Tiago de Almeida

IP Security

Introduction

The version 4 of the Internet Protocol (IPv4), the still most widely adopted version, has no general-purpose mechanism to ensure security in the communication within a network. That is, the data transmitted between two communicating peers can possibly be intercepted, analyzed and even changed (Kozierok, 2005). The RFC 1636, titled "Security in the Internet Architecture", was issued by the Internet Architecture Board (IAB) in 1994 in order to identify specific needs for security mechanisms in the Internet architecture. Among these were the needs for authentication and encryption techniques. The version 6 of the Internet Protocol (IPv6) was designed including security features related to authentication and encryption. Such security capabilities were fortunately also designed to be used by IPv4 (Stallings, 2013).

The security capabilities for the IP level of a network are combined in a set of protocols called IP Security (IPSec). This set aims to provide security in three functional areas: (1) Authentication assures the packet's sender really is who the packet header informs, as well as assures the packet as a whole has not been modified since it left the source. (2) Confidentiality assures the communication between two peers is indeed confidential; therefore the packets will not be inspected by any other node in the network. Finally, (3) the Key Management capability assures security for the exchange of keys in the communication (Stallings, 2013).

It is known that several applications have their own security mechanisms regardless of IP level security resources. However, the goal of implementing security in the IP layer is to ensure secure networking not only for those applications that have their own security mechanisms but also for those that do not have any security feature regarding networking. By deploying IPSec an

organization can run secure private networks inside its own network, or even communicating securely with another network across the Internet by encrypting outbound packets and authenticating inbound packets (Stallings, 2013).

Applications and Benefits

IPSec can encrypt and/or authenticate all traffic at the IP level. This feature enables it to support several kinds of distributed applications securely. Remote logon, e-mail, Web access, file transfer and client/server applications, for example, will operate securely with no need for any modification. According to (Stallings, 2013) the following uses are examples for the deployment of IPSec:

- Secure communication among company's branches: IPSec can be used to deploy a Virtual Private Network (VPN) over the Internet. This way a company can ensure branches to communicate and access its network securely. There is no need for private networks since the company will be able to rely on existing Internet services.
- Secure remote access: A remote host upon using IPSec can connect securely to the company's network. It enables organizations and teams to work in a distributed and remote manner, saving costs of transportation.
- Secure connectivity with partners: Secure networks between different organizations can be deployed through IPSec, ensuring authentication, confidentiality and providing a key exchange mechanism.
- Secure electronic commerce security: Web and e-commerce applications usually have their own security mechanisms. However, the use of IPSec enhances the security by encrypting and authenticating the traffic, therefore adding an extra layer of security.

Implementing a security technology in the IP level brings several benefits, most of them related to the fact that upper level protocols e applications do not need to worry about securing its communication, neither they have to deal with the overhead of such operations. (Stallings, 2013) mentions the following examples of benefits from IPSec:

- IPSec can be deployed on a firewall or edge router. This strategy secures all the traffic that passes through the edge of the network. Traffic within the private network therefore doesn't have to deal with the overhead of security operations. Devices running IPSec encrypt and compress the outbound traffic, while decrypting and decompressing inbound traffic.
- IPSec runs below the transport layer, therefore below TCP and UDP. This makes it transparent to applications and higher layers of the OSI protocol stack. By implementing IPSec in the firewall or router there is no need to modify software on hosts and servers within the local network. Even when implementing IPSec directly in end systems, like a remote host, upper-layer applications are not affected.
- IPSec is transparent not just to devices within a local network and upper-layer software, but also to end users. By deployment IPSec solutions one avoids the need of training users on security mechanisms and managing their use of keys and security resources.

Documentation

IPSec is not a single protocol and therefore it is not defined by a single Internet standard. Instead, IPSec is a collection of techniques and protocols defined by several RFCs. The two main ones are mentioned below.

- RFC 6071 - “IP Security (IPSec) and Internet Key Exchange (IKE) Document Roadmap” - “Snapshot of IPSec- and IKE-related RFCs. It includes a brief description of each RFC, along with background information explaining the motivation and context of IPSec's outgrowths and extensions. It obsoletes RFC 2411, the previous "IP Security Document Roadmap." ([RFC6071])
- RFC 4301 - “Security Architecture for the Internet Protocol” - “Describes an updated version of the "Security Architecture for IP", which is designed to provide security services for traffic at the IP layer. It obsoletes RFC 2401.” ([RFC4301])

Several other RFCs define protocols, architectures and services related to authentication, cryptographic algorithms, and key exchange management, among others. All of them are presented and briefly described in the RFC 6071. Since the totality of IPSec specification is scattered across dozens of documents, it is in fact considered the most complex and difficult to grasp of all IETF specifications (Stallings, 2013).

Components and Protocols

In order to engage in a secure communication, two devices must agree in secure methods and protocols and establish a secure path between them. This path may cross several insecure networks and must still provide the desired security. According to (Kozierok, 2005), in order to set up such path, both devices must perform the following steps:

- Security protocols must be defined and both must use the same;
- Similarly, an encryption algorithm must be defined and used by both to encode data;
- Keys must be exchanged in order to encrypt and decrypt data;

IPSec provides the necessary tools to support these activities through a number of different components. The two main pieces, the ones that actually encode information and ensure security, are as follows:

- Authentication Header (AH): It is a protocol built to provide authentication services for IPSec. Besides allowing the recipient to verify whether the supposed originator of the message was indeed the one that sent it, AH also allows integrity verification. That is, the recipient can verify whether the data has been modified on the way (Kozierok, 2005). Its current specification is defined by RFC 4302.

  Authentication here is provided by computing a cryptographic hash-based message authentication code over most of the fields of the IP packet. The non-covered fields are those that may be modified in transit, such as TLL and the header checksum. The authentication code is then stored in a new AH header (Figure 1). This is usually placed between the original IP header and the payload, but more on that will be said in the following sections. Finally, the packet is sent to the other end (Friedl, 2005).

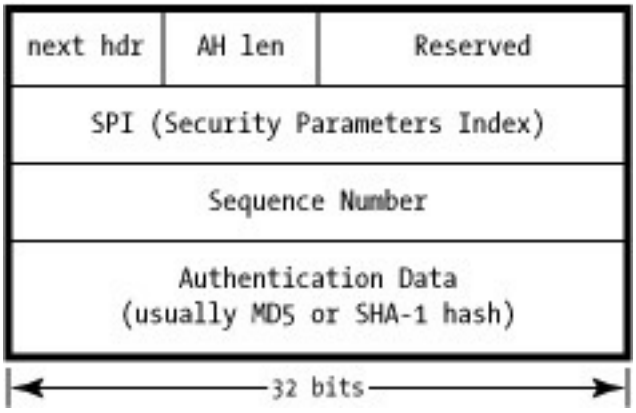

Figure 1: IPSec AH header (Friedl, 2005)

The last part of the header, called Authentication Data, carries an Integrity Check Value (ICV) that is usually built on top of some standard cryptographic hash algorithm such as MD5 or SHA-1. Using a straight checksum would provide no real security against intentional tampering because an attacker would be able to re-compute the hash. Instead, AH uses a Hashed Message Authentication Code (HMAC) that incorporates a secret value while creating the ICV and thus prevents the attacker of recreating the proper ICV (Friedl, 2005). The RFC 2104 describes HMAC. Figure 2 shows how the ICV is computed with the message and the contribution of a secret key.

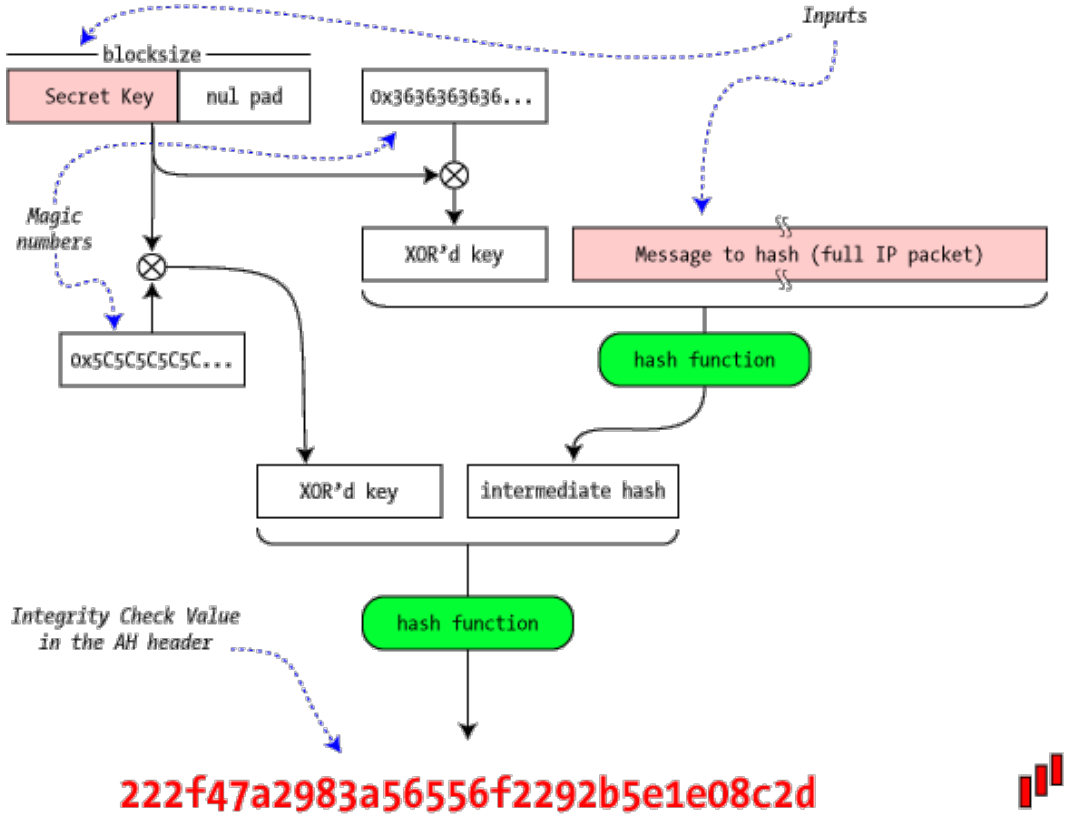

Figure 2: HMAC for AH Authentication (Friedl, 2005)

According to (Stallings, 2013) AH is actually currently considered deprecated because message authentication is also provided by ESP, presented below.

- Encapsulating Security Payload (ESP): The Authentication Header protocol ensures integrity of the data and provides an authentication mechanism. However, it doesn't provide a way to ensure the privacy of the information transmitted. If the data in the packet cannot be seen by anyone else besides its source and recipient, the Encapsulating Security Payload protocol needs to be used (Kozierok, 2005).

  ESP encapsulates the packets with header and trailer and provides encryption and optional authentication. Its current specification is defined in the RFC 4303. ESP not just adds a header like AH does. In order to provide encryption for the packet, ESP surrounds the encoded data with new header and trailer. The IPSec specification does not define any particular encryption algorithm, and the one used for a specific communication is defined by the communicating elements (Friedl, 2005). According to (Friedl, 2005) the most used algorithms are DES, 3DES, AES and Blowfish. Figure 3 shows the ESP encapsulated packet.

  As an optional feature, ESP also provides authentication through the same HMAC method used by AH. Contrasting AH, however, it authenticates the ESP header and encrypted payload only, that is, the authentication does not cover the whole IP packet. According to (Friedl, 2005) this strategy does not substantially weaken the security of the authentication, but it does provide some important benefits. Figure 4 shows the ESP encapsulated packet with authentication data.

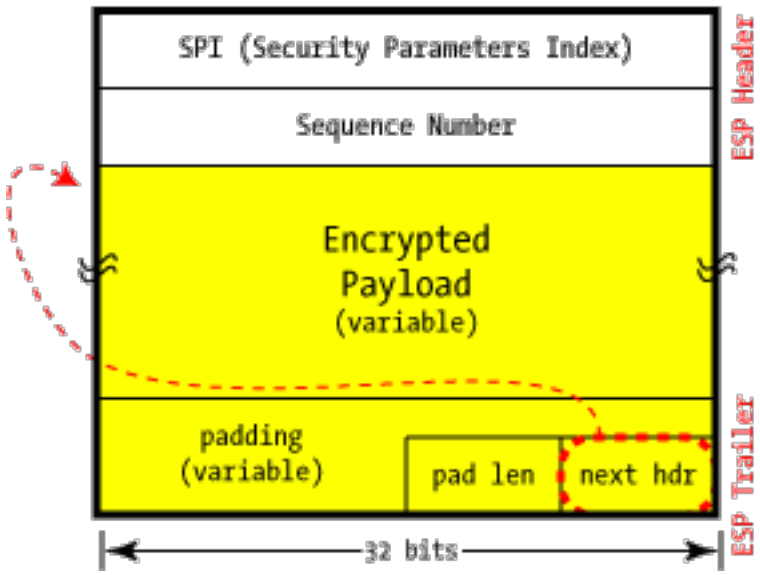

Figure 3: ESP encapsulation without authentication. (Friedl, 2005)

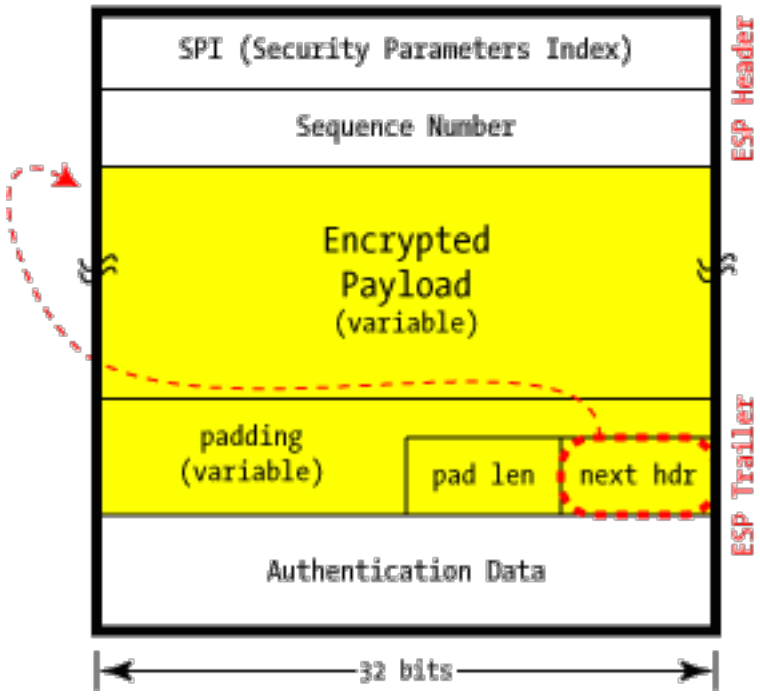

Figure 4: ESP encapsulation with authentication. (Friedl, 2005)

When an IP packet containing ESP data is intercepted the only usual data found is in the IP header, and it usually means the source and destination addresses. The IP header also reveals that the packet contains ESP data, but the actual payload type is also encrypted (Friedl, 2005). Besides that, it is impossible to determine what is inside the packet. It is even impossible to know whether or not the packet carries authentication data. This is actually determined by the Security Parameters Index, which references the pre-defined shared parameters and algorithms for this communication (Friedl, 2005). According to (Friedl, 2005), however, it's fairly easy to determine if a packet is carrying VoIP data inside ESP over the Internet. In this case, the QoS tags are in the outside header and reveal the IP precedence 3 (VoIP signaling) or IP precedence 5 (RTP traffic), for example.

Modes of Operation

The specification of the IPSec standard provides two modes of operation, called transport mode and tunnel mode. According to (Kozierok, 2005) these models are closely related to the function of the two core protocols presented above, the AH and ESP. Both of these protocols add a header to the datagram. This header contains security data and is used by the protocol to provide security to the communication. Choosing between the two modes of operation does not affect the header and its data. In fact, it defines which parts of the datagram are actually protected and how the headers are arranged to accomplish this.

- Transport Mode: In this mode the protocol transports the packet from the transport layer to the IP layer, adding its header in between. The packet is processed by AH/ESP and the

correspondent header is created. This header is placed in front of the transport (TCP/UDP) header. The IP header is then added in front of that (Kozierok, 2005).

From the IP layer's point of view the datagram received from the transport layer is the payload that will be encapsulated by the IP protocol. In the Transport Mode the IPSec is integrated with IP and the AH/ESP protocol acts only on this payload, placing its header between the payload and the IP header. Figure 5 illustrates this process (Kozierok, 2005).

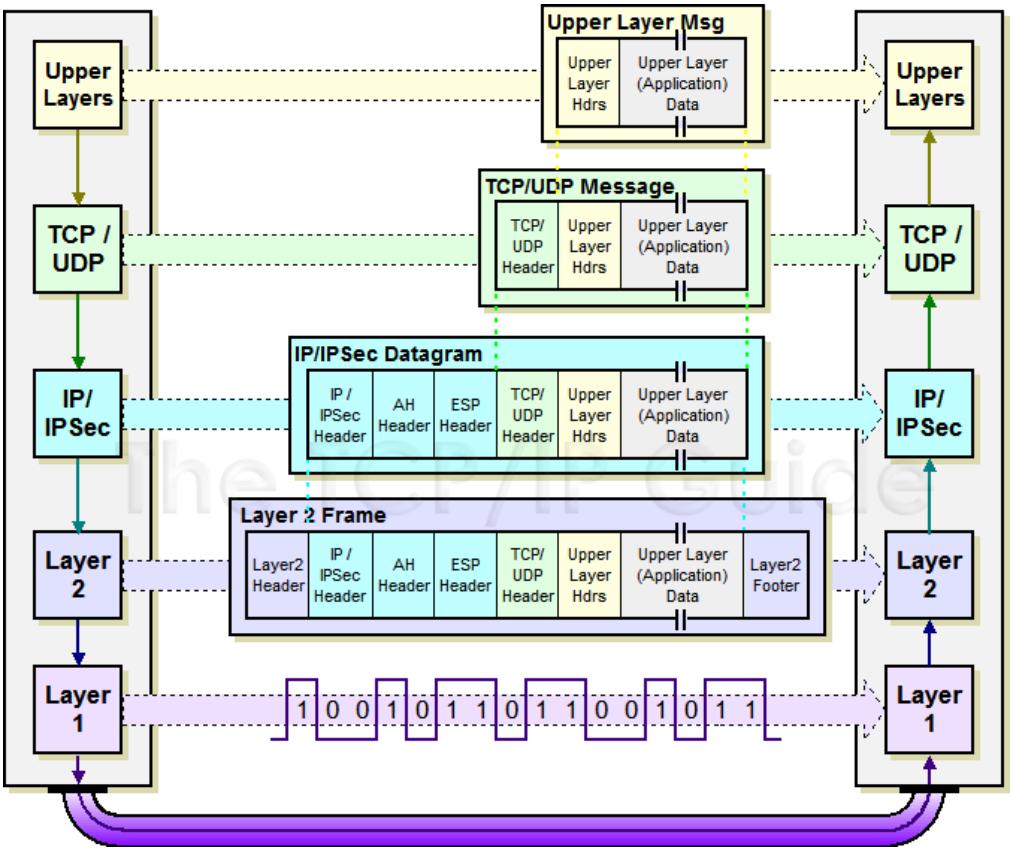

Figure 5: IPSec Transport Mode (Kozierok, 2005)

- Tunnel Mode: In this mode, IPSec acts on a complete encapsulated IP packet. That is, after the IP header has already been placed on the packet received from the transport

layer. Therefore, the IPSec is placed in front of the IP packet and, consequently, in front of the original IP header. When the encryption is used, the whole IP packet is encrypted. Then an IP header is added in front of the IPSec header. This process means the entire original IP packet is secured and then encapsulated within another IP packet. This encapsulation represents a virtual tunnel between IPSec-capable devices (Kozierok, 2005). The source and destination addresses of the new IP header can actually be different from those of the encompassing packet, protecting even more the original source and destination hosts (Friedl, 2005). Figure 6 illustrates this process.

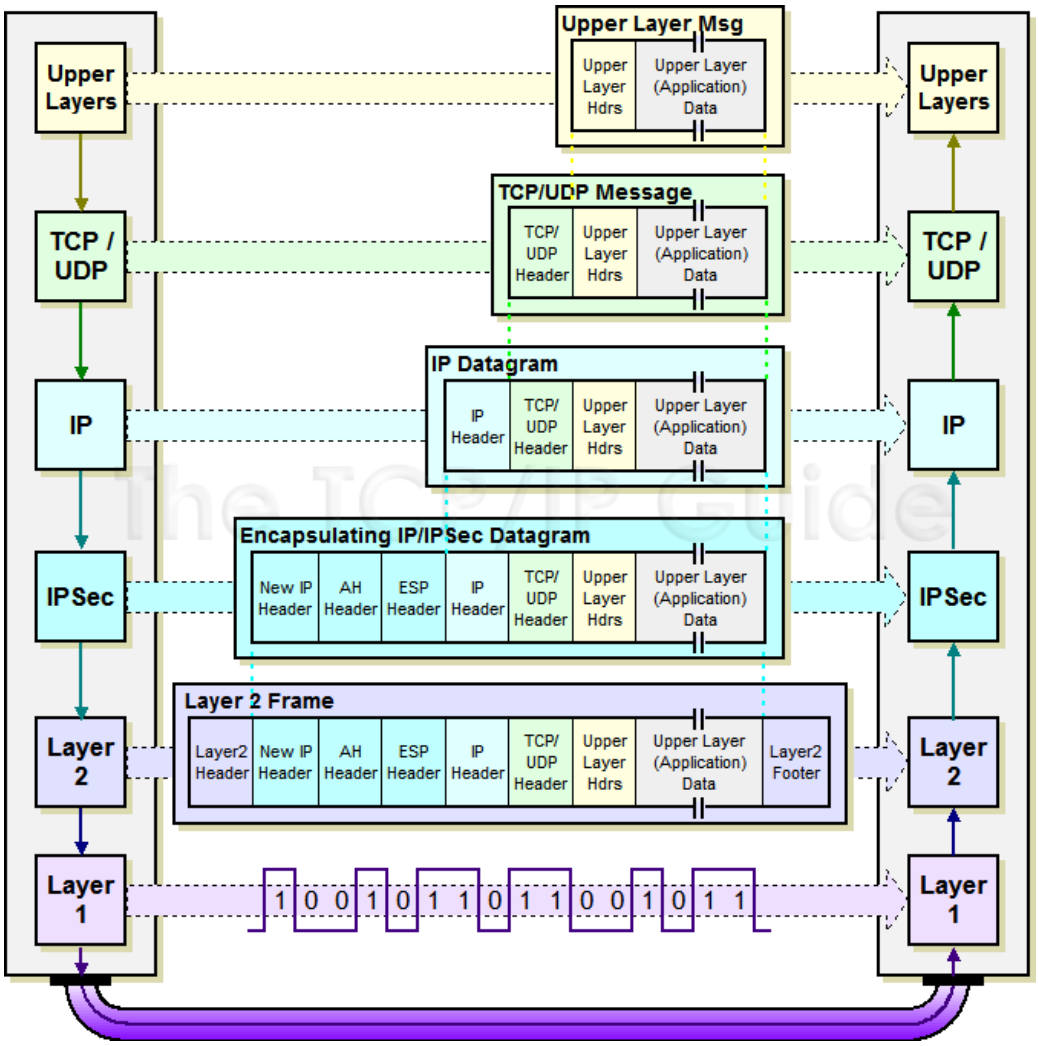

Figure 6: IP Tunnel Mode (Kozierok, 2005)

Hence we can say the tunnel mode protects the original IP packet as a whole, including its header, while the transport mode does not. Transport mode is commonly used for deploying end-to-end secure communication with hosts that run IPSec directly. Tunnel mode, on the other hand, is the usual choice when deploying VPNs (Kozierok, 2005).

Key Exchange

IPSec provides authentication and encryption in order to establish a secure communication channel. Such operations are based on the concept of a shared key. The shared key is a piece of information known only by the peers supposed to be securely communicating. This key is used to encode and decode the messages and thus make it readable only by those that posses the key. Without the key, one is prevented either from reading the message (if encrypted with ESP) or from tampering with it undetected (if AH is used) (Kozierok, 2005). Before IPSec can secure the messages, however, the communicating peers need to exchange the key that will be used in the communication session. The most straightforward way to share the key is via manual configuration. Someone generates the key and manually install them in the communicating devices. This process, however, is not scalable and might not be secure either (Friedl, 2005).

IPSec uses a protocol called Internet Key Exchange (IKE) to establish the secure exchange of keys. IKE is defined in RFC 5996 and according to (Kozierok, 2005) it is one of the most complicated of the IPSec protocols to comprehend. This paper does not intend to explore deeply the IKE protocol and therefore is going to cover just a brief outline of the protocol.

IKE combines the functions of a set of three other protocols and is therefore considered a “hybrid” protocol. The first in this set is the Internet Security Association and Key Management

Protocol (ISAKMP), which provides a framework for exchanging encryption keys and allowing negotiation of other security parameters through a series of phases (Kozierok, 2005).

The ISAKMP framework supports many different key exchange methods. In IKE, the framework is used as the basis for a specific key exchange strategy built with features of two key exchange protocols, called Oakley and SKEME (Kozierok, 2005). From ([RFC5996]) we have:

- "ISAKMP provides a framework for authentication and key exchange but does not define them. ISAKMP is designed to be key exchange independent; that is, it is designed to support many different key exchanges."
- "Oakley describes a series of key exchanges (called "modes") and details the services provided by each (e.g. perfect forward secrecy for keys, identity protection, and authentication)."
- "SKEME describes a versatile key exchange technique which provides anonymity, repudiability, and quick key refreshment."

IPSec X SSL - Strength and Weaknesses

IPSec's ability to tunnel any IP-based protocol and to keep the tunnel as a permanent connection is one of its key strengths. With this feature IPSec becomes an attractive tool to build secure VPNs. However, its application-agnostic design is also its weakness. Even though IPSec provides authentication, authorization and encryption when extending a corporate network through a VPN, it cannot differentiate traffic and restrict access to specific resources. The tunnel connects the remote host or network to the corporate network as if they were directly plugged. Consequently, remote users can typically access any corporate resource. This lack of granular control becomes a problem when the amount and variety of remote users and devices increases.

Different users and different devices usually are allowed access to varied parts of the corporate network and are required different levels of security. IPSec cannot differentiate among them. When the number of remote peers increases, the manageability of VPNs become harder to maintain. IPSec is not NAT-friendly, for example. When NAT is used, special configuration is required to ensure IPSec works flawlessly with the NAT setup (Saxena, 2012).

SSL, on the other hand, is easier to manage. It was designed for secure remote access and do not require special software to be installed. Instead, the browser is used to establish a secure session. SSL also provides tools for managing remote access. Among them is the ability to control access at a granular level with different profiles of authentication and authorization. SSL also has built-in logging and auditing capabilities as well as the ability to run host compliance checks to ensure remote hosts have the required security software and patches installed (Saxena, 2012).

Nevertheless SSL is not the magic answer to all issues IPSec is unable to deal with. IPSec is indeed the right solutions for many scenarios. If a remote host requires always-on connectivity to the corporate network, for example, SSL cannot provide it. IPSec also can support several legacy protocols due to its application-agnostic design. Both VPN methods have their importance and are currently deployed for different purposes.

Conclusion

IP Security is an open standard widely adopted and implemented. According to ("IPSec Vs. SSL", 2005), Check Point Software Technologies, Cisco Systems, Juniper Networks, Nortel Networks, Sonicwall and WatchGuard all offer IPsec VPNs with integrated firewalls.

IPSec provides secure communication for several network setups while allowing some control over how this security is implemented. The ability to choose between modes of operations, encryption algorithms and whether the packet is encrypted or just authenticated turns IPSec into a customizable solution. Besides that, it uses public domain encryption algorithms and protocols that are backed up by a community of developers and users.

The IPSec standard turns out to be a great solution for deploying security over a network. Its widespread adoption reveals it has successfully reached the goal of securing communication.

References

Frankel, S. and S. Krishnan, "IP Security (IPSec) and Internet Key Exchange (IKE) Document Roadmap", RFC 6071, February 2011, <http://www.rfc-editor.org/info/rfc6071>.

Friedl, S. (2005, August 24). An Illustrated Guide to IPSec. Retrieved September 12, 2014, from http://www.unixwiz.net/techtips/iguide-ipsec.html

IPSec Vs. SSL: Picking The Right VPN. (2005, September 1). Retrieved September 12, 2014, from http://www.networkcomputing.com/careers-and-certifications/ipsec-vs-ssl-picking-the-right-vpn/d/d-id/1213971

Kaufman, C., Hoffman, P., Nir, Y., and P. Eronen, "Internet Key Exchange Protocol Version 2 (IKEv2)", RFC 5996, September 2010, <http://www.rfc-editor.org/info/rfc5996>.

Kent, S. and K. Seo, "Security Architecture for the Internet Protocol", RFC 4301, December 2005, <http://www.rfc-editor.org/info/rfc4301>.

M. Kozierok, C. (2005, September 20). The TCP/IP Guide - IP Security (IPSec) Protocols. Retrieved September 12, 2014, from http://www.tcpipguide.com/free/t_IPSecurityIPSecProtocols.htm

Saxena, A. (2012, October 5). Quick overview of IPSEC and SSL VPN technologies. Retrieved September 12, 2014, from https://supportforums.cisco.com/document/113896/quick-overview-ipsec-and-ssl-vpn-technologies

Stallings, W. (2013). Chapter 20 IP Security. In *Cryptography and network security: Principles and practice* (6th ed., pp. 626-659). Prentice Hall.