Global Information Assurance Certification Paper

What is IPsec?

By

Luis Rosello

SANS Security Essentials

GSEC

Practical Assignment Version 1.3

16 April 2002

Many of you may already know what it is, but many of you may not. This paper presents high-level concepts and is targeted to those of you who do not know what IPsec means or who have heard of it, or even use it, but are not familiar with its inner workings.
Generally speaking, data on a vanilla IP network is visible to anyone, could come from anywhere, and might be modified to say anything, unless there are intentional and proactive safeguards in place. Most people using IP networks are not aware of this issue and assume that they can trust the origin and content of the data they work with, and that their transactions, if not secret, are reasonably private.

The IP protocol

In order to understand how IPsec works, we first need to understand the foundation blocks that IPsec builds upon. Let's start with a basic overview of the IP protocol, the Internet's universal medium of communications.
The IP protocol works at layer 3 (network) of the OSI 7 layer model (ISO 7498). IP routes packets, reads information in the header of these packets, and it finds a way over the Internet in which to route them to their final destination. It is responsible for routing through a network and for network addressing. IP outputs packets called "datagrams," and each datagram is prefixed with an IP header that contains source and destination IP addresses. If IP has to fragment the packet further, it creates multiple datagrams with sequence numbers, so that they can be reassembled by IP on the other (receiving) end.
All computers around the world that communicate on the Internet do so via IP. Higher-level applications (telnet, FTP) can be different; lower-level protocols (Ethernet, Token Ring) can be different; but IP is the *one* thing, the common language that makes all communications possible. A typical IP packet is illustrated in figure 1.

Figure 1: Typical TCP/IP packet

IPsec technology provides security for traffic at the IP layer in both the IPv.4 and IPv.6 environments. For the sake of simplicity, all references from now on to the IP protocol and its headers will apply to IPv.4. Also, in order to simplify the explanation of IPsec, from now on I will refer exclusively to the TCP/IP 4 layer model, not the OSI 7 layer model. To see how the TCP/IP 4 layer stack compares with the OSI 7 layer stack, refer to Figure 2 below:

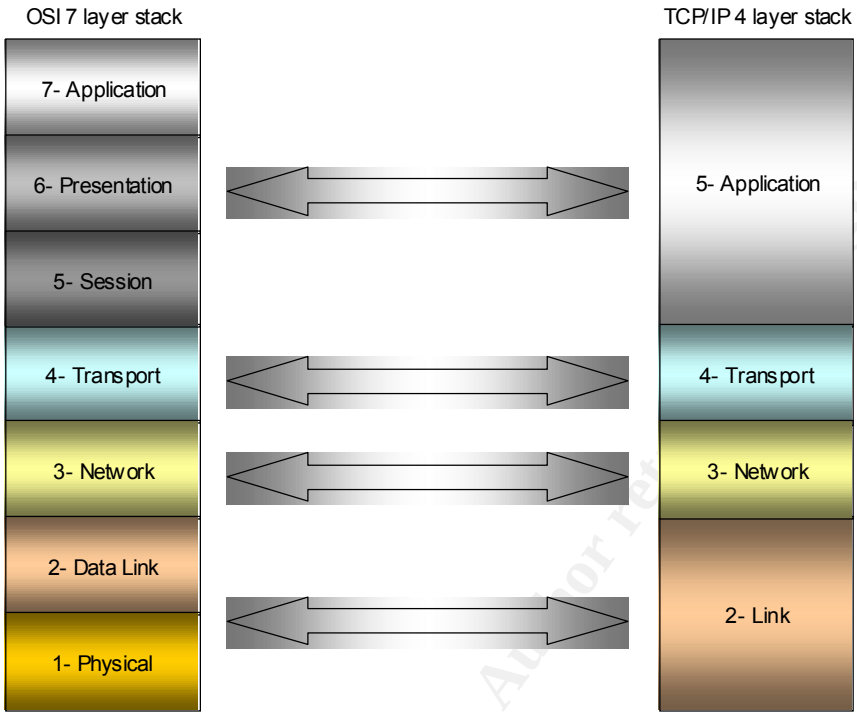

Figure 2: comparison of the OSI and TCP/IP protocol stacks

What is IPsec?

IP stands for Internet Protocol (part of the TCP/IP suit), and sec is the abbreviation for security. Hence, you can deduce that it has to do with secure IP. That's exactly what the designers of IPsec had in mind for its main purpose. Many Internet engineers were concerned with the viability of IPv4, and many others were concerned and involved with addressing security for the new, upcoming protocol, IPv6.
The Internet Engineering Task Force (IETF) developed IPsec as a hybrid, both a retrofit to address many flaws and vulnerabilities inherent in IPv4 and a preemptive strike against insecurity in IPv6. IPsec was also designed to have an open, modular architecture. This modularity allows it to evolve to address new requirements, new cryptographic technologies, and newly identified problems with existing security mechanisms.
So now that we know what IPsec is, and we also know where these inherent vulnerabilities in existing IP implementations exist, you say, that's it! Secure the IP protocol, and you can secure **all data** traversing the wire. And I say, yes, that is the basic idea behind IPsec.

The threats

As we said earlier, there are certain security risks or vulnerabilities inherent in the IP protocol. This is because the Internet Protocol, by design, is a very

"informative" thing. It needs to advertise source and destination information for routing to occur, and its payload is easily read by anybody motivated enough to eavesdrop. Widely available tools such as "sniffers" allow anybody with access to the transmission media to "read" the data contained within these packets. Security risks specific to IP networks fall into two broad categories:

- Data theft
- Data tampering

In today's world of e-mail, e-commerce, e-trading, e-business, e-bills, e-cash, e-books, e-forms, e-cards, e-learning, e-Bay and every possible e-(fill in the blank) you can imagine, the confidentiality of some data transmitted can be extremely valuable. Compromising of such data could lead a company to bankruptcy. Just imagine if the blueprints of a revolutionary new car design were transmitted from corporate headquarters to the production plant via FTP, and the competition was "listening" to the transmission and gained this invaluable data. These potential security breaches are the main reason why leased private lines are used to connect company's remote campuses (As we will see latter, IPsec through VPN implementations does away with this need). And although other means of protection at higher layers exist for certain types of transmissions, such as SSL for HTTP or PGP/Web-of-Trust for e-mail encryption, these technologies are limited to their particular applications.

How do I make it secure?

As described in RFC 2401, "IPsec provides security services at the IP layer by enabling a system to select required security protocols, determine the algorithm(s) to use for the service(s), and put in place any cryptographic keys required to provide the requested services." In other words, IPsec ensures that you are receiving original data from the claimed sender, and that the data was not "looked at" or inspected by somebody else. This protection is achieved through services including data origin authentication, connectionless data integrity authentication, data content confidentiality, anti-replay protection, and limited traffic flow confidentiality. These mechanisms and components are explained in greater detail in the following sections.

Security Policy Database

The Security Policy Database (SPD) defines IPsec's protection requirements. This database must have an administrative interface to allow management, either by the system administrator or an application acting as such. The SPD is consulted for processing of all traffic, inbound and outbound, and applies 3 basic processing modes to it based on IP and transport layer information, and its policies. It will:

- Discard —will not let this packet in or out.
- Bypass —will not apply security services to an outbound packet and will not expect security on an inbound packet.
- Apply —will apply security services on outbound packets and will require

inbound packets to have security services applied.
The SPD policy entries must be ordered to ensure predictable, consistent processing. As described in RFC 2401, "This requirement is necessary as the effect of processing traffic against SPD entries must be deterministic, but there is no way to canonicalize SPD entries given the use of wildcards for some selectors."
IPsec must have access to the IP source code to implement security on IP traffic. These implementations are known as "Bump-in-the-stack" (BITS), and "Bump-in-the-wire" (BITW). BITS does not require access to IP source code. It implements IPsec under the existing IP protocol stack, between the native IP and the local network drivers, and is usually employed by hosts. Either a host or a gateway can implement BITW. It mostly consists of a pass-through encrypting unit that functions transparently to the rest of the network (gateway), or could be quite analogous to BITS when supporting a single host.
IPsec utilizes two protocols to provide these services; they are the Encapsulating Security Payload and the Authentication Header.

Encapsulating Security Payload:

ESP provides confidentiality, data integrity, and data source authentication for IP traffic and also provides protection against replay attacks. The actual set of services provided will depend on the Security Associations (SAs) established. The ESP header contains a Security Parameter Index (SPI) to help locate the SA with which the packet is processed. At this point, it may not be clear what this means, but as we will see, ESP implements mechanisms to address all these requirements. To understand how ESP works, refer to the typical TCP/IP packet in figure 1. As you can see, the packet is built with the lower-level protocols on the outside and higher-level protocols closer to the inside. In a typical transmission scenario, the application hands the data to the transport layer where a virtual connection, reliable (TCP) or unreliable (UDP), is created. (To learn more about transport protocols, refer to RFCs 768, 793, 1180, and other pertinent material.)
ESP jumps in at this point. It takes that original packet that would otherwise be handed down the stack for transmission, and it encapsulates the data and layer 4 protocol with encryption (confidentiality) as one, in a new packet. This "new" packet is handed down the stack, and the process continues as usual. Authentication for ESP is optional and is achieved by a creating a "hash," or digital signature of the data encrypted. The receiving host can then use the hash to verify the data's authenticity.
At this point, we can see that the packet is still routable since the IP header is not encrypted. This feature allows IPsec to be implemented between two hosts, or gateways, without having to worry if the devices in between are IPsec compliant. You can also see (Figure 3 and 5) that all higher level (layer 4 and higher) protocols are protected (encrypted) by ESP, which ensures that if somebody were to sniff the connection, it would be impossible (without cracking the encryption) to gain any useful information. I should mention that ESP attaches

both a header and trailer to the datagram. Other than the fact that Authentication and Encryption is applied to the packet, the details pertaining to these components are beyond the scope of this paper. Please refer to RFC 2406 for more information on ESP. The encryption algorithms used and other settings are exchanged during the IKE SA negotiation phase, more about that latter….
The anti-replay service is dependent on the data origin authentication service and is enabled by default. Sequence numbers are mandatory and part of the ESP header, but since it's up to the receiver to implement this service (enabled by default), anti-replay services can be disabled.

Authentication Header:

AH, just like ESP provides connectionless integrity, data source authentication, and optional protection against replay attacks, but it does not afford confidentiality (encryption). It accomplishes these requirements by computing a cryptographic function (signature) for the packet utilizing a secret authentication key. This signature is used by the recipient to verify the identity of the data's owner and the data's integrity. AH affords security to as much of the IP header as possible (unlike ESP), as well as for all upper layer protocols. But since some fields in the IP header may change in transit, AH exempts these fields from protection. The AH header, like the ESP header, contains an SPI to help locate the SA with which the packet is processed.
Anti-replay services, just as with ESP, are enabled by default but are only effective when the receiver checks the mandatory sequence numbers in the AH header. Since AH does not encrypt the packet, you can imagine that it is a much simpler and faster protocol to implement than ESP. AH can be implemented alone, combined with ESP, or nested in tunnel mode.
Details regarding the AH header are beyond the scope of this paper. Please refer to RFC 2402 for more information on AH.

Security Associations:

Most all published material refers to SAs as "contracts." These contracts determine the terms for communication between two hosts, two gateways, or a host and a gateway. They define what AH authentication algorithm's mode and keys to use, the ESP encryption algorithms and keys to them, the cryptographic synchronization for those algorithms, protocols for authentication, keys and their lifetimes, and even the lifetime of the SA itself. As you can see, this level of detail allows for the customization of the SA to fit the level of security that you need for a determined connection.
RFC 2401 mandates support for both manual and automated SA and cryptographic key management. Manual management is the simplest; in this case the security administrator or an application acting as such configures all settings. Automated SA management creates Internet-standard, scalable protocols needed for widespread IPsec deployment. The default automated key management protocol selected for use with IPsec is IKE. Many key management

protocols exist, but for simplicity, I will only refer to the default, IKE.
As we mentioned before, both the AH header, like the ESP header, contains a Security Parameter Index (SPI) to help locate the SA with which the packet will be processed. The SPI is a mandatory, arbitrary 32 bit number selected by the destination system that along with the destination IP address identify the SA to be used.
Implementations of IPsec always build and maintain an SA Database (SAD), where these parameters associated with the SA are defined. SAs are unidirectional, "simplex" connections. When two hosts create a secure connection, two SAs are utilized by each host, an inbound SA and an outbound SA.

Modes: Transport and Tunnel
ESP and AH can operate in one of two modes. In transport mode the protocol operates primarily on the payload of the original datagram. In tunnel mode the protocol encapsulates the original datagram in a new one, treating the original as the data payload. Tunnel mode is mostly used by gateways to transmit data from a host with a non routable IP address securely over the internet, creating what is know as Virtual Private Networks (VPNs).
Modes are defined by Security Associations (SA).

Transport mode:

Transport mode is defined as an SA between two hosts since it can only protect packets where the communications and cryptographic endpoints are the same. ESP transport mode provides protection to upper layer protocols as illustrated in figure 3. AH, just like ESP transport mode, protects the packets payload, but it extends the protection to the IP header (excluding certain fields) as illustrated in figure 4.
In Transport mode, when both AH and ESP are applied to a packet, it is very important that ESP is applied first. From what we learned earlier, we see that if the packet is first protected with AH and then ESP, then the data integrity is applicable only for the transport payload as the ESP header is added later on. This is obviously undesirable because the data integrity should be calculated over as much data as possible, including the ESP header.

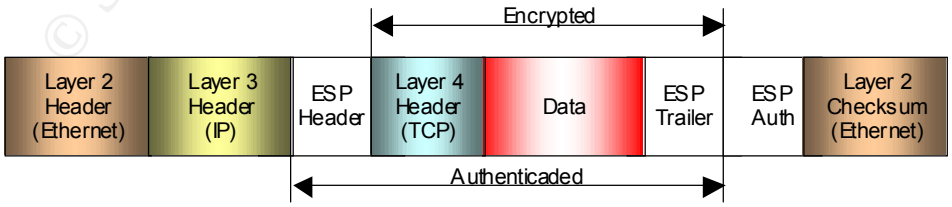

Figure 3: Packet protected by ESP in Transport mode

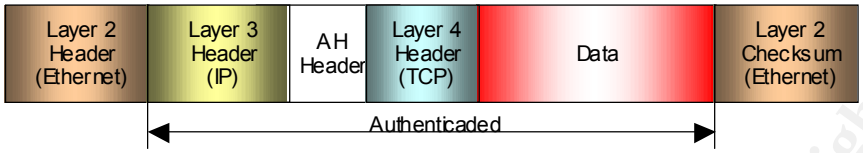

Figure 4: Packet protected by AH in Transport mode

Tunnel Mode:

Tunnel mode is defined as an SA between two gateways or a gateway and a host. In short, in a connection where one end is a gateway (an implementation of IPsec on a firewall of router; i.e. a *security* termination point), the SA must be tunnel mode. Transport mode to a gateway is allowed only when the gateway is acting as a host to the traffic, not transiting the traffic. Tunnel mode packets are characterized by having two headers (refer to figures 5 and 6). The inner header is the original IP header created by the host. The outer header is created by the sending gateway and contains routing information to get the packet to the receiving gateway. Once the receiving gateway gets the packet, it strips the outermost IP headers, processes the data as required, and sends the packets to their final destination, usually on a second interface connected to a private network. This implementation is the foundation for Virtual Private Networks (VPNs).

A VPN device (gateway) usually separates a corporation's private network from the public network (Internet). It grants access to certain connections based on their corresponding SAs as explained earlier. Packets originating on hosts in private networks are non routable (refer to RFC 1597). When the gateway receives a packet destined for the remote, non-routable network, it relies on its IPsec implementation to get these packets to their intended recipients. This process is completely transparent to the hosts at either end.

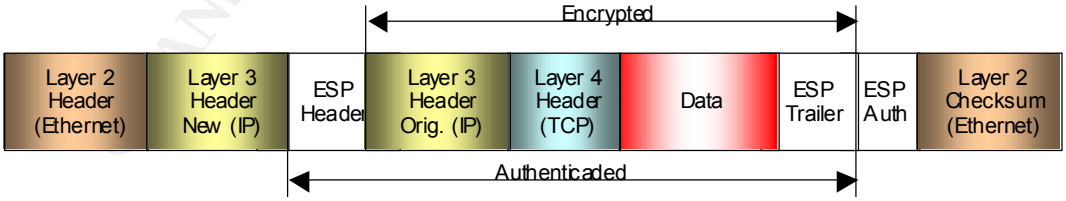

Figure 5: Packet protected by ESP in Tunnel mode

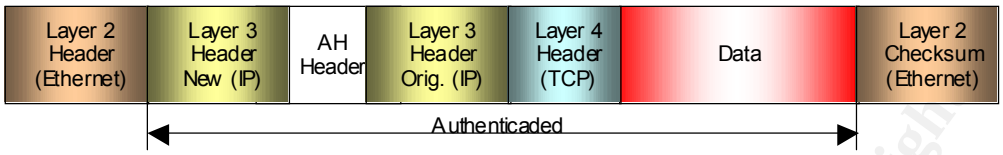

Figure 6: Packet protected by AH in Tunnel mode

Cryptography

Cryptography is the soul of IPsec. Without cryptography IPsec, or for that matter any other security protocol that utilized a public network (like the Internet) for transmission, would be rendered useless. It is also one of the most complex fields in computing, and although you don't need to understand it to implement IPsec, it's worth expanding on the basic encrypting mechanisms that make IPsec work.

Cryptography is defined as "the conversion of data into a secret code for transmission over a public network. The original text, or 'plaintext,' is converted into a coded equivalent called 'ciphertext' via an encryption algorithm. The ciphertext is decoded (decrypted) at the receiving end and turned back into plaintext."[1] In expanding on what was explained earlier, and in the context of IPsec, symmetric encryption provides data confidentiality (with some data integrity); public key encryption enables strong device and data authentication; and hash signatures provide high speed data integrity checks. Symmetric encryption occurs when the same key is used both to encrypt and decrypt the data. The most common symmetric encryption algorithm is Data Encryption Standard (DES). With today's computers getting faster and cheaper, this algorithms 56-bit key is becoming easy prey to brute force attacks. This vulnerability has been addressed by the newer "triple" DES (3DES) standard. It simply runs the data through the encrypting process three times with three different keys. This makes it much more resilient to attacks, but at the same time, it makes it three times slower since the result of the first encryption is processed twice more. It's worth mentioning that many other algorithms exist for encryption, including Rivest-Shamir-Adleman (RSA), Digital Signature Algorithm (DSA) and others.

A Public Key is one of a pair of separate, related keys utilized to encrypt data. This is known as *asymmetric key encryption* because one key is used for encryption and the other for decryption. In order for any application of this type to be effective, the private key must remain secret to secure both privacy and proof of origin for the data affected. The public key—known as asymmetric key—is used to decrypt the data by the recipient and to verify security.

One-way hashes are functions that when applied to data produce a fixed-length output value, known as the hash signature or simply hash or signature. These functions are called one-way because the results of the functions cannot be

[1] Techencyclopedia definition, www.techweb.com

reverse-engineered to reveal the original data that created the signature. Also, it is almost impossible to produce the same signature when two different data sources are processed. So, the hash is applied at the senders end once the data is delivered to the recipient, and its integrity is determined by comparing the result of the hash processed by the recipient to the one received from the sender. Hashes also have the added benefit of running quickly on general-purpose CPUs. The standard mechanism for hash functions is the Hashed Message Authentication Codes (HMAC), defined in RFC 2104.

Public Key Infrastructure

Public Key Infrastructure (PKI), as its name suggests, refers to the infrastructure needed to support public key encryption. A number of components are needed to make PKI work. I will now explain how each of them works and interrelates:
Certificate Authority (CA): CAs are the foundation component for PKIs. A CA uses its own private key to sign the private keys of trusted individuals, using a digital certificate format. The CA thus vouches for the authenticity of an entity's public key. Anyone who trusts the CA may by inference trust any party carrying a valid certificate with that CA's signature.

Internet Key Exchange:

IKE is a hybrid protocol that integrates the Internet Security Association and Key Management Protocol (ISAKMP) with a subset of the Oakley key exchange Scheme.
The whole purpose of IKE is to establish shared security parameters and authenticated keys—in other words, security associations (SAs)—between IPsec peers that define the processing done on a specific IP packet. IKE is actually a general-purpose security exchange protocol and may be used for policy negotiation and establishment of authenticated keying material for a variety of needs, for example SNMPv3, OSPFv2, etc. The specification of what IKE is being used for is done in a Domain of Interpretation (DOI). The DOI for IPsec is defined in RFC2407, which establishes how IKE negotiates IPsec SAs. In order to create IPsec SAs with a remote entity, you speak IKE to that entity. The protocol is a request-response type with an initiator and a responder. The initiator is the party that is instructed by IPsec to establish some SAs as a result of an outbound packet matching an SPD entry; it initiates the protocol to the responder.
IKE functions in two phases; phase one is used to establish a secure channel for doing IKE, called the IKE SA. This channel can be accomplished through main mode exchange or aggressive mode exchange.
In order for the secure channel to be established, the initiator "proposes" these six things:

1. The encryption algorithms to protect the data
2. The hash algorithms to deduce data for signing
3. An authentication method for signing the data
4. Information about a group over which to do a Diffie-Hellman exchange

5. A PRF for verification purposes (optional)
6. The type of protection to use (ESP of AH)

Main mode exchange is used to establish the first phase IKE SA, which is used to negotiate future communications. The main goal here is to agree on enough things to be able to communicate securely long enough to set up an SA for future communication.
Aggressive mode exchange provides the same services as main mode (it establishes the IKE SA). It accomplishes this in less negotiation exchanges than main mode, but unlike main mode, the quickness of the negotiation process does not provide identity protection for the communicating parties. The advantage of Aggressive mode, however, is speed.
In phase 2 general purpose SAs are negotiated, accomplished through quick mode exclusively. Once the communicating parties have established an IKE SA (phase 1), they can use quick mode. Quick mode is less complex than either main or aggressive mode since it's already inside a secure channel. Its basic function is to negotiate general IPsec security services and generating fresh keying material.

Diffie-Hellman is a mechanism for security key exchanges in an insecure environment. In its most basic form, it works like this: two people independently and randomly generate 2 values (private and public key pair). Each sends their public value to the other; each then combines the public key they received with their private key using the Diffie-Hellman combination algorithm. The resulting value is the same on both sides and can be used for fast symmetric encryption by both parties.

IKE is a very complex protocol, well beyond the scope of this paper. For more detail on IKE, please refer to RFC 2409.

Conclusion:

Writing about such complex technologies like IPsec is never easy, but the scope of this paper is to provide you with a minimum level of understanding and an incentive to learn more about this subject. Security on the Internet has become a priority for many organizations, and IPsec is a robust protocol that delivers on the requirements of today's business needs. In the advent of IPv.6, IPsec is a proven technology, flexible enough to work on today standards but also ready for tomorrow's challenges. All security specialists around the world should know the mechanisms by which IP security is implemented through the use of the IPsec protocol.

References:

S.Kent, R.Atkinson, "Security Architecture for the Internet Protocol", Network Working Group, Standards Track, RFC2401, November 1998
URL: http://www.ietf.org/rfc/rfc2401.txt

S.Kent, R.Atkinson, "IP Authentication Header", Network Working Group, Standards Track, RFC2402, November 1998
URL: http://www.ietf.org/rfc/rfc2402.txt

S.Kent, R.Atkinson, "IP Encapsulating Security Payload (ESP)", Network Working Group, Standards Track, RFC2406, November 1998
URL: http://www.ietf.org/rfc/rfc2406.txt

D.Piper, "The Internet IP Security Domain of Interpretation for ISAKMP", Network Working Group, Standards Track, RFC2407, November 1998
URL: http://www.ietf.org/rfc/rfc2407.txt

D.Maughan, M.Schertler, M.Schneider, J.Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", Network Working Group, Standards Track, RFC2408, November 1998
URL: http://www.ietf.org/rfc/rfc2408.txt

D.Harkins, D.Carrel, "The Internet Key Exchange (IKE)", Network Working Group, Standards Track, RFC2409, November 1998
URL: http://www.ietf.org/rfc/rfc2409.txt

A. Krywaniuk, ”Security Properties of the IPsec Protocol Suite”, IETF, IP security Working Group, Internet Draft, November 21, 2001
URL: http://www.ietf.org/internet-drafts/draft-ietf-ipsec-properties-01.txt

“ISO OSI 7 Layer Model forced with TCP/IP”
URL: http://mike.passwall.com/networking/netmodels/isoosi7layermodel.html - NET

Chris Weber,”Using IPSec in Windows 2000 and XP”, December 5, 2001
URL: http://online.securityfocus.com/infocus/1519

Chris Weber,”Using IPSec in Windows 2000 and XP: Part Two”, December 20, 2001
URL: http://online.securityfocus.com/infocus/1526

Webopedia
URL: http://www.pcwebopaedia.com/

TechEncyclopedia

URL: http://www.techweb.com/encyclopedia

Chris Weber, “Using IPSec in Windows 2000 and XP: Part Three”, January 2, 2001
URL: http://online.securityfocus.com/infocus/1528

Eddie Younker, “IP Security Protocol-based VPNs”, October 9, 2001
URL: http://rr.sans.org/protocols/IPsec.php

Microsoft Technet, “Step-by-Step Guide to Internet Protocol Security (IPSec)”
URL:http://www.microsoft.com/technet/treeview/default.asp?url=/TechNet/prodtechnol/windows2000serv/deploy/confeat/ispstep.asp

Doraswamy Naganand and Harkins Dan. “IPSEC, The New Security Standard for the Internet, Intranets, and Virtual Private Networks”. Upper Saddle River, NJ: Prentice Hall, 1999.