IPSec Tutorial

Scott Cleven-Mulcahy

Keywords: IPSec, Training, Security, IKE, AH, ESP

Table of Contents

1 Acronyms 3

2 Security Concepts 4

2.1 Ciphers 4

2.2 Hashes 5

3 IPSec Overview 5

4 Policies, Selectors, and Actions 8

5 How It Works 10

5.1 Step 1: Policy, Selector, and Action 10

5.2 Step 2: IKE 10

5.3 Step 3: AH and ESP 12

5.4 Interacting With IPSec 13

6 Implementing IPSec on W2K 14

6.1 IPSec Policies 15

6.2 Configuring IKE 15

6.3 Creating Filters (Selectors) 16

6.4 Creating Actions 16

6.5 Concluding W2K IPSec 18

7 Problems, Pitfalls, and Solutions 18

7.1 Dynamic vs. Static Policies 18

7.2 MTU Size 19

7.3 ICMP 19

7.4 W2K Tools 20

7.5 Protecting IPSec 21

8 Case Studies 21

8.1 IPSec Packet Filtering 22

8.2 IPSec Gateway 24

Appendix A 27

Appendix B 28

List of References 29

1 Acronyms

- AH = Authentication Header
- AD = Active Directory
- DES = Data Encryption Standard
- DH = Diffie-Hellman
- DoS = Denial of Service
- ESP = Encapsulating Security Payload
- HIDS = Host Intrusion Detection System
- HMAC = Hashed Message Authentication Codes
- ICV = Integrity Check Value (used by AH and ESP for authentication)
- IDS = Intrusion Detection System
- IKE = Internet Key Exchange
- IPSec = Internet Protocol Security (RFC2401)
- L2TP = Layer 2 Tunneling Protocol
- MD5 = Message Digest 5 (hashing algorithm)
- MTU = Maximum Transmittable Unit
- NAT = Network Address Translation
- NIDS = Network Intrusion Detection System
- PFS = Perfect Forward Secrecy
- PGP = Pretty Good Protection
- PKI = Public Key Infrastructure
- PMTU = Path Maximum Transmittable Unit
- PPTP = Point to Point Tunneling Protocol
- RFC = Request For Comment (Internet standard documents)
- RMR = Revenue Management Re-Engineering (project at A-B)
- SHA = Secure Hash Algorithm
- SSL = Secure Sockets Layer
- TLS = Transport Layer Security
- TTL = Time to Live
- VPN = Virtual Private Network

2 Security Concepts

It is important to understand some key concepts prior to delving into IPSec. IPSec borrows from cryptography and Public Key Infrastructure (PKI) technologies heavily. Keys, Hashes, Signatures, Ciphers, and many other security concepts are used to create IPSec. Although these are all important in the creation of the IPSec standard, detailed knowledge of them is not necessary to design, implement, and support IPSec solutions. This section will not attempt to teach the mathematics behind these technologies. Instead it focuses on understanding concepts and providing the ability to make informed decisions on the choices IPSec provides.

IPSec security focuses around a few basic concepts. These concepts are defined and given names. The table below summarizes the concepts and the term that describes how IPSec uses them.

Term	Concept	Description
Confidentiality	Hiding information from prying eyes	Encrypt the information with a cipher
Integrity	Prevent unauthorized changes to information	Generate a number based on the information in such a way that the number changes if the message changes
Non-repudiation	Authenticity of the sender	Verify the identity of the sender (person or device) so that they cannot deny sending the information
Anti-replay	Prevent playback of old packets	Give information a Time To Live (TTL)

2.1 Ciphers

There are two types of ciphers that we are interested in: symmetric and asymmetric. Symmetric ciphers use the same key/method to encrypt and decrypt messages. Generally, symmetric algorithms are considered weaker than asymmetric algorithms. However, symmetric algorithms are typically much faster and better suited to encrypting large chunks of data than asymmetric algorithms. Asymmetric algorithms are one-way functions and usually come in the form of public key/private key. The private key can decrypt anything encrypted by the public key and the public key can decrypt anything encrypted with the private key. Both keys are incapable of decrypting a message encrypted with itself.

IPSec uses symmetric ciphers to encrypt the data streams. However, symmetric keys are both vulnerable and powerful. So IPSec defines how often a symmetric key, which is created by a symmetric cipher, is regenerated. Asymmetric keys are used to encrypt the transmission of the symmetric key. Since asymmetric keys are computationally stronger they are usually allowed a longer life time. This strategy takes advantage of the speed of symmetric ciphers while avoiding their weakness by protecting that key with power of an asymmetric cipher.

2.2 Hashes

Hashes are fairly simple in concept: they take a message and reduce it to something called a digest. It is analogous to what Reader's Digest. Reader's Digest reduces a book to something much smaller. Similarly, a hashing function takes a variable length message and reduces it to a fixed length output. The result is called a hash or a digest. Hashing functions will produce the same hash every time if the same text is used as input.

Hashes are often used as a signature. Signatures provide a means of checking the integrity of a message. If the signature of message is the same when it is received as when it was sent then we know the message has not changed. Conversely, if the signature has changed then we know the message must have been changed.

This leads us to two issues. First, if the signature (remember signature=hash=digest) is sent in the clear then there is nothing to prevent someone from changing the message and generating a new signature. Second, it would be convenient if we could use the signature to also prove who sent the message to us. The method used to protect the signature and provide non-repudiation is called Keyed Hashing. IPSec uses a particular type of keyed hashing called Hashed Method Authentication Codes (HMAC).

HMAC, like all keyed hashing, hashes the original message with a secret key. In other words, the secret key is considered part of the message when it is sent through the hashing function. Now if a message is intercepted and the contents changed the attacker is unable to generate an acceptable signature because they don't have the secret key. If the signature is valid then we know the message hasn't changed and who sent the message. This allows IPSec to consolidate integrity checking with authentication.

3 IPSec Overview

IPSec is not a protocol in itself; rather it is a protocol suite that enables authentication, confidentiality, integrity, and anti-replay protection between systems. It is important to note that it does not authenticate users; it authenticates devices. Additionally, IPSec can filter out what communication a device will accept or specify what requirements are necessary to establish a connection between devices. The final feature that IPSec offers is VPN functionality.

IPSec operates in either "Transport mode" or "Tunnel mode". Transport mode is only available in end-to-end communication. In transport mode IPSec maintains the original IP header information and inserts IPSec fields into the packet. Tunnel mode is most frequently used in gateway-to-gateway communication or with a Virtual Private Network (VPN). Tunnel mode encapsulates the original IP packet inside of an IPSec IP packet. Each mode provides strong protection, but using a slightly different solution. Transport and Tunnel mode are discussed in more detail in the section "How it Works".

IPSec is broken into multiple protocols.

- Internet Key Exchange (IKE) protocol
- Authentication Header (AH) protocol
- Encapsulating Security Payload (ESP) protocol

Together these protocols secure the communication between devices. IKE creates and manages the keys used by it, AH, and ESP. AH provides authentication, integrity, and anti-replay services at the network layer and up. ESP provides authentication, confidentiality, integrity, and anti-replay services at the transport layer and up. Confidentiality is a fancy way of saying encryption. Most commonly, AH and ESP are used together to provide a high level of security.

(adapted from a diagram by Jason Fossen "IPSec, RRAS, & VPNs")

IPSec Protocol	Function
IKE	• Creates a secure channel for passing keys • Provide keys for AH • Provide keys for ESP
AH	• Authentication of devices • Integrity checking of entire packet • Anti-Replay protection
ESP	• Authentication of devices • Integrity checking of transport and application layer • Confidentiality of transport and application layer • Anti-replay protection

A huge advantage for the IPSec protocol suite is that all the protocols work together to provide a comprehensive and modular security framework. This allows it to protect differing network layers as your needs dictate. In contrast, other security technologies focus on only one network layer and most fail to provide anti-replay protection. Below is a diagram that shows where various security protocols map into the DoD networking model (J. Fossen, "IPSec, RRAS, & VPNS").

DoD Network model	**Security protocols**
Application (FTP, HTTP)	PGP
Transport (TCP, UDP)	ESP, SSL/TLS
Network (IP,ICMP)	IKE, AH, L2TP
Physical (Ethernet)	Crypto-hardware

As mentioned, there are disadvantages to IPSec. The biggest disadvantage is that IPSec in Transport mode doesn't work with many proxies or Network Address Translation (NAT). These common security devices modify various fields in a packet, which causes IPSec to discard the packet because integrity has been violated. IPSec in tunnel mode is more NAT friendly but at the expense of further reducing available payload space and thus communication efficiency.

There are other security protocols available. Each has strengths and weaknesses. Below is a summary of some popular security protocols.

(adapted from a diagram by J. Fossen, "IPSec,RRAS, & VPNs")

Protocol	Network Layer	Advantages	Disadvantages
PGP	Application	• Strong security • Integrates well with email applications such as Outlook	• Difficult to deploy • Not user transparent
SSL/TLS	Transport	• Simple • Confidentiality • Works with NAT and proxies • Authentication • Integrity checking • Anti-replay	• Transport layer only • Applications must be aware of and request security functions

ESP	Transport	• User and application transparent • Authentication • Integrity checking • Confidentiality • Anti-replay	• Does not protect entire packet • May not work with NATs or proxies • Only works with TCP/IP
L2TP	Network	• Works with NATs and proxies • User transparent • Works with any protocol • Confidentiality	• Very new • Less supported • No authentication • No replay protection • No integrity checking
AH	Network	• User and application transparent • Authentication • Integrity checking • Anti-replay • Protects entire packet	• No confidentiality • Unable to use NATs or proxies • Only works with TCP/IP

4 Policies, Selectors, and Actions

IPSec is implemented through the creation of policies, selectors, and actions. It is through the IPSec policy definition that a device determines when, if, and how to implement IPSec.

In order to implement IPSec the device must be assigned a policy. The assigned policy is also known as an **active policy**. Each device may only have one active policy. This policy is then used for all communication sent and received from the device. If two devices don't have compatible policies then communication between those devices will fail. This makes it critical to properly plan an IPSec deployment, especially if a device must communicate with other devices not under your control.

Within each policy are one or more **Selectors** that determine what, if any, action the policy should take. Please note, W2K calls a Selector a filter. This document will use the term Selector rather than W2K's term of filters. Selectors act as filters for incoming or outgoing packets. IPSec processes packets that match the Selector, whereas it ignores packets that don't match any Selectors. A policy may have multiple Selectors. Below is a list of common IP fields that are used to define Selectors.

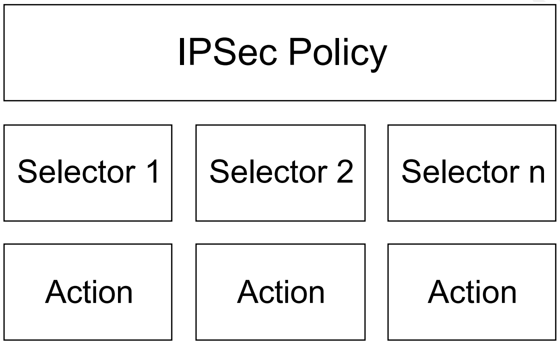
- Source IP address
- Source port
- Destination IP address

- Destination port
- Transport protocol

Each Selector has an **action** associated with it. The selector that most closely matches a packet performs the defined action. Only one action is allowed per selector. Below is a list of actions.

- Permit
- Block
- Use AH (Tunnel or Transport mode)
- Use ESP (Tunnel or Transport mode)
- Use AH & ESP (either or both in Tunnel or Transport mode)

The actions of permit and block allow IPSec to be used as a packet filter. For instance, a device could have a selector that Blocks any packet that comes from host A and another selector that permits any packet coming from host B. So one selector could block all traffic from any source, while another selector permits traffic from any source to TCP port 80 on its host. Since the selector that most closely matches the packet is selected, any traffic destined for TCP port 80 is allowed and all other traffic is blocked. Using IPSec as a packet filter shows how modular IPSec is: you can choose to use any of its security features or none at all. Below is a diagram showing the relationship between Policies, Selectors, and Actions.



In order to use authentication, confidentiality, integrity, and anti-replay, a selector must define the appropriate action. In other words, a selector's action must specify the use of AH, ESP, or both. The action can be so specific as to specify the type and strength of authentication AH or ESP should use, the type and strength of encryption (confidentiality), and the type and strength of the signature (authentication and integrity). In W2K these options are set in Local Security Settings - Policy - Selector - Action - Custom setting. Below are W2K configurable options.

Options	Values

Security method	• High (ESP) • Medium (AH) • Custom
Data and address integrity without encryption (AH)	• MD5 • SHA1
Data integrity and encryption (ESP)	Integrity • None • MD5 • SHA1 Encryption Algorithm • None • DES • 3DES
Frequency of Key generation	• In Kbytes (default 100,000) • In seconds (default 900)

5 How It Works

As with the TCP/IP protocol suite, IPSec protocols work in unison to create a secure communication. The whole process can be broken down into three phases:

- Determine if a communication requires IPSec
- Negotiate and establish a secure connection
- Transmit the data

5.1 Step 1: Policy, Selector, and Action

The active IPSec policy and its selectors determine if a communication requires IPSec. If a packet matches a selector within the active policy then the specified action is performed. If that action is only to block or permit a packet then steps 2 and 3 are skipped. The process only proceeds to step 2 if a selector's action is to use AH and/or ESP. It is important to note that if IPSec is enabled on a host then **every** packet goes through this step.

Usually, it is very important to create a mirror of each selector. In other words, if one selector permits traffic from any address to any address on TCP 80 then another rule is required to permit traffic from any address on TCP 80 to any address. This permits the two-way flow of communication. Under W2K, checking Mirrored automatically creates the mirrored selector. Other platforms may require that the mirrored selector be entered.

5.2 Step 2: IKE

IKE creates the keys that the subsequent steps will use to encrypt and sign packets. However, IKE is faced with a problem. If those keys are sent over an insecure connection then someone could "steal" those keys and view or modify the packets we are attempting to secure. In essence, IKE is faced with a chicken and egg problem: a secure connection requires keys but it can't send the keys until it has a secure connection. To tackle this problem IKE is broken in to two phases.

Phase 1 solves the problem of creating a secure channel over an insecure connection with a mathematical algorithm that permits anyone to view the communication occurring between the hosts and yet be unable to capture or predict the key that results from this communication. This algorithm is called the Diffie-Hellman (DH) key exchange.

When configuring Windows 2000 IPSec, there are two DH groups to select from: Low and Medium (there is no High). The important thing to know about DH groups is that it determines the strength of the phase 1 keys that are generated. It is strongly recommended that Medium be selected.

Unfortunately, DH is extremely CPU intensive. As a compromise W2K, as recommended by the RFC, only generates a DH key at the start of a communication and not for each packet sent. In fact, the same DH keys can be used for multiple, independent communication streams between two devices. If necessary, IPSec can be configured to generate new DH keys for each communication and periodically recreate those keys during a communication.

Phase 2 uses the DH keys to create a secure channel. This secure channel is used to create a subsequent set of keys that AH and ESP will use to encrypt and sign packets. It is important to note that phase 2 requires the phase 1 keys to work. The sequence of events is listed below. (www.networksorcery.com/enp/rfc/rfc2409.txt)

1. Device A communicates to Device B using IKE on UDP 500
2. Each side generates a Diffie-Hellman key
3. Device A and Device B create an encrypted connection using the keys from step 2
4. Device A and Device B negotiate the highest level of security supported on both devices
5. Device A and Device B create phase 2 keys for use with IPSec (AH and/or ESP)
6. IPSec communication (AH and/or ESP) begins using phase 2 keys created in step 5

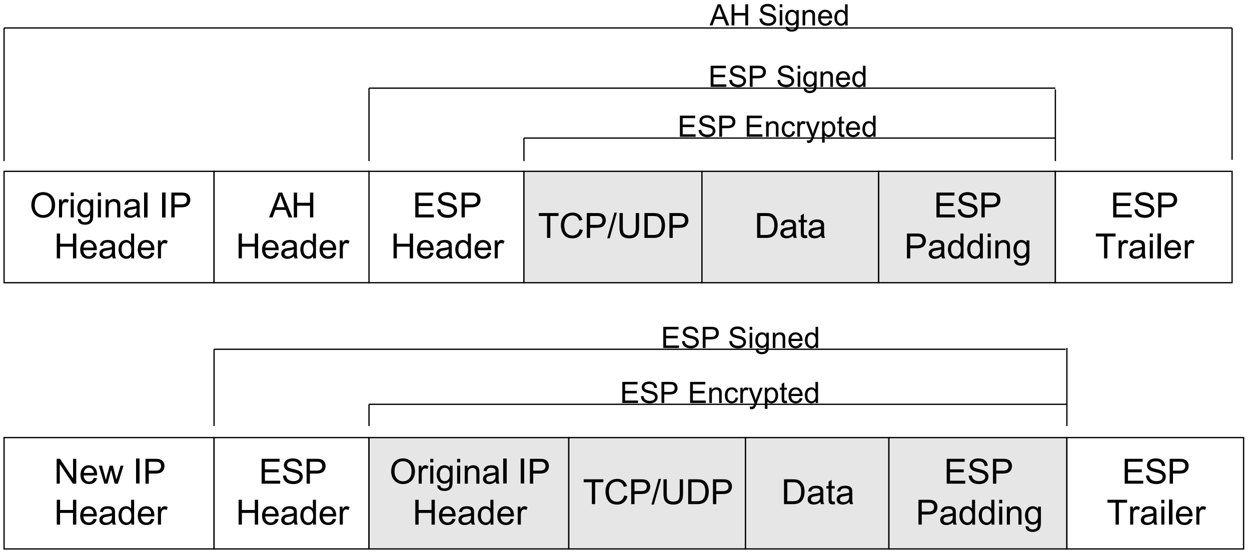
Step 1 deserves some special attention. As mentioned earlier IPSec can act as a packet filter: passing packets that match a selector or dropping packets that do not match. IKE is treated specially by IPSec: UDP 500 is automatically accepted. If it were not then a chicken and egg problem would arise: device A needs to negotiate a secure channel with device B, to do so it must connect on UDP 500, in order to make that connection it must establish a secure connection, to do so it must connect on UDP 500…. Obviously this is unacceptable. The IPSec RFCs require that IKE packets are recognized as such and processed appropriately.

5.3 Step 3: AH and ESP

AH and ESP perform separate but similar functions. AH verifies the identity of the sender using IKE Phase 2 keys to sign the IP packet (authentication and integrity) and keeps track of the packet sequence and lifetime of the phase 2 keys (anti-replay). AH does not encrypt the packet (confidentiality) and it works at the network layer. ESP can verify the identity of the sender, sign the transport layer, and keep track of the packet sequence. However, it also has the ability to encrypt the packet, but only encrypts the transport layer and higher.

Both AH and ESP are available in Transport and Tunnel mode. As discussed previously, Tunnel mode encapsulates the entire packet. Although AH supports tunnel mode, in practice it is not usually deployed in this fashion since it provides no additional protection over Transport mode. If tunnel mode is required, ESP is typically used. Since ESP tunnel mode protects the entire original packet AH is not necessary.

It should be noted that W2K provides minimal support for Tunnel mode. Since W2K is not typically deployed as a gateway there is little need for Tunnel mode support in W2K. Below is a diagram of transport and tunnel mode packets (AH tunnel mode is not depicted).



As can be seen, the header and trailer are not encrypted. AH keeps its HMAC in the header and ESP stores its HMAC in the trailer. When the destination device receives the packet it calculates the HMAC and compares it to the HMAC in the trailer. Not encrypting the HMAC allows IPSec to quickly determine if the packet has been modified and, if so, discard the packet.

5.4 Interacting With IPSec

Years of experience with IPv4 have exposed us to various aspects of TCP/IP communication: transport protocols have port numbers, NAT and proxy servers have become standard networking features, we have learned to accept replay attacks as vulnerability, and Network IDS is considered a requirement by many security professionals. IPSec changes the rules on how and what we deploy in our networks and in some cases we no longer need accept certain limitations of TCP/IP.

Unlike many transport layer protocols, such as TLS, TCP, and UDP, ESP uses only a protocol number. AH also uses a protocol number as is typical of network layer protocols. Specifically, AH uses protocol number 51 and ESP uses protocol number 50. If the two are used together then the protocol number of the outer most protocol is exposed. AH and ESP in transport mode will expose protocol 51 whereas ESP in tunnel mode and AH in transport mode will expose protocol 50.

As was mentioned in previous sections, AH and ESP may not work with a proxy server or NAT. More generically, IPSec, in either transport or tunnel mode, does not work with a device that modifies an IP packet. This is because AH and ESP sign their portions of each packet. AH signs the entire IP packet with the exception of a few fields that must change in normal IP communication (such as TTL). Tunnel mode allows the new IP header to be modified enroute, unless AH in transport mode is also used, but the rest of the packet is still off limits. The result is that regardless if you use AH or ESP the payload is always off limits for modifications, tunnel or transport, and the IP and transport header information may also be off limits.

Recently Network and Host IDS have become an important tool in the security professional's toolbox. NIDS are basically protocol analyzers. They capture packets on the network, analyze the packets to determine if it is malicious, alert security administrators of malicious packets, and in some cases will terminate communication containing malicious packets.

All IDS' work on the premise that the packet is transparent and that the contents are in an expected position. IPSec complicates this by encrypting portions of the packet, ESP, or moving the location of parts of a packet, tunnel mode. The solution will depend on the organization. Tunnel mode moves the original IP header and Transport layers from the expected location within the packet. However, tunnel mode is almost exclusively used between gateways. NIDS will need to be located inside of each gateway, prior to IPSec tunneling being applied, to permit the NIDS to work.

Since transport mode does not relocate important packet information, in principle NIDS will work with transport mode. If ESP is used with encryption there is no simple solution, however. One solution is to not use ESP with encryption on a network that requires NIDS. Another solution is to use ESP in tunnel mode to a gateway. Once at the gateway the IPSec is decrypted and available for inspection by NIDS before reaching the next gateway, which encrypts the packet before sending it to the final destination. Potentially a Host IDS that provides packet analysis can be used. This solution only works if the HIDS is able to inspect the packet after the packet has been decrypted. HIDS solutions can quickly become extremely expensive since every host in the network requires the software.

Replay and man-in-the-middle attacks are difficult to protect against. Fortunately, they are somewhat difficult to carry out since they require direct access to the network the systems use. Replay attacks capture the packets of a real session, alter the packets, and then retransmit the packets. The end-point hosts believe they are communicating directly with each other, but in fact they are communicating with a server in between. This is attack is extremely difficult to detect.

As has been mentioned many times, IPSec provides anti-replay protection. This is done by using a sliding window. IPSec gives each packet a sequence number. If a 64-packet sliding window is used then a host will accept packets 1 – 64 without problem. When packet 65 arrives the host will no longer accept packet number 1. Likewise, when packet 100 arrives, any packets below number 36 are dropped. In addition, IPSec does not accept duplicate sequence numbers. In other words, once packet 125 arrives for a given IPSec session, IPSec does not permit another packet 125. This also means that IPSec keys must be renegotiated and a new IPSec session established before IPSec sequence numbers wrap. This scheme makes replay attacks very difficult since the attacker only has a narrow window of time to retransmit the packet.

IPSec provides only limited man-in-the-middle protection. This protection is dependent on authentication method selected. In W2K, 3rd party certificates, Kerberos, and shared secret are supported. Of these, only 3rd party certificates provide strong man-in-the-middle protection.

6 Implementing IPSec on W2K

The IPSec RFC's permit great latitude on some implementation details. As is common with all software vendors, Microsoft used some of that latitude to differentiate their products from other vendors. It is important to note that the Microsoft IPSec implementation is RFC compliant. RFC compliance does not mean that all IPSec implementations are configured the same or support the same options.

The only area where Microsoft deviates greatly from the rest of the Internet community is in terminology. Specifically, Microsoft sporadically chooses to use the term "Filter" to refer to what everyone else calls a selector. The problem is made worse by Microsoft's inconsistence use of the term "filter". Fortunately, Microsoft at least uses "filter" consistently within W2K.

6.1 IPSec Policies

A strength of the W2K implementation is that IPSec is implemented in an object oriented fashion. A key concept of objects is reusability. Reusability reduces work by eliminating the need to duplicate identical work. Since work need not be duplicated the chance of error is reduced. Microsoft made a conscious effort to make many aspects of W2K objects. W2K's IPSec implementation benefits from this greatly.

Although several policies can be created only one policy can be active. A policy is made up of a collection of filters (remember MS using the term filter rather than selector). Under W2K filters are objects. The same filter object can be assigned to multiple policies. Similarly, filter actions are objects. This allows an action definition to be assigned to multiple filters. There is still only one action per filter, but since actions are objects an action only needs to be defined once and can then be reused. It is strongly recommended to reuse filter action objects as much as possible.

6.2 Configuring IKE

IKE first negotiates a secure channel and then negotiates keys for IPSec protocols. In order to complete phase 1 negotiation IKE must negotiate a DH group. The IKE RFC defines 5 groups, but only requires the first one is implemented. The RFC also supports the creation of more groups later on.

1. Prime modulus – 768-bit (required to implement)
2. Prime modulus – 1024-bit (implemented in W2K)
3. Elliptic – 155-bit
4. Elliptic – 185-bit
5. Prime modulus – 1680-bit

W2K implements the first two as Low (1) and Medium (2). Group 3 is considered of equal strength as group 1 and group 4 is equal strength as group 2. However, groups 3 and 4 are typically quicker to calculate. Hopefully at some point in the future Microsoft will implement the remaining groups. Undoubtedly 3rd parties will provide additional DH groups to supplement the W2K offering.

Another important option negotiated by IKE is Perfect Forward Secrecy (PFS). Normally IKE phase 2 keys are derived from the phase 1 keys. If an attacker were to break the IKE phase 1 key he would be able to break all keys derived from that root key. PFS ensures that phase 1 and phase 2 keys do not share a relationship. Although this increases security it does so at great expense to performance. Use PFS only when absolutely necessary and make sure to set user expectations accordingly.

IKE must also negotiate the transforms that IPSec will use: MD5 or SHA1 HMAC, DES or 3DES cipher, and Kerberos or 3rd party certificate or pass phrase for authentication. Microsoft refers to the HMAC as a hash within W2K. It is always better to make a conscious decision of what security methods you will use rather than let Microsoft select for you.

6.3 Creating Filters (Selectors)

Selectors are at the heart of IPSec. As mentioned at the beginning of this section W2K defines selectors as objects. At first this can be confusing, but in the end this can save time and prevent errors.

Unlike W2K policies that only allow one to be active, multiple selectors can be defined per policy. Checking the box to the left of the selector enables it. If you wish to add additional selectors you can choose to either use the W2K wizard or do it manually. By default the wizard is launched. Unselecting the Wizard check box can change this. Although the selector is being created while editing a particular policy you are still creating a selector object and that object is not automatically included in the policy. So adding a new selector is a two-step process: create the selector object and then assign the selector object to the policy.

Treating selectors as objects has another important side effect. If you edit a selector within a policy you are editing the selector object! Any changes made to that object will be reflected in every policy that uses that selector. Make sure to give selectors descriptive names and fill in the description field. Above all, be very careful when editing existing selectors.

The choices are the same whether you are creating a new selector or editing an existing one. First you need to define where the packets come from. The general rule is be as specific as possible while still remaining flexible. Typically this means that either “My IP Address”, meaning all IP addresses on the system, or “Specific IP Address”, “Any IP Address”, or “Specific IP Subnet”. When possible avoid using DNS, DNS names are easily spoofed, and Specific IP Address, not very flexible.

Next you specify where the packets are received. The options are the same as above and the same recommendations apply. Remember: specific yet flexible. This is especially important if these selectors will be used in a group policy.

The final options for a selector are to define the protocol, if applicable what port number, and deciding if you wish to mirror the selector. W2K also has a protocol option of other that requires you to enter the protocol number. TCP communication is bi-directional and almost always requires the selector is mirrored. UDP is unidirectional and often does not require mirroring. W2K mirrors selectors by default.

6.4 Creating Actions

Once a packet is matched to a selector some action is performed. W2K implements these actions as objects that can be reused by all selectors. This reusability makes it critical that all actions are clearly named and the description field is exactly that: descriptive! As with selectors, if you edit an action you are editing all selectors that use that action.

There are only three possible actions for a selector: pass, block, and apply security. Applying security is where the fun comes in. Hopefully the action you want is already defined so you just have to check the appropriate box. If the action is not created then you will need to create one. Like with filters, the default action of W2K is to launch a wizard. Unchecking the Wizard box disables this.

At first glance W2K makes the choice of action appear simple: pass, block, or negotiate security. If the choice is to pass or block the action is done. If the choice is to negotiate security several other options become available. Probably of greatest importance is an option that you should NEVER check: Allow unsecured communication with non IPSec-aware computer. This option defeats the whole purpose of IPSec and basically says, if the devices can’t negotiate a security policy then send the communication in the clear.

When negotiating security W2K presents a simple set of selections: High (ESP), Medium (AH), and custom. As with IKE negotiation it is better to define your own actions than allow Microsoft to decide for you. Select custom. Once that is done many new selections appear.

The first two options are the same as the previous menu where you chose custom: AH and ESP. However, you now have the option to select the integrity algorithm: MD5, SHA1, or none. The thing to keep in mind when making this decision is that SHA1 is stronger than MD5 and is also slower.

Since ESP also offers confidentiality it offers those additional choices in addition to the same integrity selections AH offered: DES, 3DES, or none. 3DES is the strongest of the choices and also is slower. IPSec does not permit both integrity and encryption to use none – allowing that selection is pointless.

The final options available affect the vulnerability of keys by limiting their lifetime. This provides additional protection against replay attacks. The IPSec session lifetime can be measured in Kbytes, seconds, or both. By default these options are not checked. When they are checked they default to 100,000 Kbytes and 3600 seconds (1 hour). These are both reasonable and should only be changed if you have a strong need for greater security. Keep in mind that both of these settings affect performance. It is probably better to use PFS before reducing these values.

Actions also support multiple definitions. In other words, an action can specify to use ESP with SHA1 and 3DES, but if the other device doesn't support that try using ESP with MD5 and 3DES. The negotiation starts at the top selection and then works down the list. The selections can be matched up as required to provide extraordinary flexibility. Before exercising that flexibility think carefully about what goals you are trying to achieve. Simple is always better.

The final option for an action is the method of authentication to use. As mentioned previously, W2K supports three authentication methods: Kerberos, a Certificate from a CA, and pass phrase. Except when testing or troubleshooting you should not use pass phrase. The RFC requires its inclusion and discourages against using it at the same time. Unless every device that will use IPSec also shares the same Kerberos implementation you should select Certificate from a CA. You can create your own CA or use a 3rd party CA, which ever is most appropriate for your implementation.

6.5 Concluding W2K IPSec

This covers the most common implementations of IPSec using W2K. There are more options available, however. Tunnel mode was not discussed for instance. Although tunnel mode is extremely important in IPSec (a case study using IPSec tunnel mode is provided at the end of this document), it is not typically deployed on host-to-host communications. Tunnel mode is usually reserved for gateway implementations.

7 Problems, Pitfalls, and Solutions

When deployed on hosts, IPSec is typically simple to configure and causes few problems. One common problem under W2K is dynamic vs. static IPSec policies. When deployed on gateways two other problems may occur. The first problem relates to packet size and packet fragmentation, the second problem is with ICMP (ping) packets. Fortunately, all of these problems have a solution. This is followed by a short list of W2K troubleshooting tools. The section ends with a discussion on how IPSec defends itself against attack.

7.1 Dynamic vs. Static Policies

IPSec Policies applied through an AD policy or through the command line tool IPSecPol.EXE creates dynamic IPSec policies. Dynamic policies are removed from the host when the IPSec service is restarted. Although this has potential uses, typically this is a problem. The solution is to use static policies. The details of this are beyond the scope of this document, however section 7.4 provides some information on the tool. More complete documentation can be found in Jason Fossen's "Windows 2000: IPSec, RRAS, and VPNs".

Currently Group Policies only support dynamic policies. Only IPSecPol.exe supports static policies (policies that survive a reboot or restarting the IPSec service). Microsoft intends to provide a solution for applying static IPSec policies using AD group policies. Until that time, **it is strongly recommended that all IPSec deployments on a server use an IPSecPol script to create a static IPSec policy.** Although this means not using the powerful tool of group policies it is a necessary step until Microsoft provides a solution. Group policy deployment is probably adequate for clients, however.

Another recommendation is to put reciprocal policies on the appropriate servers. For example, if Server A should not communicate with Server B then create a selector on Server A that blocks packets from Server B and a selector on Server B that blocks packets from Server A. In this way, if an attacker gains control of Server A and removes the IPSec policies, Server A is still prevented from communicating to Server B. This is especially important for servers in a DMZ or that contain sensitive information.

7.2 MTU Size

When a gateway uses IPSec the Maximum Transmittable Unit (MTU), the largest packet size in Bytes, can cause problems. On a LAN most hosts will use a 1518 Byte MTU size. Since AH and ESP use a portion of that space either the MTU size must grow or the amount of data in the packet must shrink. On Ethernet networks 1518 Bytes is the largest MTU value allowed, so the only solution is for the packet to shrink.

One method of reducing the amount of data in a packet is to allow packet fragmentation. Unfortunately, not all operating systems allow packet fragmentation. For instance, all versions of Windows do not allow packet fragmentation. The only other option is to reduce the MTU size on any host traveling through the gateway. If the gateway supports Path MTU (PMTU) discovery then most operating systems, including Windows, will automatically discover the correct MTU size. Otherwise, each host must reduce its MTU size.

In the event that it is not acceptable to reduce the MTU size on every client RFC 2401 specifies a solution: An IPSec gateway must process PMTU messages. The DF must still be honored; however, PMTU messages must be forwarded so the sender is informed of an MTU problem. Under W2K, the MTU size is reduced to the size specified in the ICMP PMTU message and the packet is resent.

Alternatively, the IPSec gateway may fragment the IPSec packet, even though the encapsulated packet does not permit fragmentation, to fit the MTU size and send the resulting packets forward. In some cases this solution provides better throughput. This solution also permits blocking of all ICMP.

7.3 ICMP

ICMP also can cause problems for an IPSec enabled gateway. This is because IPSec may overwrite a critical portion of the ICMP packet. RFC 2401 is also specific about how to resolve this problem. It states that an IPSec gateway "**SHOULD** be processed and forwarded in a tunnel mode SA." This handles ICMP error messages generated by AH or ESP protected packets. A non-protected ICMP error message that must traverse an IPSec tunnel to get back to the source host is handled by another portion of RFC 2401. In the end, RFC 2401 provides mechanisms that either must or should be implemented by an IPSec gateway that allow a host to receive the proper ICMP error messages.

7.4 W2K Tools

There are many other problems that could arise in an IPSec network. Microsoft provides tools to assist in troubleshooting those problems. Below is a list of some of those tools. (Jason Fossen "Windows 2000: IPSec, RRAS, and VPNs")

- IPSecmon.exe
- Netdiag.exe
- IPSecPol.exe

<u>IPSecMon</u>

IPSecMon is a graphical tool that can monitor IPSec statistics. It provides information on many aspects of an IPSec communication: Security Associations (SA), various types Bytes sent, Oakley mode in use, and some failure information. Some of these statistics are not discussed in this document and would require further research (SA's for instance). It is an excellent tool to use for difficult to track down problems or when W2K is configured as an IPSec gateway or tunnel mode is being used. The only user configurable option in this tool is the refresh frequency for the displayed statistics.

This is an excellent tool to determine if an IPSec negotiation is failing; possibly indicating that the IPSec configurations on the devices are not compatible. For instance, if once device specifies only the use of a 3DES cipher is acceptable and another device only permits DES, then these two devices will not successfully communicate. IPSecMon would show this failure.

<u>Netdiag</u>

NetDiag comes with the Windows 2000 support tools. Unfortunately, there's not a lot of information on the tool. NetDiag /? from a command prompt displays a help window. NetDiag performs diagnostic tests and the command is netdiag /test:ipsec /debug. This also creates a log file in the directory that the command was run called NetDiag.Log. Inside the log file are the IPSec statistics that resulted from the diagnostics performed.

IPSecPol

IPSecPol is a command line tool that comes with the W2K Resource kit. This tool is able to create and configure IPSec policies from the command line. The most valuable aspect of the tool is its ability to create static policies (as previously discussed in Dynamic vs. Static Policies). The one draw back is how cryptic the tool is to use. However, it does have a very good help screen. Jason Fossen also provides an excellent tutorial on how to use the tool in Windows 2000: IPSec, RRAS and VPNs.

7.5 Protecting IPSec

There are three defensive measures that IPSec takes special precautions to prevent: Denial of Service (DoS) attacks, limiting exposure if a key is cracked, and preventing replay attacks. The first problem, DoS, uses a devices very own security apparatus against it. The second limits the amount of damage if a code is broken. The third prevents an attacker from breaking an old code offline, modifying the message, and then replaying the modified message.

DoS attacks that use a defender’s security system work on the principle that it is less costly to send the attack than it is to perform a security function on the attacker’s packets. To defend against this IPSec must carefully and quickly choose when to perform expensive security operations. If IPSec can identify a packet as not requiring expensive security operations before it performs those operations then it has a good chance at keeping up with a DoS attack. To help prevent DoS attacks, the IPSec receiver calculates a HMAC for the packet. If the result matches the HMAC the sender placed in the packet (in clear text) then IPSec proceeds with decrypting the packet. If the hash values do not match the packet is discarded. This prevents IPSec from performing expensive decryption.

IPSec has two methods of limiting the damage of a cracked key. First, it requires that all keys are regenerated over time. Second, IPSec permits the regenerated key to be created without dependence on the previous key. This means that if an attacker were to crack a key they would be unable to use that knowledge to crack subsequent keys. This feature is called Perfect Forward Secrecy (PFS). Although PFS provides excellent security it is computationally expensive.

IPSec also provides anti-replay protection. This is achieved by providing a sliding window. IPSec assigns a sequence number to each packet. The sliding window only permits so many packets to be outstanding. So a packet number 1 may be good after the packets 2 – 64 have arrived, but as soon as packet number 65 arrives packet 1 is no longer accepted. Then when packet 66 arrives packet 2 is no longer accepted. In this example, the window is 64 packets wide.

This creates an opportunity for an attacker to move the sliding window far enough that IPSec will drop packets. In response, IPSec doesn't move the window until it verifies that the packet is valid. For example, packet number 1 is not considered invalid until packet 65 (or higher) arrives and is determined to be a valid packet.

8 Case Studies

Two case studies are provided. The first uses IPSec as a packet filter. The second implements an IPSec gateway. These two examples exemplify the flexibility of IPSec. The packet filter case study provides a fairly simple implementation. Contrasting the simple IPSec packet filter is the IPSec gateway implementation with somewhat conflicting requirements.

8.1 IPSec Packet Filtering

The network consists of a web server that is multi-homed, one interface in the DMZ and the other a "Services" segment, an LDAP server, and an MS SQL database. The LDAP and MS SQL database reside in the Services. In order to authenticate users the web server communicates to the LDAP server and the LDAP server then queries a SQL database. User account information is kept in the MS SQL database. The LDAP and SQL servers are single homed.

Requirements:

- Prevent web server from communicating to an MS SQL database
- Only allow TCP 80 on the web servers' DMZ interface.
- Prevent MS SQL database from communicating to web server

Solution: Implement IPSec packet filtering on the web and SQL servers.

In order to make the IPSec implementation robust the web servers have IPSec rules preventing it from accepting any communication from the SQL server and the SQL server has IPSec rules preventing it from accepting any communication from the web servers. This mutual exclusion is an extra layer of protection in the event that one of the servers is compromised. Also, server IP addresses in the Services segment are clumped into supernet ranges to simplify IPSec rules.

The DMZ segment uses 64.1.1.0 network and the services segment are in the 192.2.2.0 network. Supernetting is used for IPSec packet filtering rules only. This allows additional servers to be added without modifying existing IPSec policy – only the new server needs to add an IPSec policy. Of course in a real network, planning is needed to ensure that enough IP addresses are available to each server type.

Network Information

Server Type	IP Range	IPSec Network Masks
Network/Security	192.2.2.1 - 14	192.2.2.0/28

Web	192.2.2.64 - 126	192.2.2.64/26
LDAP	192.2.2.192 – 206	192.2.2.192/28
SQL	192.2.2.240 – 254	192.2.2.192/28

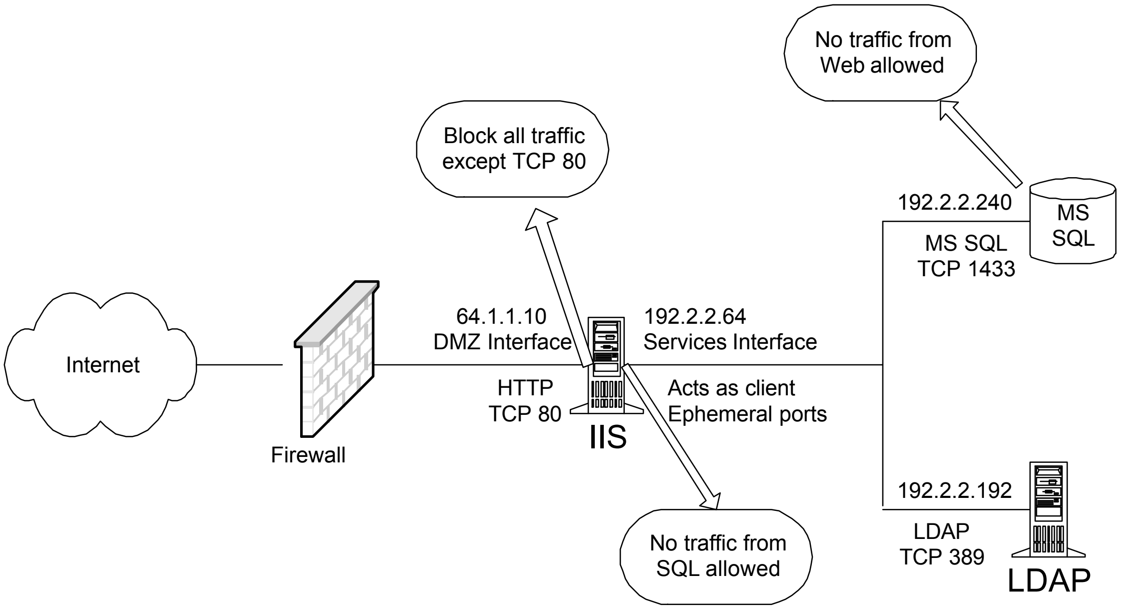
Web Server

From IP	Mask	From Port	To IP	To Port	Protocol	Action
Any		Any	64.1.1.10	Any	Any	Block
Any		Any	64.1.1.10	80	TCP	Permit
192.2.2.240	255.255.255.240	Any	My IP	Any	Any	Block

SQL

From IP	Mask	From Port	To IP	To Port	Protocol	Action
192.2.2.64	255.255.255.64	Any	My IP	Any	Any	Block

The web server needs three rules to accomplish two tasks. The first two rules permit only TCP port 80 (standard HTTP) and the third rule blocks any traffic from the SQL server (My IP denotes all IP addresses on the host). It is important to remember that the most specific IPSec rule is applied. This is what allows the first rule, which says to block any traffic sent to the DMZ interface, and the second rule, allow any IP address to communicate to TCP port 80, to work together. The SQL server, in contrast, only requires one rule. That rule says to block any packets received from the web server. Below is a diagram of the environment.


No traffic from Web allowed
Block all traffic except TCP 80
192.2.2.240
MS SQL
MS SQL TCP 1433
Internet
Firewall
64.1.1.10 DMZ Interface
HTTP TCP 80
IIS
192.2.2.64 Services Interface
Acts as client Ephemeral ports
192.2.2.192
LDAP TCP 389
LDAP
No traffic from SQL allowed

As practice, what additional IPSec rules are necessary to only permit TCP 389 from web server to LDAP server? The LDAP server is single homed and has the IP Address of 192.2.2.192. Remember to create a flexible rule that would work even if additional LDAP or web servers were added at a later date (Hint: review the Network Information table). The answer is in Appendix A.

8.2 IPSec Gateway

Clients and W2K servers at an international site must access corporate network resources, including W2K domain controllers, W2K and NT4 servers, Unix servers, and Oracle databases running on Unix. Although the international site is trusted the communication channel between corporate and the remote site is not. Additionally, the client should not have to authenticate separately to access corporate resources. The final hurdle is that security requires Network IDS to be able to analyze this traffic before it enters the corporate network.

Requirement:

- Transparent to client
- Data confidentiality
- Cross-platform support (various flavors of Unix, NT, and W2K)
- RPC, NetBIOS, SMB, and other insecure communication will occur between sites
- Packets must be unencrypted prior to entering corporate network for analysis by Network IDS
- Corporate firewall is application proxy based (key word is proxy)

Solution: Implement IPSec gateways at the borders of the international site and the corporate network.

IPSec gateways are put in place at the International site and at the border to the corporate network. The corporate IPSec firewall sits between two security devices. In order to keep the design simple, an IPSec router is used for the gateway. Outside of the corporate IPSec router is an external router with ACLs and in front of the IPSec router is an application proxy firewall. On the same segment as the IPSec router and the internal firewall resides a Network IDS.

The two IPSec routers will use ESP in tunnel mode to provide integrity, confidentiality, anti-replay, and authentication. The external router permits UDP 500 and protocol 50 to pass through. The internal firewall allows any traffic from the International network IP ranges to travel into the corporate network. In essence, we are using IPSec for VPN connectivity.

Network Information

Device	IP Address
International office network	10.2.0.0/16
International IPSec router	External interface: 64.2.2.2/16
Corporate IPSec router	External interface: 64.1.1.1/16
Corporate network	10.1.0.0/16

International Gateway

From IP	From Port	To IP	To Port	Protocol	Action
64.1.1.1	Any	64.2.2.2	Any	Any	ESP tunnel mode
64.2.2.2	Any	64.1.1.1	Any	Any	ESP tunnel mode
Any	Any	Any	All	Any	Block

Corporate Gateway

From IP	From Port	To IP	To Port	Protocol	Action
64.2.2.2	Any	64.1.1.1	Any	Any	ESP tunnel mode
64.1.1.1	Any	64.2.2.2	Any	Any	ESP tunnel mode
Any	Any	Any	All	Any	Block

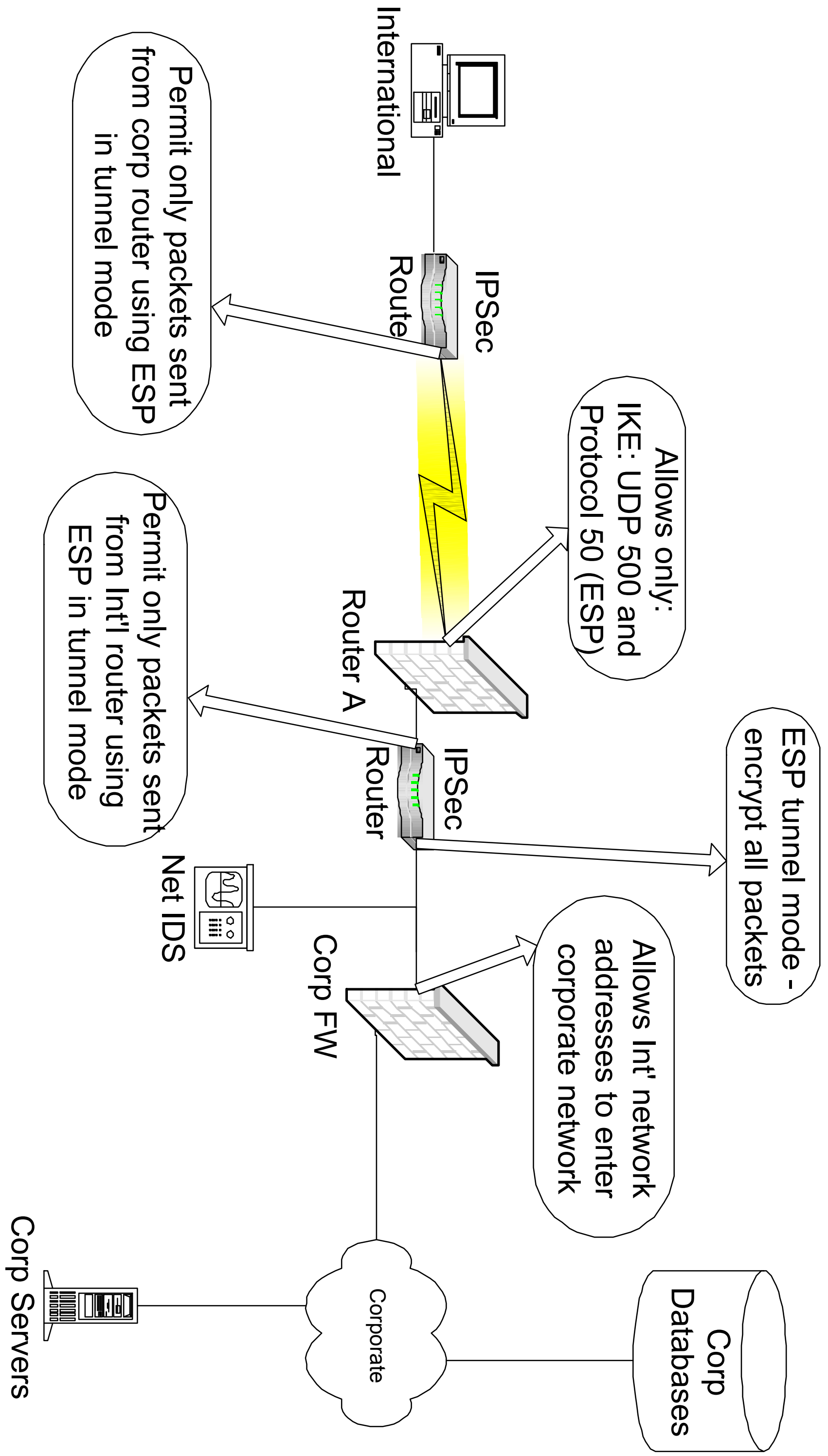
In this scenario, only two rules are needed on each side. The first two rules for both IPSec routers allow traffic from its partner only if ESP in tunnel mode is used. The third rule blocks all other traffic. In a real-world scenario these routers may also have to pass other traffic, but for now we are assuming that no other communication is allowed. You may also notice that this example shows the mirrored selector (the second selector for each router). Creating the mirrored selector is a feature found in W2K that is not found in most gateway implementations.

It's important that any security device external to these IPSec routers permit UDP 500 and protocol number 50 to pass through. UDP 500 is used by IKE and protocol number 51 is ESP. In this case study, a separate router with an Access Control List (ACL) is external to the IPSec router. In the real world the functions of these two routers may be combined into one router. This case study uses two routers for illustrative purposes.

The IPSec router is external to a corporate side firewall to prevent untrusted traffic from entering the corporate network until a Network IDS has had a chance to inspect the traffic. It is for that reason the Network IDS has to reside on the same switch as the IPSec router and the corporate firewall. Because it is also a proxy, it is important that the traffic is unencrypted before getting to the internal firewall.

As mentioned under Problems, Pitfalls, and Solutions, section 7.2, IPSec gateways face some unique problems. To briefly recap, IPSec places additional overhead on the packet reducing the space available for payload. We have two choices to solve this problem: forward PMTU packets to the sending host or permit the IPSec packet that is encapsulating the original packet to be fragmented. In this scenario, the router will be configured to fragment IPSec packets. This should allow for optimum network efficiency, especially if there are any dial-up users at either site (dial-up typically provides a 576B or less MTU size).

International IPSec Implementation


International
IPSec
Route
Permit only packets sent from corp router using ESP in tunnel mode
Allows only:
IKE: UDP 500 and Protocol 50 (ESP)
Router A
Permit only packets sent from Int'l router using ESP in tunnel mode
IPSec
Router
ESP tunnel mode - encrypt all packets
Net IDS
Corp FW
Allows Int' network addresses to enter corporate network
Corporate
Corp Servers
Corp Databases

As practice, add IPSec selectors that will allow internal clients to use HTTP to the Internet. See Appendix B for a solution.

Created by: Scott Cleven-Mulcahy; Last saved on: 7/18/2001 9:08 AM; Printed on: 7/17/2001 12:48 PM

Appendix A

Four additional selectors will permit only TCP port 389 traffic between Web and LDAP.

Web Server

From IP	Mask	From Port	To IP	To Port	Protocol	Action
192.2.2.192	255.255.255.240	Any	192.2.2.64	Any	Any	Block
192.2.2.192	255.255.255.240	389	192.2.2.64	Any	TCP	Permit

LDAP

From IP	Mask	From Port	To IP	To Port	Protocol	Action
192.2.2.64	255.255.255.192	Any	My IP	Any	Any	Block
192.2.2.64	255.255.255.192	Any	My IP	389	TCP	Permit

The first web server rule blocks all traffic from the LDAP server IP range. The second rule permits only TCP 389 from the LDAP. Since the web server acts as a client to the LDAP server we are unable to specify a port the LDAP will communicate with.

Similarly, the first LDAP server rule blocks all traffic from the web server IP range. The second rule permits the web server to connect to the LDAP server on TCP 389. Remember, the web server is a client and clients use ephemeral ports above 1023. This prevents us from specifying what port the web server will use to connect to TCP 389.

Appendix B

To allow HTTP traffic to pass unencrypted through the IPSec router requires TCP port 80 traffic from the internal network be permitted to pass through the router. Thus the two selectors remember most gateways do not automatically create mirrored selectors, permit outbound traffic to TCP 80 and inbound from TCP 80.

Also notice how much more information is necessary to allow the internal network to access web servers on the Internet. Although "From IP" could be "Any" for IPSec and HTTP, it is more prudent to specify the internal IP Network address as the source. This requires net mask information as well. If any were used as the source IP address then this would also permit any Internet address to access your internal network over TCP port 80. Most likely this is not what is desired and this would be considered a serious security vulnerability.

International Gateway

From IP	Mask	From Port	To IP	To Port	Protocol	Action
10.2.0.0	255.255.0.0	Any	Any	80	TCP	Permit
Any	255.255.0.0	80	10.2.0.0	Any	TCP	Permit

Corporate Gateway

From IP	Mask	From Port	To IP	To Port	Protocol	Action
10.1.0.0	255.255.0.0	Any	Any	80	TCP	Permit
Any	255.255.0.0	80	10.1.0.0	Any	TCP	Permit

Special Note

The requirement to use Network IDS seems to exclude the use of ESP. In this case the traffic went through a gateway and was decrypted prior to reaching the NIDS. If there was no gateway or firewall, ESP in transport mode could be used and still allow the NIDS to inspect traffic! ESP supports the use of encryption but does not require it. Although deploying ESP in such a way would be unusual, it would work. The use of a NAT, as long as port translation was not also done, is also supported. IP spoofing may not be a problem either. ESP does authenticate the transport layer and if a packet's IP address from an existing connection were spoofed an IPSec SA would not exist for this connection, which would cause the packet to be dropped. (SA's were not discussed in this tutorial.) The point of all this is that IPSec is flexible. That flexibility permits creative solutions to problems that may not have had a solution before.

List of References

Fossen, Jason. Windows 2000: IPSec, RRAS, and VPNs. Baltimore: SANS 2001, 12 – 83

Doraswamy, Naganand and Harkins, Dan. IPSec Architecture: The new Security Standard for the Internet, Intranets, and Virtual Private Networks. Upper Saddle River: Prentice Hall, 1999

RFC 2401: IETF. Security Architecture for Internet Protocol.
URL http://www.rfc-editor.org/rfc/rfc2401.txt

RFC 2402: IETF. IP Authentication Header (AH).
URL http://www.rfc-editor.org/rfc/rfc2402.txt

RFC 2403: IETF. The Use of HMAC-MD5-96 within ESP and AH.
URL http://www.rfc-editor.org/rfc/rfc2403.txt

RFC 2404: IETF. The Use of HMAC-SHA1-96 within ESP and AH.
URL http://www.rfc-editor.org/rfc/rfc2404.txt

RFC 2405: IETF. The ESP DES-CBC Cipher Algorithm.
URL http://www.rfc-editor.org/rfc/rfc2405.txt

RFC 2406: IETF. IP Encapsulating Security Payload (ESP).
URL http://www.rfc-editor.org/rfc/rfc2406.txt

RFC 2409: IETF. The Internet Key Exchange (IKE).
URL http://www.rfc-editor.org/rfc/rfc2409.txt

Pirooz , Hamid, SANS. Diffie-Hellman Public Key Distribution Scheme: A Complete Overview. URL http://www.sans.org/infosecFAQ/encryption/diffie.htm

Feier, Alan, Karlton, Philip, and Kocher, Paul, Netscape. The SSL Protocol Version 3.0
URL http://home.netscape.com/eng/ssl3/ssl-toc.html

RFC 2246: IETF. The TLS Protocol Version 1.0.
URL http://www.rfc-editor.org/rfc/rfc2246.txt

Microsoft Corporation. Diagnosis and Treatment of Black Hole Routers
URL http://support.microsoft.com/support/kb/articles/Q159/2/11.asp

Microsoft Corporation. Basic IPSec Troubleshooting in Windows 2000
URL http://support.microsoft.com/support/kb/articles/Q257/2/25.ASP

Cisco Systems, Inc. Sample Configuration: IPSec Tunnel through a Firewall with NAT
URL http://www.cisco.com/warp/public/707/ipsecnat.html

Cisco Systems, Inc. Configuring Router-to-Router Dynamic-to-Static IPSec with NAT
URL http://www.cisco.com/warp/public/707/ios_804.html