDp+zIR/R9SqmQlmpBi…
#IN KEY 0x4200 4 1 AQ0JftKCCIxNBok62JPKDp+zIR/R9Sq…
# (0x4200 = auth-only host-level, 4 = IPSec, 1 = RSA)
Modulus: 0x897ed282088c4d06893ad893ca0e9fb3211fd1f…
PublicExponent: 0x03
# everything after this point is secret
PrivateExponent: 0x16ea786b016cb78116df2418a1ad1a9d…
Prime1: 0xeb68c0fbc15ac4aad9f12aa635a0faa2f5afb26073…
Prime2: 0x958599132dc8fba2ad1695db77a7d041d280dedf…
Exponent1: 0x9cf080a7d63c831c914b751c423c0a7174e75…
Exponent2: 0x63ae660cc930a7c1c8b9b93cfa6fe02be1ab3f…
Coefficient: 0x6f977d547341741b37ccc285e47aac46081cdc…
```

Each of the lines which ends in an ellipsis (…) above will actually be quite long in the generated signature file(s) you create using rsasigkey.

Everything after the equal sign on the line that begins with "#pubkey=" is the host's public key. The part of the public used by InJoy IPSec is marked green in the above figure.

All nine lines beginning with the line that starts with "Modulus:" is the host's private key. These lines are marked red.

Exchanging Keys
Once you have generated a public key for each of the hosts involved, you can exchange those public keys using normal means (including email) because it is only the private key which must remain secure for authentication.

Given two hosts, Host A and Host B, host A must sent its public key to Host B, and Host B must sent its public key to Host A, so that these public keys can be inserted into each host's **ipsec\ipsec.cnf** file.

Security Association and Public Key Configuration
You can configure an SA to use RSA DSS by selecting RSA-Keys authentication in the Authentication Dialog in the Tunnel Workshop.

1. Preshared key	
2. Authentication Method	RSA-Keys

You can also enter the remote host's public key in the Authentication—Advanced dialog, opened by clicking on the Advanced button in the Authentication dialog.
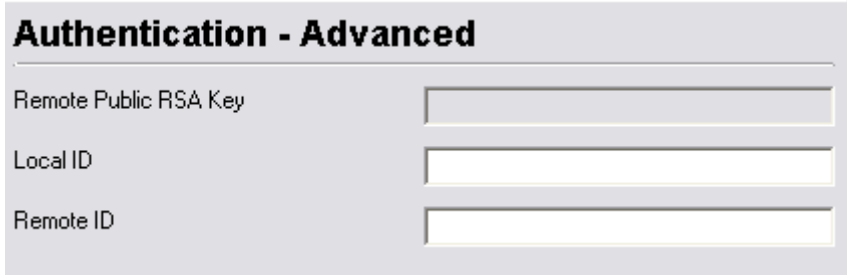
Authentication - Advanced

Remote Public RSA Key	
Local ID	
Remote ID	

For general information on starting and using the Tunnel Workshop please refer to Section 5, "Configuration".

To enable RSA DSS and insert a remote host's RSA public key directly into a security association in the **ipsec\ipsec.cnf** file, set Auth-Type to RSA-Keys, then use the Remote-Public-Key keyword for the public key, taking care to enclose the entire key in quotes. For example:

```
IPSecClient   Description = "Road Warrior RSA VPN client"
              Local-IP = "0.0.0.0",
              Remote-IP = "vpn.mycompany.com",
              ESP = 3DES,
              Auth-Type = RSA-Keys,
              Remote-Public-Key = "<insert key here>",
```

Private Key Configuration
After you have entered the remote host public keys into the **ipsec\ipsec.cnf** file, you must enter the local private keys on each host into the Pluto.secrets file, using the following format (notice indentation):

```
host [host…]: RSA
        {
                <insert key here>
        }
```

The word "host" should be replaced by any number of host identities, each of which can be listed either by IP address, by domain name or by e-mail address. These are the identities (see next section) to which this RSA private key will apply.

Domain names must be prefixed by an @ (at) symbol, allowing them to be matched as text against the fully qualified domain name of the host. A particular user can be specified as well, using the e-mail address (user@domain.com) format. Plain domain names without the @ symbol will not be resolved to IP addresses and will be treated as erroneous text.

For example, for a pair of hosts with 10.1.1.1 and vpn.softdev.com identities, one possible entry would read:

```
10.1.1.1 @vpn.softdev.com: RSA
        {
                <key appears here>
        }
```

Note that a single host may have a number of private keys matching a number of disparate identities.

RSA DSS and Identities
During IKE negotiation, each endpoint has two identities—the local identity (the label under which a host identifies itself) and the remote identity (the way it is perceived by the remote host). When RSA DSS authentication takes place, each endpoint searches its list of private keys for a key that matches

both of these identities. If found, it is this private key which will be used for authentication.

The default values for both the local and remote identities are the IP address of the host in question. However, the InJoy software includes management tools for authentication based on multiple identities.

Two security association keywords, Local-ID and Remote-ID are used to explicitly specify the ID of the local host, or the ID of the remote host, respectively, if an identity other than a host IP address is desired. Local-ID of the IPSec Client must match the Remote-ID of the IPSec Server (and vice verse). For example, consider the following security associations.

On the local host:

```
IPSecClient   Description = "Road Warrior RSA VPN client"
              Local-IP = "192.168.0.2",
              Remote-IP = "192.168.0.1",
              Auth-Type = RSA-Keys,
              Remote-Public-Key = "<insert key here>",
              Remote-ID = "@server.ournet.com",
```

On the remote host:

```
IPSecServer   Description = "RSA VPN server"
              Local-IP = "192.168.0.1",
              Remote-IP = "192.168.0.2",
              ESP = 3DES,
              Auth-Type = RSA-Keys,
              Remote-Public-Key = "<insert key here>",
              Local-ID = "@server.ournet.com",
```

For authentication to occur between these two hosts, the following must both be true:

- Because no identity has been specified for the host 192.168.0.2, a private key of IPSecClient must be saved in this host's PLUTO.SECRETS file under the host identities "192.168.0.2" and "@server.ournet.com". This key must be a match for the public key stored in the IPSecServer security association on the host 192.168.0.1.

- A private key of IPSecServer must be saved in the PLUTO.SECRETS file on the host 192.168.0.1 under the same identities ("192.168.0.2" and "@server.ournet.com"). This key must be a match for the public key stored in the IPSecClient security association on the other endpoint.

By working easily with multiple identities, RSA DSS adds unmatched flexibility to host-based network authentication while at the same time remaining both robust and secure.

16.4. Group Authentication

Group authentication works together with Xauth authentication, providing additional layer of security and great configuration possibilities for VPN Servers.

Introduction

Group authentication implements the technique known as two-factor authentication, where the user's VPN Client contains pre-filled group name and password and user is prompted user name and password each time before connecting to VPN Server.

Group authentication is often used on VPN Servers to uniquely determine the group of users that the traffic has originated from and apply relevant access level and rights.

Limitations

Group authentication is supported in client mode only.

Configuration

To make use of Group authentication, you need to know the following information (examples are listed in parenthesis):

- Group name (sup3rgr0up)
- Group password (gr0uppwd)
- User name (sup3ru5er)
- User password (u5erpwd)

The exact configuration depends on the type of the VPN Server, but the common guidelines are listed here (which fully apply to Cisco VPN3000 equipment):

1 The authorization, authentication and identification part of the SA must be as follows:

```
Auth-Type = Unused,
Authorization = XauthV6_Cli,
Identification = IdKeyId,
Authentication = PSK,
```

2 Put group name and group password into Local-ID and Preshared-Secret fields, respectively:

```
Local-ID = "sup3rgr0up",
Preshared-Secret = "gr0uppwd"
```

3 Put user name and user password into User-ID and Password fields, respectively:

```
User-Id = "sup3ru5er",
Password = "u5erpwd",
```

4 Set up other SA fields according to VPN Server configuration.

An example of working and ready-to-use Security Association for Cisco VPN3000:

```
VPN3000          Description = "Cisco VPN3000",
                 Local-IP = "0.0.0.0",
                 Local-Net = "192.168.4.0",
                 Local-Mask = "255.255.255.0",
                 Remote-IP = "16.17.18.19",
                 Remote-Net = "0.0.0.0",
                 Remote-Mask = "0.0.0.0",
                 Auth-Type = Unused,
                 Authorization = XauthV6_Cli,
                 Identification = IdKeyId,
                 Authentication = PSK,
                 User-Id = "sup3ru5er",
                 Password = "u5erpwd",
                 Local-ID = "sup3rgr0up",
                 Preshared-Secret = "gr0uppwd"
                 Cisco-Delete = Yes,
                 Aggressive = Yes,
                 Aggressive-Oakley = "AES;SHA;PK;DH2",
```

16.5. X.509 Certificates

The purpose of this section is to provide a basic introduction to X.509 (RFC 2459) authentication, along with working examples. It should be noted that X.509 is a flexible feature that can be deployed in an endless number of combinations, most of which are beyond the scope of this document.

Introduction

A digital certificate is a "package" that contains a set of attributes (e.g. issuer name, lifetime, etc) and a public key. Certificates are created and digitally signed by a CA (Certificate Authority) – typically a software application that encrypts the certificate.

X.509 uses essentially the same encryption scheme as raw RSA keys, however with additional containers (the actual certificate files). When you have a public key of a CA, you can use it to validate the digital signature (the identity) of the certificate owner.

In order to use a certificate as a valid digital signature, a user needs to have both a public and a private key. The certificate only contains the public key. The public key however cannot be used to create a digital signature. This is the reason why anyone is allowed to know the certificate and why the certificate is public.

Each certificate (read: public key) has an accompanying private key, which is generated at the Certificate Signing Request (i.e. when the certificate is first issued by the CA). The private key is used in conjunction with the public key to establish a secure environment during IKE negotiations. Only the user who owns the certificate must know the private key.

The representation of an X.509 certificate is a file (text or binary), and it can be encoded in many different ways: PEM, base64, pkcs12, etc.

The X.509 files allow a certificate trust-inheritance scheme, where one certificate is used to sign another. Also, the certificate files themselves contain useful auxiliary information (e.g. company name, company department, and the user's e-mail address), which helps to strengthen the overall security.

The X.509 trust-inheritance scheme begins with a topmost self-signed certificate – the Certificate Authority (a root certificate that is implicitly considered valid).

To handle certificates which get compromised or stolen, a technique exists of declaring a certificate invalid. The technique is generally referred to as the Certificate Revocation List (CRL), on which any given certificate can be revoked by the administrator.

X.509 in IPSec

X.509 support enables IPSec to handle certificate authentication, similar to RSA-Key authentication.

With X.509 public key is stored in the certificate, while with RSA authentication the public key is stored directly in the IPSec configuration file.

The X.509 certificate is sent via the IKE protocol during Phase 1 negotiation, allowing IPSec to use the public key of the certificate to validate the user's digital signature.

In IPSec, X.509 can be used with certificates stored in PEM (encoded by base64) or DER (binary) formats only (IPSec auto-senses the type). Also, passphrases for private keys are not prompted for, but instead stored in configuration files.

An implication of X.509 in IPSec is its impact on the network MTU during the first part of IKE negotiations. The Maximum datagram size (a.k.a. MTU) of the Ethernet protocol is 1500 bytes, and since the IKE server employs UDP for exchanging certificates, the administrator has to ensure that the size of individual certificates do not grow unacceptably big and thus exceed the Ethernet packet limit (of 1500 bytes). One method of making sure that a certificate stays within the realistically manageable size is to not use big key lengths (e.g. more than ~6000 bits). Long keys are likely to result in lost packets and failed authentication, if no general MTU handling procedure for IPSec is in place.

X.509 in IPSec introduces no additional requirements. However, if you generate certificates on your own, respective third-party software is required, such as OpenSSL.

Deployment Steps

The **first step** in deploying X.509 is to set up a CA server to generate and sign the certificates. The CA server software will typically offer command line tools or publish a website where users can make "certificate requests". A

certificate request contains data, such as the user name, department, etc. The administrator of the CA server can accept or refuse the certificate request. In many cases, the users making online certificate requests will additionally be authenticated with a phone call or similar.

As a **second step**, the IPSec administrator should decide which other IPSec authentication methods X.509 should be combined with. For example, X.509 can be combined with X-Auth or X-Auth v6. Note, Pre-shared Secrets (PSK), RSA Signatures and X.509 are mutually exclusive.

As a **third step** in deploying X.509, the administrator should consider how certificates are managed. I.e. how the process of generation, distribution and revocation is streamlined for the business.

X.509 is a flexible feature, so the above steps are very basic guidelines. For larger installations it is highly recommended that X.509 is studied in full and that capable third party software is installed for key management.

In the following sections, the OpenSSL software is used on the command line to generate and sign certificates.

Quick Getting Started

This section provides the shortest possible path to establishing a tunnel using X.509 authentication:

On **both end-points**, do the following.

1 Put the self-signed root certificate in the ipsec/ipsec.d/cacerts/ directory.

2 Put the end-point's certificate (signed by the CA) in the ipsec/ipsec.d/certs/ directory and fill in the Local-Certificate attribute in ipsec/ipsec.cnf along with the additional attributes relevant to X.509 (highlighted in red, below):

```
aaa-cert
        Local-IP = "my_ip",
        Remote-IP = "2001:268:204:7900::1",
        IPv6 = Yes,
        Reinit = Yes,
        Auth-Type = Unused,
        Authentication = X509,
        Identification = DERASN1,
        Authorization = None,
        Local-Certificate = "libra.crt",
```

3 Put the end-point's private key into the ipsec/ipsec.d/private/ directory and point IPSec to use it by adding the following line to the pluto.sec file:

    : RSA libra.key

Where libra.key is a file that contains the end-point's private key.

As needed, update the CRL file in the ipsec/ipsec.d/crls/ directory to deny authentication attempts with revoked certificates.

Generating Certificates

In this example, setting up the tunnel with X.509 authentication includes:

- Obtaining the OpenSSL software for generating the certificates
- Generating the certificates
- Editing configuration files to use the certificates.

First, obtain a copy of OpenSSL at http://www.openssl.org/ and then set it up it according to installation instructions included in the package.

Now you are ready to create the self-signed root Certificate Authority (CA), which will be used to both sign client certificates and authenticate them:

```
openssl req -new -newkey rsa:1024 -nodes -keyout ca.key -x509 -days 730 -out ca.crt
```

Where:

- `req` Is a request to create a new certificate
- `-new` Means create a Certificate Signing Request (CSR)
- `-newkey rsa:1024` Means to create a new private key of 1024 bit length. You can customize the key length, but Pluto accepts keys no larger than 8192 bits
- `-nodes` Means not to encrypt the private key
- `-keyout ca.key` Means save the private key into ca.key
- `-x509` Means that itstead of doing a CSR, it creates a self-signed certificate
- `-days 730` Certificate validity is 730 days (2 years). You can customize this value, but don't set too small value, since you will be signing client certificates using this CA
- `-out ca.crt` Save the certificate into ca.crt

OpenSSL will ask you some details about the certificate:

```
Generating a 1024 bit RSA private key
......+++++
...........+++++
writing new private key to 'ca.key'
-----
You are about to be asked to enter information that will be incorporated
into your certificate request.
What you are about to enter is what is called a Distinguished Name or a DN.
There are quite a few fields but you can leave some blank
For some fields there will be a default value,
If you enter '.', the field will be left blank.
```

```
-----
Country Name (2 letter code) [GB]:RU
State or Province Name (full name) [Berkshire]:Astrakhan
Locality Name (eg, city) [Newbury]:Astrakhan
Organization Name (eg, company) [My Company Ltd]:FX Communications
Organizational Unit Name (eg, section) []:Software Development
Common Name (eg, your name or your server's hostname) []:ca-server
Email Address []:info@fx.dk
```

Don't use special characters when entering details, e.g. don't use the **/** (slash) character, as it will conflict with the Distinguished Name (DN) syntax.

As a result of running this command, **ca.key** and **ca.crt** will appear. Copy **ca.crt** into **ipsec/ipsec.d/cacerts** at both tunnel endpoints.

As a step two, create client certificates. First, create **openssl.conf** with the following contents:

```
[ ca ]
default_ca              = CA_CLIENT
# When signing certificates, use CA_CLIENT section
#

[ CA_CLIENT ]
dir                     = ./certdb
# Directory which will hold certificate database maintained by
# OpenSSL
#
certs                   = $dir/certs
# Certificates directory
#
new_certs_dir           = $dir/newcerts
# New certificates directory
#
database                = $dir/index.txt
# Database of signed certificates
#
serial                  = $dir/serial
# File containing serial number of last issued certificate
#
certificate             = ./ca.crt
# CA file
#
private_key             = ./ca.key
# Private key of the CA
#
default_days            = 365
# Validity period of client certificate to be issued
#
default_crl_days        = 7
# CRL validity period
#
default_md              = md5
# Hash function algorithm
#
policy                  = policy_anything
# Name of the certificate policy section
#
```

```
      [ policy_anything ]
      countryName            = optional
      # Country code - optional
      #
      stateOrProvinceName    = optional
      localityName           = optional
      organizationName       = optional
      organizationalUnitName = optional
      commonName             = supplied
      # Common Name (typically hostname) is mandatory
      #
      emailAddress           = optional
```

Now create the database structure for OpenSSL:

```
mkdir cert_db
mkdir cert_db/certs
mkdir cert_db/newcerts
touch cert_db/index.txt
echo "01" > cert_db/serial
```

The last two commands generate an empty **cert_db/index.txt** and **cert_db/serial** with just "01" inside them (without quotes).

Next, do a Certificate Signing Request for the client certificate. OpenSSL parameters are same as for the CA's CSR, however, the "-x509" parameter is now omitted.

```
openssl req -new -newkey rsa:1024 -nodes -keyout libra.key -out libra.csr
```

```
      Generating a 1024 bit RSA private key
      ....................+++++
      ..............................................................+++++
      writing new private key to 'libra.key'
      -----
      You are about to be asked to enter information that will be incorporated
      into your certificate request.
      What you are about to enter is what is called a Distinguished Name or a DN.
      There are quite a few fields but you can leave some blank
      For some fields there will be a default value,
      If you enter '.', the field will be left blank.
      -----
      Country Name (2 letter code) [GB]:RU
      State or Province Name (full name) [Berkshire]:Astrakhan
      Locality Name (eg, city) [Newbury]:Astrakhan
      Organization Name (eg, company) [My Company Ltd]:FX Communications
      Organizational Unit Name (eg, section) []:Software Development
      Common Name (eg, your name or your server's hostname) []:libra
      Email Address []:libra@fx.dk

      Please enter the following 'extra' attributes
      to be sent with your certificate request
      A challenge password []:
      An optional company name []:
```

libra.key will hold the private key, and **libra.csr** is the CSR which will be signed by the CA and result in a certificate file later.

Next, sign the CSR by the CA:

```
openssl ca -config openssl.conf -in libra.csr -out libra.crt –batch
```

```
Using configuration from openssl.conf
Check that the request matches the signature
Signature ok
The Subject's Distinguished Name is as follows
countryName           :PRINTABLE:'RU'
stateOrProvinceName   :PRINTABLE:'Astrakhan'
localityName          :PRINTABLE:'Astrakhan'
organizationName      :PRINTABLE:'FX Communications'
organizationalUnitName:PRINTABLE:'Software Development'
commonName            :PRINTABLE:'libra'
emailAddress          :IA5STRING:'libra@fx.dk'
Certificate is to be certified until Dec  2 16:58:22 2006 GMT (365 days)


Write out database with 1 new entries
Data Base Updated
```

Now, **libra.crt** and **libra.key** are what will be needed to establish a tunnel using X.509 authentication. Repeat these 2 steps – creating the CSR and signing it - for each endpoint of the tunnel. The above showed how the client certificate was created, so, for the server, another certificate is created:

```
openssl req -new -newkey rsa:1024 -nodes -keyout libresse.key -out libresse.csr
openssl ca –config openssl.conf -in libresse.csr -out libresse.crt -batch
```

As a final step, copy **libra.crt** to **ipsec/ipsec.d/certs** and copy **libra.key** to **ipsec/ipsec.d/private**. Do the same on the server with **libresse.crt** and **libresse.key**.

Configuring IPSec

To configure IPSec to use certificates, point it to the certificate file:

```
Test-Cert
   Local-IP = "my_ip",
   Remote-IP = "10.11.31.2",
   ReInit = No,
   Auth-Type = Unused,
   Authentication = X509,
   Identification = DERASN1,
   Authorization = None,
   Local-Certificate = "libra.crt",
```

The relevant attributes here are:

- `Auth-Type` – "Unused" means that the Authentication, Identification and Authorization attributes are used to govern how the tunnel is established.

- `Authentication` - "X509" states that X.509 certificates will be used.

- `Identification` - Distinguished Name (DER ASN.1 DN) will be used to identify end-points.

- `Authorization` - No additional authorization will be used. Can be modified to be XAUTH or any other supported type.

- `Local-Certificate` – The file that contains the certificate of the endpoint, in binary DER or base64 PEM format (the type is sensed automatically by IPSec). This certificate must reside in the **ipsec/ipsec.d/certs** directory.

Now point IPSec to the private key of the endpoint, by adding the following line to **pluto.sec**:

```
: RSA libra.key
```

After you set up the opposite endpoint in the same manner (except for the Reinit parameter), the tunnel is ready to be established.

Certificate Revocation List

By putting a CA file into the **ipsec/ipsec.d/cacerts**, all certificates signed by this CA, are automatically declared valid, which is not always desirable.

If an administrator needs to declare one or more certificates invalid, for example, when the certificate is compromised or a user leaves the company, the Certificate Revocation List (CRL) should be used.

The CRL is basically a file with a list of invalid certificates. This file must be updated on a regular basis to reflect the revoked certificates.

To create a CRL, use the following OpenSSL command:

```
openssl ca -gencrl -config openssl.conf -out ca.crl
```

Put the resulting **ca.crl** into the directory **ipsec/ipsec.d/crls**.

To revoke a certificate, use:

```
openssl ca -config openssl.conf -revoke libra.crt
```

And update **ca.crl** on the server.

Part V
Deployment Examples

17 More Sample Scenarios

This section is designed to provide examples of common situations and VPN configurations, in order to help you as you deploy your own VPN.

Each sample scenario contains:

- A description of the VPN organization

- A diagram showing how hosts on this VPN interact with one another and the public Internet

- Sections from the relevant configuration files, so that you can use these examples in your own VPN deployment

If you haven't already, you may also wish to read Section 10, "A VPN Case Study," which provides an additional sample scenario and goes into greater depth than the scenarios in this section do.

17.1. Simple VPN Using Manual Keying

There may be times when you need to quickly create the simplest VPN possible:

- Two stand-alone hosts

- The simplest IP address authentication

- A minimum of necessary software components (i.e. no Pluto IKE Server needed)

- Basic 3DES encryption and MD5 authentication

Though not necessarily the most secure option available, a two-host, manually keyed VPN like this one can be created quickly and easily in situations when temporary, somewhat secure communication across the public Internet is required.

Note: It is not possible to configure the Manual Keys in the IPSec Tunnel Workshop GUI - ASCII configuration files must be edited in order to set up this configuration.

VPN Scenario Details

In this scenario there are two hosts with static IP addresses. Both PCs run the InJoy software and none of the PCs run the Pluto IKE Server. Only the IP addresses of the VPN hosts authenticate the identities of the hosts.

Host A
217.126.17.44

Host B
211.56.33.12
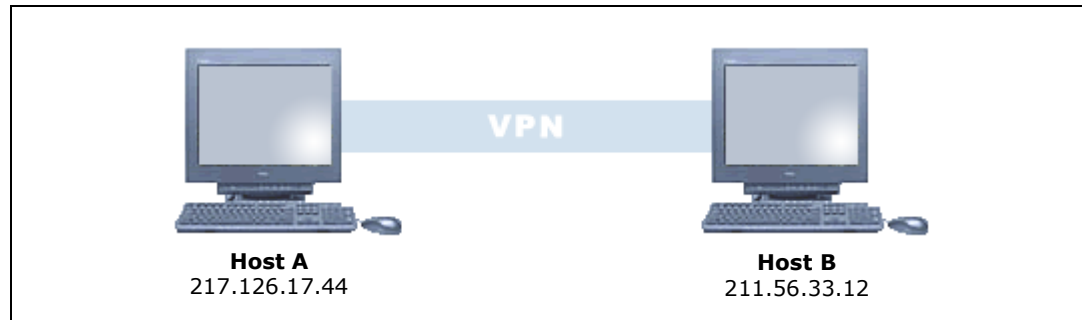
Configuration File Details

The configuration for this scenario takes place in security associations in the hosts' **ipsec\ipsec.cnf** configuration files.

The security association on host A:

```
Host_A  Local-IP = "My_IP",
        Remote-IP = "211.56.33.12",
        AH = No,
        ESP = Yes,
        Remote-ESP-Key =
"00FFEEDDCCBBAA00112233445566778899FFEEDDCCBBAAFFEEDDCCAA99887766554433221100FFAA",
        ESP-Key =
"00112233445566778899AABBCCDDEEFF00112233445566770011223344556677889 9AABBCCDDEEFF",
        ESP-Transmit-SPI = 512,
        ESP-Receive-SPI = 513,"
```

The security association on host B:

```
Host_B  Local-IP = "My_IP",
        Remote-IP = "217.126.17.44",
        AH = No,
        ESP = Yes,
        ESP-Key =
"00FFEEDDCCBBAA00112233445566778899FFEEDDCCBBAAFFEEDDCCAA99887766554433221100FFAA",
        Remote-ESP-Key =
"00112233445566778899AABBCCDDEEFF00112233445566770011223344556677889 9AABBCCDDEEFF",
        ESP-Receive-SPI = 512,
        ESP-Transmit-SPI = 513,"
```

Formatting constrains in the above examples cause the ESP keys to be wrapped around and take up two lines each. As you deploy these examples, please make sure to undo this effect, by putting both the configuration attribute and its value on the same physical line.

The Remote-IP addresses used in the examples must be updated to reflect the actual IP addresses of your network configuration.

Pay particular attention to the way that the manual keying values on Host B are mirror images to those on Host A—Host A's ESP-Receive-SPI is Host B's ESP-Transmit-SPI, and so on. This is necessary for manual keying to function properly. For more details on using manual keying, please refer to Section 15, "Using Manual Keying."

17.2. Simple VPN Using Automatic Keying

Though the Manual Keying scenario was simple and did not need the Pluto IKE Server in order to function, it was somewhat insecure compared to typical IPSec configurations. The security of the IPSec connection can be increased many times simply by modifying the scenario a little bit:

- Two stand-alone hosts

- The simplest host-based authentication available

- Both hosts using an IKE Server

- 3DES encryption and packet-level authentication that can be negotiated between the hosts

With respect to the SAs which must be created, this scenario is actually easier to duplicate than the previous one, because it obviates the need for long manual keying entries in the host configuration files.

VPN Scenario Details

Two hosts with static IP addresses, both running the InJoy software, both also running the Pluto IKE Server for Internet Key Exchange. Pre-shared Key authentication is used to verify the identities of the hosts.
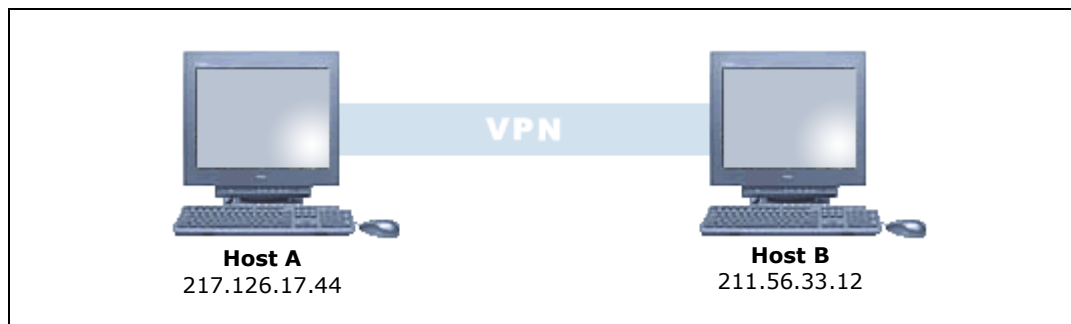
Host A
217.126.17.44

Host B
211.56.33.12

All of the configuration details shown below can also be entered using the Tunnel Workshop. For details on using the Tunnel Workshop, please refer to Section 5, "Configuration".

Configuration File Details

The configuration for this scenario also takes place in security associations in the hosts' **ipsec\ipsec.cnf** configuration files.

The security association on host A:

```
Host-A
      Local-IP = "My_IP",
```

```
        Remote-IP = "211.56.33.12",
        Preshared-Secret = "testpwd",
        Reinit = Yes,
```

The security association on host B:

```
Host-B
        Local-IP = "My_IP",
        Remote-IP = "217.126.17.44",
        Preshared-Secret = "testpwd",
        Reinit = Yes,
```

In addition to making the IPSec connection much more secure, the presence of the Pluto IKE server has helped to simplify the entries in **ipsec\ipsec.cnf** considerably, when compared to the previous scenario.

17.3. VPN With Multiple Sub-Networks

Though the two host scenarios presented are instructive and may also at times be useful, in the real world, VPNs are often more complex than simple two-host configurations. This example demonstrates:

- More than two VPN hosts or gateways

- A sub-network behind a gateway and multiple sub-networks behind a gateway

- Negotiation for 3DES encryption and packet-level authentication (MD5)

In spite of the added topological complexity of the network, these types of scenarios are not much more difficult to configure than the simple two-host scenarios presented in previous sections.

VPN Scenario Details

For this scenario, we will imagine a hypothetical company called Widgets, Inc. that has the following needs:

- The company has three offices, each located at great physical distance from the others—and they must be able to communicate securely over the public Internet

- Office A, San Jose, has a gateway at 24.248.36.1 and will route traffic for internal sub-networks 10.1.0.0 and 10.4.0.0.

- Office B, London, has a gateway at 12.254.84.24 and will route traffic for internal sub-network 10.2.0.0.

- Office C, Tokyo, has a gateway at 136.74.82.167 and will route traffic for internal sub-network 10.3.0.0.

- For the deployment trial run, Widgets, inc. wants to use Pre-shared Key authentication

San Jose

London   **Tokyo**
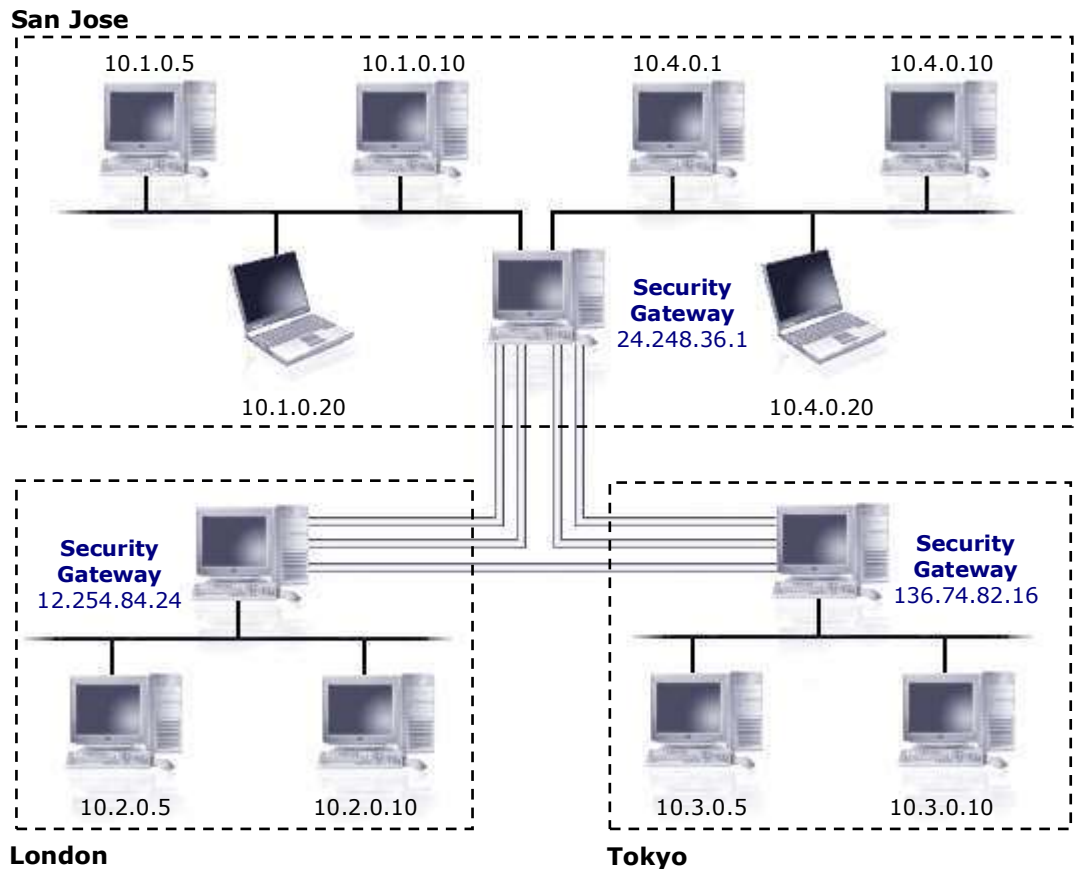
Naturally, each office is running the InJoy software and the Pluto IKE Server for the creation of this VPN. For this configuration to work, a greater number of SAs need to be created.

It can help to think of each IPSec connection as a pipe; two network segments must be connected by such a pipe in order to communicate.

As was the case in the previous scenario, all of the configuration details shown below can also be entered using the Tunnel Workshop. For details on using the Tunnel Workshop, please refer to Section 5, "Configuration".

To be able to establish two or more IPSec tunnels to the same remote host, you must manually enable the Nested-SA-Bundles option in **ipsec\options.cnf** – at both involved IPSec end-points:

```
Options      Trace-AH = No,
             Trace-ESP = No,
             Trace-Tunnel = No,
             Trace-Frag = No,
             Trace-Packets = No,
             Dump-Packets = No,
             Log-Level = Info,
             Start-IKE-Server = Daemon,
             Nested-SA-Bundles = Yes,
```

The Nested-SA-Bundles ensures that Gateway traffic matching multiple SA is properly encrypted. Also notice, the order of SAs must be exactly the same on both sides.

Configuration File Details (San Jose)

Both networks in San Jose need to be able to reach network 10.3.0.0 (12.254.84.24) in London and network 10.4.0.0 (136.74.82.167) in Tokyo. The network administrator in San Jose creates the following security associations in **ipsec\ipsec.cnf**:

```
To_London_SA_1              Description = "10.1.0.0 <-> 10.2.0.0"
                            Local-IP = "My_IP",
                            Local-Net = "10.1.0.0",
                            Local-Mask = "255.255.0.0",
                            Remote-IP = "12.254.84.24",
                            Remote-Net = "10.2.0.0",
                            Remote-Mask = "255.255.0.0",
                            AH = Yes,
                            ESP = Yes,
                            Preshared-Secret = "-3c0f9d1782099c5a"
```

```
To_London_SA_2              Description = "10.4.0.0 <-> 10.2.0.0"
                            Local-IP = "My_IP",
                            Local-Net = "10.4.0.0",
                            Local-Mask = "255.255.0.0",
                            Remote-IP = "12.254.84.24",
                            Remote-Net = "10.2.0.0",
                            Remote-Mask = "255.255.0.0",
                            AH = Yes,
                            ESP = Yes,
                            Preshared-Secret = "-3c0f9d1782099c5a"
```

```
To_Tokyo_SA_1               Description = "10.1.0.0 <-> 10.3.0.0"
                            Local-IP = "My_IP",
                            Local-Net = "10.1.0.0",
                            Local-Mask = "255.255.0.0",
                            Remote-IP = "136.74.82.167",
                            Remote-Net = "10.3.0.0",
                            Remote-Mask = "255.255.0.0",
                            AH = Yes,
                            ESP = Yes,
                            Preshared-Secret = "-3c0f9d1782099c5a"
```

```
To_Tokyo_SA_2               Description = "10.4.0.0 <-> 10.3.0.0"
                            Local-IP = "My_IP",
                            Local-Net = "10.4.0.0",
                            Local-Mask = "255.255.0.0",
                            Remote-IP = "136.74.82.167",
                            Remote-Net = "10.3.0.0",
                            Remote-Mask = "255.255.0.0",
                            AH = Yes,
                            ESP = Yes,
                            Preshared-Secret = "-3c0f9d1782099c5a"
```

Notice that there are two security associations for each of the other offices. This is because San Jose hosts two networks, 10.1.0.0 and 10.4.0.0; each of these must have its own SA.

Configuration File Details (London)

The network in London needs to be able to reach networks 10.1.0.0 and 10.4.0.0 in San Jose (both at 24.248.36.1) and network 10.3.0.0 (136.74.82.167) in Tokyo. The network administrator in London creates the following security associations in **ipsec\ipsec.cnf**:

```
To_San_Jose_SA_1          Description = "10.2.0.0 <-> 10.1.0.0"
                          Local-IP = "My_IP",
                          Local-Net = "10.2.0.0",
                          Local-Mask = "255.255.0.0",
                          Remote-IP = "24.248.36.1",
                          Remote-Net = "10.1.0.0",
                          Remote-Mask = "255.255.0.0",
                          AH = Yes,
                          ESP = Yes,
                          Preshared-Secret = "-3c0f9d1782099c5a"
```

```
To_San_Jose_SA_2          Description = "10.2.0.0 <-> 10.4.0.0"
                          Local-IP = "My_IP",
                          Local-Net = "10.2.0.0",
                          Local-Mask = "255.255.0.0",
                          Remote-IP = "24.248.36.1",
                          Remote-Net = "10.4.0.0",
                          Remote-Mask = "255.255.0.0",
                          AH = Yes,
                          ESP = Yes,
                          Preshared-Secret = "-3c0f9d1782099c5a"
```

```
To_Tokyo_SA               Description = "10.2.0.0 <-> 10.3.0.0"
                          Local-IP = "My_IP",
                          Local-Net = "10.2.0.0",
                          Local-Mask = "255.255.0.0",
                          Remote-IP = "136.74.82.167",
                          Remote-Net = "10.3.0.0",
                          Remote-Mask = "255.255.0.0",
                          AH = Yes,
                          ESP = Yes,
                          Preshared-Secret = "-3c0f9d1782099c5a"
```

Notice that there are two security associations for San Jose. This is because San Jose hosts two networks, 10.1.0.0 and 10.4.0.0; each of these must have its own security association.

Configuration File Details (Tokyo)

The network in Tokyo needs to be able to reach networks 10.1.0.0 and 10.4.0.0 in San Jose (both at 24.248.36.1) and network 10.2.0.0 (12.254.84.24) in London. The network administrator in Tokyo creates the following security associations in **ipsec\ipsec.cnf**:

```
Toi_San_Jose_SA_1         Description = "10.3.0.0 <-> 10.1.0.0"
                          Local-IP = "My_IP",
                          Local-Net = "10.3.0.0",
                          Local-Mask = "255.255.0.0",
                          Remote-IP = "24.248.36.1",
                          Remote-Net = "10.1.0.0",
```

```
                              Remote-Mask = "255.255.0.0",
                              AH = Yes,
                              ESP = Yes,
                              Preshared-Secret = "-3c0f9d1782099c5a"


To_San_Jose_SA_2              Description = "10.3.0.0 <-> 10.4.0.0"
                              Local-IP = "My_IP",
                              Local-Net = "10.3.0.0",
                              Local-Mask = "255.255.0.0",
                              Remote-IP = "24.248.36.1",
                              Remote-Net = "10.4.0.0",
                              Remote-Mask = "255.255.0.0",
                              AH = Yes,
                              ESP = Yes,
                              Preshared-Secret = "-3c0f9d1782099c5a"


To_London_SA                  Description = "10.3.0.0 <-> 10.2.0.0"
                              Local-IP = "My_IP",
                              Local-Net = "10.3.0.0",
                              Local-Mask = "255.255.0.0",
                              Remote-IP = "12.254.84.24",
                              Remote-Net = "10.2.0.0",
                              Remote-Mask = "255.255.0.0",
                              AH = Yes,
                              ESP = Yes,
                              Preshared-Secret = "-3c0f9d1782099c5a"
```

As was the case in London, in Tokyo there are also two security associations for San Jose in order to be able to reach both of San Jose's networks, 10.1.0.0 and 10.4.0.0.

VPN Scenario Security Notes

For a network of this size and complexity, simple Pre-shared Key authentication is likely to be inadequate. When Widgets, Inc. officially deploys its VPN, it must ensure that a more robust host-based authentication scheme — such as RSA DSS or X-Authentication — is in place.

17.4. VPN Using RSA DSS Authentication

All of the scenarios covered so far have suffered from one major problem: all of them have used relatively weak Pre-shared Key host authentication. In the real world, many VPN administrators will demand the use of one of the more secure authentication methods discussed in Section 16, "Authentication Method."

When each is considered alone, the most secure of the host authentication methods discussed in Section 16 is RSA DSS authentication, which provides:

- A well-tested, standards-based public/private key authentication method based on the RSA DSS

- Flexibility for managing authentication when single hosts have multiple identities (users) or change identities based on context

- A risk-free way of distributing the information needed for two hosts to authenticate themselves to one another (public keys)

The scenario in this section only demonstrates the configuration process for using RSA DSS with IPSec in action. For information on how RSA DSS works, its advantages and disadvantages, and fundamental documentation for using RSA DSS with the InJoy software, please refer to Section 16.3, "RSA Digital Signatures."

VPN Scenario Details

To illustrate the use of RSA DSS authentication, we'll adopt the scenario from the previous section. Recall that Widgets, Inc. created a VPN with multiple sub-networks at three separate offices—San Jose, London and Tokyo. Widgets, Inc. unified these sub-networks under the RFC-1918 address range 10.0.0.0 using InJoy's client-side Inner-IP support.

There was, however, one shortcoming of the VPN. Only Pre-shared Key authentication was used to authenticate each office's gateway during the VPN deployment trial-run. Widgets, Inc. is now ready to take their network live for real work, but would first like to improve the robustness and security of the host authentication between office gateways. They plan to do this using RSA DSS.

Each of the offices, San Jose, London and Tokyo, will need a number of keys:

- San Jose will need its own private key, plus public keys from London and Tokyo
- London will need its own private key, plus public keys from San Jose and Tokyo
- Tokyo will need its own private key, plus public keys from San Jose and London

After generating the keys using the rsasigkey.exe tool described in Section 16.3, the offices of Widgets, inc. are ready to make the necessary changes to their configuration files, which are shown in their original form in the previous section.

Because the original configuration details from **ipsec\ipsec.cnf** are available in the previous section and are somewhat lengthy, we'll only show the public key additions to the **ipsec\ipsec.cnf** files and the private key additions to the offices' PLUTO.SECRETS here.

Configuration File Details (San Jose)

These lines were added to the security associations London_SA_1 and London_SA_2 in **ipsec\ipsec.cnf**, just before the Preshared-Secret keywords:

```
Auth-Type = RSA-Keys,
Remote-Public-Key = "0sAQNwwO6ObsiU5JbHjwYYnkSk5…",
```

These lines were added to the security associations Tokyo_SA_1 and Tokyo_SA_2 in the same file:

```
Auth-Type = RSA-Keys,
```

```
Remote-Public-Key = "0sAQOSumCFc1oEpXStk0fTlpVLp+…",
```

At the same time, the network administrator in San Jose added the following entry to the San Jose office's pluto.secrets file:

```
24.248.36.1 12.254.84.24 136.74.82.167: RSA
    {
        Modulus: 0x6ee64975cb3ab97b33e1a8579ee06112b3bffb15…
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent: 0x127bb6e8f734743f335046b9452565831d…
        Prime1: 0xb1031cccba461879486d390749a6cc1ccdf182ea56…
        Prime2: 0xa062d66e386ff4710910f09a578c9acaf260463912…
        Exponent1: 0x7602133326d965a63048d0af866f32bddea101f…
        Exponent2: 0x6aec8ef4259ff84b5b60a066e5086731f6ead97…
        Coefficient: 0x0cf5830d402fc7205aec4c039199838e7fe88eb…
    }
```

Configuration File Details (London)

These lines were added to the security associations SanJose_SA_1 and SanJose_SA_2 in **ipsec\ipsec.cnf**, just before the Preshared-Secret keywords:

```
Auth-Type = RSA-Keys,
Remote-Public-Key = "0sAQNu5k11yzq5ezPhqFee4GESs7…",
```

These lines were added to the security association Tokyo_SA in the same file:

```
Auth-Type = RSA-Keys,
Remote-Public-Key = "0sAQOSumCFc1oEpXStk0fTlpVLp+X…",
```

At the same time, the network administrator in London added the following entry to the London office's pluto.secrets file:

```
12.254.84.24 24.248.36.1 136.74.82.167: RSA
    {
        Modulus: 0x92ba6085735a04a574ad9347d396954ba7e5c6d…
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent: 0x1874656b9339ab70e8c7998bf89918e1f…
        Prime1: 0xf05c167e391498298718c6f3c74271431ecdf9c36c…
        Prime2: 0x9c469297186637e6a3df188cd659ff669b3db4935…
        Exponent1: 0xa03d6454260dbac65a1084a284d6f62cbf33fbd…
        Exponent2: 0x682f0c65baeecfef17ea105de43bff99bcd3cdb7…
        Coefficient: 0x0f0721c77a99c19e800919f5b6ed709e3af7b1c…
}
```

Configuration File Details (Tokyo)

These lines were added to the security associations SanJose_SA_1 and SanJose_SA_2 in **ipsec\ipsec.cnf**, just before the Preshared-Secret keywords:

```
Auth-Type = RSA-Keys,
Remote-Public-Key = "0sAQNu5k11yzq5ezPhqFee4GESs7…",
```

These lines were added to the security association London_SA in the same file:

```
Auth-Type = RSA-Keys,
Remote-Public-Key = "0sAQNwwO6ObsiU5JbHjwYYnkSk5…",
```

At the same time, the network administrator in Tokyo added the following entry to the Tokyo office's pluto.secrets file:

```
136.74.82.167 24.248.36.1 12.254.84.24: RSA
  {
     Modulus: 0x92ba6085735a04a574ad9347d396954ba7e5c6…
     PublicExponent: 0x03
     # everything after this point is secret
     PrivateExponent: 0x1874656b9339ab70e8c7988bf89918e1f…
     Prime1: 0xf05c167e391498298718c6f3c74271431ecdf9c36c…
     Prime2: 0x9c469297186637e6a3df188cd659ff669b3db4935e…
     Exponent1: 0xa03d6454260dbac65a1084a284d6f62cbf33fbd…
     Exponent2: 0x682f0c64baeecfef17ea105de43bff99bcd3cdb7…
     Coefficient: 0x0f0721c77a99c19e800919f5b6ed709e3af7b1c…
  }
```

Additional Security Note

The Widgets, Inc. VPN is now ready to be deployed using RSA DSS host authentication. Extended Authentication is not supported when RSA DSS authentication is used and Preshared-Secrets are cancelled out. The Preshared-Secrets can be left in the SAs though.

17.5. VPN Using NAT-T and VPN Gateway

There may be situations in which the corporate head office wants to connect a remote user (or office) which is located behind a NAT device. In this case, VPN gateways cannot communicate directly, nor successfully establish tunnels. NAT Traversal is designed to overcome this deficiency by changing IPSec packets, allowing them to flow flawlessly through the NAT device.

VPN Scenario Details

A new branch of Widgets, Inc. is opening in Vancouver, and has been assigned the isolated network segment 192.168.1.0 and the IP address of the VPN Gateway (behind the NAT device) is 192.168.1.1. The new network administrator at the Vancouver office must take care to ensure that:

- UPD traffic between can successfully flow through the NAT device. This is because the IPSec NAT-T feature encapsulates all traffic in the UDP protocol (UDP port 4500).

- AH is set to "No", as the AH protocol conflicts with NAT.

- RSA DSS authentication is used between San Jose and Vancouver to ensure host identities

- San Jose VPN Gateway is configured to not initiate the IKE negotiations, as the IP address of the Vancouver NAT device is considered unknown (a dynamic IP address).

- Vancouver VPN Gateway is configured with small key material lifetimes, to overcome the problem of potentially "dead tunnels", when the NAT device IP address is suddenly changed.

Recall from previous scenarios that San Jose's public IP address is 24.248.36.1 and that San Jose has a Local-IP address of 10.1.0.1.

Configuration Files (San Jose)

The network administrator in San Jose adds an additional security association for the Vancouver network to the **ipsec\ipsec.cnf** file:

```
To_Vancouver_SA            Description = "10.1.0.0 <-> 192.168.1.0"
                           Local-IP = "24.248.36.1",
                           Local-Net = "10.1.0.0",
                           Local-Mask = "255.255.0.0",
                           Remote-IP = "0.0.0.0",
                           Remote-Net = "192.168.1.0",
                           Remote-Mask = "255.255.255.0",
                           AH = No,
                           ESP = Yes,
                           Auth-Type = RSA-Keys,
                           Remote-Public-Key = "0sAQN+d4Tbm…",
                           Preshared-Secret = "-3c0f9d1782099c5a",
                           Reinit = No
```

San Jose's private key, of course, is already stored in its pluto.secrets file:

```
24.248.36.1: RSA
   {
      Modulus: 0x6ee64975cb3ab97b33e1a8579ee06112b3bffb15…
      PublicExponent: 0x03
      # everything after this point is secret
      PrivateExponent: 0x127bb6e8f734743f335046b9452565831d…
      Prime1: 0xb1031cccba461879486d390749a6cc1ccdf182ea56…
      Prime2: 0xa062d66e386ff4710910f09a578c9acaf260463912…
      Exponent1: 0x7602133326d965a63048d0af866f32bddea101f…
      Exponent2: 0x6aec8ef4259ff84b5b60a066e5086731f6ead97…
      Coefficient: 0x0cf5830d402fc7205aec4c039199838e7fe88eb…
   }
```

Configuration Files (Vancouver)

The network administrator in Vancouver creates a security association for San Jose, adding it to the **ipsec\ipsec.cnf** file:

```
To_SanJose_SA              Description = "192.168.1.0 <-> 10.1.0.0"
                           Local-IP = "192.168.1.1",
                           Local-Net = "192.168.1.0",
                           Local-Mask = "255.255.255.0",
                           Remote-IP = "24.248.36.1",
                           Remote-Net = "10.0.0.0",
                           Remote-Mask = "255.0.0.0",
                           AH = No,
                           ESP = Yes,
                           Reinit = Yes,
                           Auth-Type = RSA-Keys,
```

```
                              Remote-Public-Key = "0sAQNu5k11yz…",
                              Preshared-Secret = "-3c0f9d1782099c5a",
                              IPSec-Lifetime = 70,
                              ISAKMP-Lifetime = 80,
```

The network administrator in Vancouver adds the Vancouver gateway's private key to the pluto.secrets file:

```
192.168.1.1 24.248.36.1: RSA
    {
        Modulus: 0x0x7e7784db98c38094cab48a116a220735d5f1aa…
        PublicExponent: 0x03
        # everything after this point is secret
        PrivateExponent: 0x1513eb79eecb4018cc736c583c5b0133a…
        Prime1: 0xb76c3e5249f03aa86fa95aefadc69575c1cf244b52…
        Prime2: 0xb081f0b5a29b7375f30bd47938b023bedbe253df0…
        Exponent1: 0x7a48298c314ad1c59fc63c9fc92f0e4e8134c2dc…
        Exponent2: 0x75abf5ce6c67a24ea207e2fb7b2017d49296e29…
        Coefficient: 0x3666dee6ef03e25026a601a06897696e4dd344…
    }
```

Part VI
References

18 Appendix A – Utility Programs

18.1. "ipsec" (IPSec management utility)

Introduction

The ipsec utility program allows you to monitor IPSec statistics, activate IPSec configuration changes in real time, or perform management tasks like changing a password or closing a VPN connection.

Synopsis

```
ipsec –COMMAND
```

Where -COMMAND is one of the following:

```
-init
-reconnect
-stat
-disconnect {name}
-password [config_dir template_dir] User-Id old_password new_password
```

Description

The **–init** command re-initializes the Pluto IKE Server and triggers key negotiations as if Pluto had just been started. If Pluto was started before the InJoy host application, InJoy IPSec will initialize Pluto automatically (the default), but if Pluto was restarted after the host application you need to invoke the ipsec tool with the –init parameter.

The **–reconnect** command causes the configuration files to be re-read and tunnels to be renegotiated. This is useful if you have recently updated one or more security associations and want connections to use the new security policies immediately. This command is similar to the command "sync –ipsec".

The **–stat** command causes the ipsec utility to display the current table of security associations.

The **-disconnect** command allows you to terminate a particular VPN tunnel using the name of the SA as an argument - as it appears in the SA configuration. The keyword "all" can be specified to terminate all tunnels.

The **-password** command allows you to create a new encrypted password suitable for use in the vpn-auth.cnf file, by supplying a username and password as arguments. You may also choose to do this directly in the GUI.

For additional details on using the ipsec utility program, invoke it from the command line, with the "-?" option.

18.2."rsasigkey" (RSA Signature Generation)

Introduction

Rsasigkey generates private and public RSA keys, which can be used to authenticate IPSec endpoints.

Rsasigkey emits its output as standard ASCII data to the screen.

For more information about the practical use of rsasigkey in IPSec VPN scenarios, please refer to advanced section 16.3, "RSA Digital Signatures".

Synopsis

```
Rsasigkey [ --verbose ] nbits
```

Description

The **--verbose** option makes rsasigkey give a running commentary on the screen. By default, it works in silence until it is ready to generate output.

The **nbits** specifies the number of bits in the generated keys. That is, two primes each of exactly nbits/2 bits. nbits must be a multiple of 16.

Note that key generation may be a lengthy process and for example a 1024-bit key on a 2GHz Pentium takes roughly 20 seconds. A 2048-bit key on the same system would take several minutes.

Example

To use the rsasigkey utility program to generate a 1024-bit signature and save it to a file called rsakey.txt, issue the following command:

rsasigkey 1024 > rsakey.txt

19

Appendix B - IPSec Interoperability

Test Engine Verification

The InJoy IPSec has been tested with the following public test engines.

- SSH Communications Security ISAKMP test engine

- National Institute of Standards and Technology test engine - IP Security Web Based Interoperability Tester (IPSec-WIT)

Interoperable Third-Party Solutions

The InJoy IPSec implementation has been tested for compatibility with the third party IPSec products in this section. Please note that testing generally does not include the full feature-set, but only the very basic protocols. This list is far from complete and several of the listed solutions have not been tested by F/X Communications, but are added as the result of user-feedback.

- NetScreen with X-auth v6 and AES (F/X Approved)

- F-Secure VPN for Windows - http://www.f-secure.com (F/X approved)

- PGP 6.5 Mac and Windows IPSEC Client - http://www.pgp.com (F/X approved)

- IRE Safenet/SoftPK WinNT Client - http://www.ire.com (F/X approved)

- Nortel Connectivity Extranet Switch (with ID_KEY_ID based authentication and inner IP address) - http://www.nortelnetworks.com (F/X approved)

- Cisco PIX and Cisco IOS (with extended authentication and IP address configuration) - http://www.cisco.com (F/X approved)

- Cisco VPN5008, SW 5.2.21.001

- Cisco VPN62xx (F/X approved)

- Microsoft Windows 2000 built-in IPSec - http://www.microsoft.com/windows2000 (F/X approved)

- Microsoft Windows XP built-in IPSec – http://www.microsoft.com/windowsxp (F/X approved)

- Linux IPSec [FreeS/WAN] – http://www.freeswan.org (F/X approved)

- Raptor Firewall 5 for Windows NT

- Xedia Access Point/QVPN

- Borderware 6.0 and Freegate 1.3 beta

- TimeStep Permit/Gate (2520)

- OpenBSD IPSec

- FreeBSD IPSec
- IBM Firewall on RS/6000 server

20 Appendix C – Protocol Support Summary

This section provides a summary of the protocols and the features that the IPSec Plugin provides:

- SA Types:
  - Tunnel mode
  - Transport mode
- ISAKMP SA negotiation methods:
  - Main mode
  - Aggressive mode
- IPSec protocols:
  - AH (RFC 2402)
  - ESP (RFC 2406)
- Policies:
  - Authentication
  - Encryption
  - Encryption/authentication
  - Authentication/encryption
- AH transforms:
  - HMAC MD5 (RFC 2403)
  - HMAC SHA (RFC 2404)
- ESP transforms:
  - 1-DES CBC (RFC 2405)
  - 3-DES CBC
  - AES (128, 192 and 256 key lengths)
  - BlowFish
  - NULL-ESP
- Key Exchange:
  - ISAKMP/Oakley (RFC 2412)
  - Extended Authentication within ISAKMP/Oakley (XAUTH) (draft-ietf-ipsec-isakmp-xauth-04.txt, draft-ietf-ipsec-isakmp-xauth-06.txt)
  - The ISAKMP Configuration Method (draft-ietf-ipsec-isakmp-mode-cfg-04.txt)

- o Perfect Forward Secrecy (PFS)
  - o RSA Digital Signatures Authenticating
  - o X.509 Digital Certificates
- Other VPN-related features:
  - o Firewall compatibility
  - o Road Warrior support (dynamic IP addresses)
  - o Road Warrior subnet support (network behind RW's)
  - o Logging
  - o Alert support

21 Appendix D – Configuration Attributes

21.1.Security Associations ("IPSEC.CNF")

Configuration Attributes

Attributes	Possible Values	Description
Section name	- text string (Maximum length = 31 characters)	Name of SA bundle. Useful for diagnostic purposes such as logging and monitoring.
Description	- text string (Maximum length = 50 characters)	Optional description of the SA.
SA-Status	- Enabled - Disabled	If SA-Status = Disabled, the SA description is ignored. The default value is Enabled.
Mode	- Transport - Tunnel	If Mode = Transport, IPSec transformations are applied only to the IP packet payload. The original IP packet header is not modified. The AH or ESP header is inserted right after the IP header. This mode can be used only for Host to Host links. Transport mode cannot be used in combination with the NAT Traversal feature (as that would be insecure). If Mode = Tunnel, the original (encapsulated) datagram becomes the payload for a new IP header. This mode must be used if at least one endpoint is an IPSec gateway. This is the only mode allowed for NAT Traversal.
Local-IP	- IP address - "0.0.0.0"	Local IPSec host/gateway IP address. Traditionally this is the public IP address of the local IPSec end-point. If Local-IP = "0.0.0.0", the local side is a Road Warrior. Road Warrior specifies that the local IP address is assigned dynamically. For more information,

		please refer to section 11, "Using Road Warrior Support."
		If Local-IP specifies the IP address of a PC that uses NAT Traversal, the internal IP address must be specified (e.g. 10.0.0.1). In addition, this IP address must belong to the Local-Net IP address range. If this is not possible, an alternative is to use the Inner-IP to specify a virtual IP address that actually does fit the Local-Net IP address range. For more information, please refer to Section 13, "Using IPSec behind NAT."
Local-Net	- net address	If the local IPSec endpoint acts as an IPSec gateway, it is possible, using the Local-Net and Local-Mask attributes, to specify an intranet for which all traffic will be processed according to the SA.
Local-Mask	- net mask	Netmask for Local-Net.
Remote-IP	- IP address - DNS name - "0.0.0.0"	Remote host or IPSec gateway IP address. Traditionally this is the public IP address of the remote IPSec end-point. If Remote-IP = "0.0.0.0", the remote end is a Road Warrior. Such an SA would be used for all remote dynamic-IP IPSec end-points. Note that IKE negotiations remote Road Warriors are authenticated using the same preshared secret. For more information, please refer to section 11, "Using Road Warrior Support." When NAT Traversal is used, it is recommended to have Remote-IP set as 0.0.0.0. Alternatively, the Remote-IP may be specified as the **public IP address** of the NAT device - not the internal address of NAT-T client. Consequently, there currently is no way to have two or more NAT-T clients connected from behind the same NAT device.
Remote-IP-2	- IP address - DNS name	Fail-over IP address, used when IKE negotiations with Remote-IP failed or timed out. Leave field empty to disable fail-over.

Remote-Net	- net address	If the remote IPSec end-point acts as an IPSec gateway, it is possible to specify the intranet for which all traffic will be processed according to the SA. When NAT Traversal is used, the Remote-Net / Remote-Mask IP address range must include the IP address of the NAT-T client. For more information, please refer to Section 13, "Using IPSec behind NAT."
Remote-Mask	- net mask	Netmask for the Remote-Net.
AH	- Yes - MD5 - SHA1 - NULL-Auth - No	This is the IPSec Authentication Header (AH) transform. You may specify MD5, SHA1, NULL-Auth or none. If you set this to Yes, MD5 will be used as the preferred value. AH should be disabled when NAT Traversal is used or when IPSec is attempted port-mapped. This is because the use of AH security prevents the changes that a NAT device must be able to apply. NULL-Auth value tells IPSec to not use ESP Authentication – i.e. only encrypt without introducing additional 12-byte authentication payload.
ESP	- Yes - AES - AES-192 - AES-256 - BF - 3DES - DES - NULL-ESP - No	This is the IPSec ESP transform. You may specify AES, AES-192, AES-256, BF, 3DES, DES, NULL-ESP or none. If you set this to Yes, 3DES will be used as preferred value. It is recommended NOT to use simple DES in high security setups. Even though 1 time DES does give an advantage in speed, it is vulnerable to cracking attempts of modern (powerful) computers. NULL-ESP means that no actual encryption will be performed on datagrams, i.e. tunneled data will be sent in the clear.
Reinit	- Yes - No	If Reinit = Yes, IKE negotiates to establish new SAs when the host application is (re-)started or (re-)

		connected. For example, with the InJoy Dialer™ that is when a new connection to the ISP is established or when the ipsec utility has been run with -reconnect option. It is normal and highly recommended to synchronize between endpoints at this point, except in the scenario specified below. Reinit = No specifies that IKE negotiations won't be started until traffic between endpoints will appear. This is the normal approach when Remote-IP = "0.0.0.0" (Road Warrior case), as the real IP address is not known and negotiations accordingly need to be started by the Road Warrior.
Exclude-Local-IP	- Yes - No	If Exclude-Local-IP = Yes, the local gateway will be excluded from the SA bundle. All traffic that has the local gateway as source or destination will NOT be processed by IPSec. Instead, only the local subnet will be covered by the SA bundle. This option is for compatibility with other vendors' IPSec implementations (e.g. SafeNet/Soft-PK). Please refer to the separate interoperability guides for more details. Exclude-Local-IP = No is a default value.
Exclude-Remote-IP	- Yes - No	If Exclude-Remote-IP = Yes, the remote gateway will be excluded from the SA bundle. All traffic that has the remote gateway as source or destination will NOT be processed by IPSec. Only the remote subnet will be covered by the SA. This option is for compatibility with other IPSec vendors. Exclude-Remote-IP = No is a default value.
ISAKMP-Lifetime	- seconds	ISAKMP SA key material lifetime. The ISAKMP SA is renegotiated at the given time interval. The default value is 3600 (1 hour). Not all VPN gateways will accept negotiation of this feature. When NAT Traversal is used, it is recommended to specify a relatively small lifetime (e.g. in the 60-120

		range). A small lifetime helps reduce problems related to the NAT device changing its public IP address. Different values for the ISAKMP-Lifetime and IPSec-Lifetime are recommended.
IPSec-Lifetime	- seconds	IPSec key material lifetime. The IPSec SA is renegotiated at the given time interval. The default value is 28800 (8 hours). Not all VPN gateways will accept negotiation of this feature. The NAT Traversal guidelines specified for the ISAKMP-Lifetime also applies to the IPSec-Lifetime.
Inner-IP	- Inner (Red Node) IP address	The Inner-IP attribute is used in an SA to specify a private (virtual) IP address for tunneled traffic. Intended for use on the IPSec client side, the Inner-IP provides translation of the public ISP-assigned IP address (Black Node IP) to a private (static) IP address (e.g. 10.1.1.1). NAT is used for the implementation of Inner-IP and the feature is fully transparent with standard (non call-back) TCP and UDP connections. Inner-IP support is also compatible with the Nortel Connectivity Switch (version 2.6 or newer) and Cisco PIX/IOS VPN gateways. With Nortel, the Inner-IP attribute must be manually preset, while with Cisco, the "config mode" feature can auto-assign the Inner IP address to the InJoy Client. Refer to the interoperability guides for examples.
Auth-Type	- IP-Address - Client-IdKeyId - Client-Xauth - Server-Xauth - RSA-Keys - Unused	Auth-Type defines how IPSec end-points identify themselves during IKE negotiations. The default method is to use IP address. In this case authentication is based only on the pre-shared secret and IP addresses of the IPSec end-points. If Auth-Type = Client-IdKeyId, the User-Id value is used as ID_KEY_ID in aggressive mode authentication. This

| | | method is used when the remote side is a Nortel Extranet Switch. Refer to the interoperability guide for examples. You should provide User-Id and Password as well in this case.

If Auth-Type = Client-Xauth, then your side is configured as a client for an Extended Authentication server. You should provide User-Id and Password as well in this case.

Auth-Type = Server-Xauth defines that your side is an X-authentication server that authenticate clients based on their User-Id and Password. In this mode it is also possible to configure the clients with predefined parameters, such as a static Inner IP address.

Auth-Type = RSA-Keys defines that both sides authenticate themselves by the use of RSA Digital Signatures. RSA-Keys require a local Private key in pluto.secrets and a correct remote Public key in the Remote-Public-Key attribute.

Auth-Type = Unused means that Auth-Type will not be used to identify authentication, and instead, Authentication, Identification and Authorization attributes will be used. |
| --- | --- | --- |
| User-Id | - text string
- "prompt" | Used as a user name for authentication with Cisco Pix, Cisco IOS and InJoy XAUTH enabled servers.

Set User-Id to "prompt" to have IPSec prompt the user for a login at each connect, rather than storing the password on the harddisk.

Also used as ID_KEY_ID when Auth-Type = Client-IdKeyId for compatibility (with e.g. the Nortel IPSec Gateways). |
| Password | - text string | Used in combination with User-Id for authentication with ID_KEY_ID or XAUTH enabled servers. |

AH-Receive-SPI AH-Transmit-SPI ESP-Receive-SPI ESP-Transmit-SPI	- unique integer	Used only in combination with Manual Keying. Security Parameter Indices of the remote IPSec end-point. Only specified by the IPSec administrator when "Manual Keying" is used (i.e. no IKE Server). For more information about these fields, please refer to Section 15, "Using Manual Keying".
AH-Key Remote-AH-Key	- key string	Used only in combination with Manual Keying. AH Keys for the local and remote side, respectively. MD5 or SHA1 key in hexadecimal representation. 16 bytes (32 characters) for MD5. 20 bytes (40 characters) for SHA1. For more information about these fields, please refer to Section 15, "Using Manual Keying".
ESP-Key Remote-ESP-Key	- key string	Used only in combination with Manual Keying. If AH is used, ESP-Key includes either a DES (8 bytes or 16 hex characters) or 3DES (24 bytes or 48 hex characters) key. If AH = no, the ESP-Key contains either a DES or 3DES key, followed by either an MD5 or SHA1 key. For that reason, these keys can consist of 24, 28, 40 or 44 bytes - which is equal to 48, 56, 80 and 88 hex characters, respectively. For more information about these fields, please refer to Section 15, "Using Manual Keying".
Preshared-Secret	- text string	Pre-shared secret is used to pre-authenticate IPSec end-points before any other authentication. Pre-shared secrets must be shared by contacting end-points on beforehand

		(manually). If RSA DSS authentication is used (Auth-Type of RSA-Key), the Preshared-Secret is not used by IPSec. If manual keying is used, the Preshared-Secret is also not used.
Aggressive-Oakley	text string of "enc;hash;auth;group" enc: DES | 3DES | AES | AES-192 | AES-256 | BF hash: SHA1 | MD5 auth: PK group: DH1 | DH2 | DH5	Specifies the Oakley transform to be used in Aggressive Mode. By default, the Pluto IKE Server uses the "3DES;SHA1;PK;DH5" transforms. For example, the IBM AIX IPSec accepts DH Groups 1 and 2 only. Aggressive-Oakley key for AIX could be: "3DES;SHA1;PK;DH2".
Initial-Retry-Delay	- Fast - Medium - Slow - VerySlow	Defines the initial retry delay for the initial IKE Main Mode packets. The faster the value, the smaller the packet resend interval.
Local-ID	- @FQDN - user@FQDN - empty	Defines the ID of the local side. The ID is an identification payload used by the IKE servers to uniquely identify each other. The ID can be: @FQDN: Fully Qualified Domain Name (@example.com); user@FQDN: e-mail address (user@example.com); An IP address (which – if the Local-ID is left blank – is automatically inserted into this field). If the Local-ID is explicitly set to an IP address, it causes an error.
Remote-ID	- @FQDN - user@FQDN - empty	Similar to Local-ID.
Transform-Order	text string of "enc;hash;auth;group" sequences separated by whitespace	Specifies the order in which Oakley transforms appear in the IKE Server proposal. For advanced users only.

	enc: DES | 3DES | AES | AES-192 | AES-256 | BF hash: SHA1 | MD5 auth: PK group: DH1 | DH2 | DH5	
PFS	Yes No	Turns on Perfect Forward Secrecy for Phase II negotiations. Read above for more information about PFS.
Direct-Nets	Whitespace-separated list of networks with wildcards	Specifies which networks to skip when processing the traffic.
Authentication	PSK Tokencard RSA-Keys X509	Specifies which method to use when verifying authenticity of the remote endpoint. Default: PSK.
Identification	IP-Address User-FQDN Host-FQDN DERASN1 IdKeyId	Identification type to be used when authenticating peers, i.e. this is a selector used for choosing authentication material from the policy database. IP-Address will use IP addresses of the tunnel endpoints to identify them. User-FQDN and Host-FQDN are user's e-mail and user's host name, respectively. Refer to Local-ID attribute description for more details. DERASN1 is identification type used for X509 authentication and is the only type possible for X509. IdKeyId is special type for interoperating proprietary implementations of IKE server (like OpenPGP). Default: IP-Address.
Authorization	None Xauth_Cli Xauth_Srv XauthV6_Cli XauthV6_Srv	Specifies which method will be used for additional verification of user identity. None – no additional authorization will be performed.

		Xauth_Cli and Xauth_Src – client and server mode of XAUTH, respectively. XauthV6_Cli and XauthV6_Src – client and server mode of XAUTH v6, respectively. For XAUTH type of authorization, client and server must not have same types of it, e.g. both must not have Xauth_Cli. Default: None.
Local-Certificate	Path to certificate file	Specifies this endpoint's certificate file. The file must reside under **ipsec/ipsec.d/certs** directory and be in PEM or DER format (the IKE server auto-senses the format).
NAT-Traversal	Auto No	NAT-Traversal allows IPSec end-points to function behind a NAT device. NAT-Traversal = No explicitly disables this feature. This is useful only when it is known beyond doubt that no NAT processing will occur between IPSec endpoints. NAT-Traversal = Auto allows the IKE Server to detect the presence of a NAT device between IPSec endpoints and automatically apply relevant procedures (for instance UDP encapsulation of IPSec traffic). If one of the IPSec end-points doesn't support NAT Traversal, the NAT-T capable IKE Server auto-disables the feature.

Default Values

The default values of any IPSec SA, are specified in the file **template\ipsec.cfg**:

```
; TEMPLATE FILE
;
; Provides default values for the user crafted configuration records.
; Any value may be selectively overwritten in the product config files.
```

```
TEMPLATE    Description = "<Secure Tunnel>",
        SA-Status = Enabled,
        Exclude-Local-IP = No,
        Exclude-Remote-IP = No,
        Mode = Tunnel,
        Reinit = Yes,
        Auth-Type = IP-Address,
        AH = No,
        ESP = Yes,
        IP-Compression = No,
        ISAKMP-Lifetime = 28800,
        IPSec-Lifetime = 3600,
        Local-IP = "",
        Local-Net = "",
        Local-Mask = "",
        Remote-IP = "",
        Remote-IP-2 = "",
        Remote-Net = "",
        Remote-Mask = "",
        Inner-IP = "",
        User-Id = "",
        Password = "",
        ESP-Key = "",
        AH-Key = "",
        Remote-ESP-Key = "",
        Remote-AH-Key = "",
        ESP-Receive-SPI = 0,
        ESP-Transmit-SPI = 0,
        AH-Receive-SPI = 0,
        AH-Transmit-SPI = 0,
        Aggressive = No,
        Cisco-Delete = No,
        Preshared-Secret = "<secret>",
        Aggressive-Oakley = "3DES;SHA1;PK;DH5",
        Initial-Retry-Delay = Medium,
        Remote-Public-Key = "",
        Local-ID = "",
        Remote-ID = "",
        NAT-Traversal = Auto,
        Authentication = PSK,
        Identification = IP-Address,
        Authorization = None,
        Vendor-ID = No,
        Remote-CA = "",
        PFS = 0,
        Keying-Tries = 3,
        IPv6 = No,
```

```
        IKE-Flags = None,
        Direct-Nets = "",
        Transform-Order = "",
        Local-Certificate = "",
        Remote-Certificate = "",
```

21.2. IPSec Options ("OPTIONS.CNF")

Configuration Attributes

Attributes	Possible Values	Description
Trace-AH	- Yes - No	When set to Yes, the trace options will result in comprehensive tracing for the protocol they cover. For instance, if you set Trace-AH to Yes, all AH packet modifications will be logged. The trace information is logged to logs/ipsec.log. These attributes are designed to offer profound debugging capability and should generally be left disabled.
Trace-ESP	- Yes - No	
Trace-Tunnel	- Yes - No	
Trace-Frag	- Yes - No	
Trace-Packets	- Yes - No	
Trace-IPCOMP	- Yes - No	
Trace-Internal	- number	This configuration attributes is reserved to diagnostic purposes and should be left at its default value.
Dump-Packets	- Yes - No	Set to Yes, to get full packet dumping in hexadecimal representation.
Log-Level	- Internal-Error - Error - Minimal - Warning - Info	Specifies the logging level of the IPSec engine – from most quiet to most verbose.
Start-IKE-Server	- Daemon - Yes - No	With this option, IPSec allows auto-starting of the Pluto IKE server at start-up. Daemon: Starts Pluto as an invisible process (sometimes also referred to as a detached or "backgrounded" process); Yes: Starts Pluto as a visible process – in a command prompt. Note that it

		may be started minimized on some Operating Systems. No: Assumes Pluto is started manually before IPSec.
IKE-Server-Parameters	- text string	List of command line parameters to pass to the Pluto IKE Server (when auto-started). See also Start-IKE-Server.
Nested-SA-Bundles	- Yes - No	Controls SA bundle nesting – used only in very complex, multi-level tunnels. By default Nested-SA-Bundles is disabled.
Log-Limit	- Kbytes	Limits the maximum size of the logs\ipsec.log and the ipsec\pluto.log log files. When the log files hit their maximum file size, they are deleted. The file size must be entered in Kilobytes (e.g. 10000 = 10MB).
Path-MTU-Discovery	Yes No	Specifies whether built-in Path MTU Discovery is activated.
Local-MTU	- MTU value	Specifies MTU value to be used for forwarding datagrams to local internal network (specified by Local-Net in **ipsec.cnf**).
Remote-MTU	- MTU value	Same as Local-MTU, but for remote internal network.
Lookup-Period	- seconds	Lookup-Period specifies the time interval for periodic DNS resolving of Remote-IP host names. If IPSec detects a change in the Remote-IP domain name, IPSec re-establishes the tunnel, using the new IP address.

Default Values

The default values of the common IPSec options are specified in the file **template\options.cnf**:

```
; TEMPLATE FILE
;
; Provides default values for the user crafted configuration records.
; Any value may be selectively overwritten in the product config files.


TEMPLATE   Trace-AH = No,
           Trace-ESP = No,
```

```
Trace-Tunnel = No,
Trace-Frag = No,
Trace-Packets = No,
Trace-Internal = 0,
Dump-Packets = No,
Trace-IPCOMP = No,
Log-Level = Internal-Error,
Start-IKE-Server = No,
IKE-Server-Parameters = "",
IKE-Server-Name = "pluto",
Nested-SA-Bundles = No,
Log-Limit = 10240,
Lookup-Period = 300,
IPSec-Flags = 0,
Hardware-Acceleration = No,
Path-MTU-Discovery = No,
Local-MTU = 1500,
Remote-MTU = 1500,
```