Computer Network Security Protocols:  
A Review  

Anil Kumar  

Programmer-Cum-Networking Engineer, Haryana Roadways, Transport Department, Govt of Haryana, India.

ABSTRACT  
Network security is a complicated subject, historically only tackled by well-trained and experienced experts. However, as more and more people become ``wired'', an increasing number of people need to understand the basics of security in a networked world. This document was written with the basic computer user and information systems manager in mind, explaining the concepts needed to read through the hype in the marketplace and understand risks and how to deal with them.

Keywords: Protocol, Security, Secure Socket Layer (SSL) and Transport Layer Security (TLS) Protocols; secure IP (IPSec); Secure HTTP (S-HTTP), secure E-mail (PGP and S/MIME), DNDSEC, SSH.

1. INTRODUCTION  
The rapid growth of the Internet as both an individual and business communication channel has created a growing demand for security and privacy in this electronic communication channel. Security and privacy are essential if individual communication is to continue and e-commerce is to thrive in cyberspace. The call for and desire for security and privacy has led to several security protocols and standards. Among these are: Secure Socket Layer (SSL) and Transport Layer Security (TLS) Protocols; secure IP (IPSec); Secure HTTP (S-HTTP), secure E-mail (PGP and S/MIME), DNDSEC, SSH, and others. In this paper I discuss these protocols and standards within the framework of the network protocol stack as follow:

<table>
<thead>
<tr>
<th>Application Layer</th>
<th>Transport Layer</th>
<th>Network Layer</th>
<th>Data Link Layer:</th>
</tr>
</thead>
<tbody>
<tr>
<td>PGP, S/MIME</td>
<td>SSL, TLS</td>
<td>IPSec, VPN</td>
<td>PPP, RADIUS, TACACS+</td>
</tr>
<tr>
<td>S-HTTP</td>
<td>SET, KERBEROS</td>
<td></td>
<td></td>
</tr>
<tr>
<td>HTTPS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>S-HTTP</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2. SECURITY IN THE APPLICATION LAYER  
Pretty Good Privacy (PGP)  
The importance of sensitive communication cannot be underestimated. The best way, so far, to protect such information is to encrypt it. Encryption of e-mails and any other forms of communication is vital for the security, confidentiality, and privacy of everyone. This is where PGP comes in and this is why PGP is so popular today. Pretty Good Privacy (PGP), developed by Phil Zimmermann, is a public-key cryptosystem. PGP works by creating a circle of trust among its users. In the circle of trust, users, starting with two, form a key ring of public key/name pairs kept by each user. Joining this "trust club" means trusting and using the keys on somebody’s key ring. Unlike the standard PKI infrastructure, this circle of trust has a built-in weakness that can be penetrated by an intruder. However, since PGP can be used to sign messages, the presence of its digital signature is used to verify the authenticity of a document or file. This goes a long way in ensuring that an e-mail message or file just downloaded from the Internet is both secure and un-tampered with.

Secure/Multipurpose Internet Mail Extension (S/MIME)  
Secure/Multipurpose Internet Mail Extension extends the protocols of Multipurpose Internet Mail Extensions (MIME) by adding digital signatures and encryption to them. To understand S/MIME, let us first look at MIME. MIME is a
technical specification of communication protocols that describes the transfer of multimedia data including pictures, audio, and video. The MIME protocol messages are described in RFC 1521; a reader with further interest in MIME should consult RFC 1521. Because Web contents such as files consist of hyperlinks that are themselves linked onto other hyperlinks, any e-mail must describe this kind of inter-linkage. That is what a MIME server does whenever a client requests for a Web document. When the Web server sends the requested file to the client’s browser, it adds a MIME header to the document and transmits it. This means, Internet e-mail messages consist of two parts: the header and the body. Within the header, two types of information are included: MIME type and subtype. The MIME type describes the general file type of the transmitted content type such as image, text, audio, application, and others. The subtype carries the specific file type such as jpeg, orgif, tiff, and so on. S/MIME was then developed to add security services that have been missing. It adds two cryptographic elements: encryption and digital signatures. Encryption:- S/MIME supports three public key algorithms to encrypt sessions keys for transmission with the message: Diffie-Hellman, RSA, and triple DES. Digital signatures:- From a hash function of either 160-bit SHA-1 or MD5 to create message digests.

Secure-HTTP (S-HTTP)

Secure HTTP (S-HTTP) extends the Hypertext Transfer Protocol (HTTP). When HTTP was developed, it was developed for a Web that was simple, that did not have dynamic graphics, that did not require, at that time, hard encryption for end-to-end transactions that have since developed. As the Web became popular for businesses users realized that current HTTP protocols needed more cryptographic and graphic improvements if it were to remain the e-commerce backbone it had become. Each S-HTTP file is either encrypted, contains a digital certificate, or both. S-HTTP design provides for secure communications, primarily commercial transactions, between a HTTP client and a server. It does this through a wide variety of mechanisms to provide for confidentiality, authentication, and integrity while separating policy from mechanism. HTTP messages contain two parts: the header and the body of the message. The header contains instructions to the recipients (browser and server) on how to process the message’s body. During the transfer transaction, both the client browser and the server, use the information contained in the HTTP header to negotiate formats they will use to transfer the requested information. The S-HTTP protocol extends this negotiation between the client browser and the server to include the negotiation for security matters. Hence S-HTTP uses additional headers for message encryption, digital certificates and authentication in the HTTP format which contains additional instructions on how to decrypt the message body.

Hypertext Transfer Protocol over Secure Socket Layer (HTTPS)

HTTPS is the use of Secure Sockets Layer (SSL) as a sub-layer under the regular HTTP in the application layer. It is also referred to as Hypertext Transfer Protocol over Secure Socket Layer (HTTPS) or HTTP over SSL, in short. HTTPS is a Web protocol developed by Netscape, and it is built into its browser to encrypt and decrypt user page requests as well as the pages that are returned by the Web server. HTTPS uses port 443 instead of HTTP port 80 in its interactions with the lower layer, TCP/IP.

Secure Electronic Transactions (SET)

SET is a cryptographic protocol developed by a group of companies that included Visa, Microsoft, IBM, RSA, Netscape, MasterCard and others. It is a highly specialized system with complex specifications contained in three books with book one dealing with the business description, book two a programmer’s guide, and book three giving the formal protocol description. For each transaction, SET provides the following services: authentication, confidentiality, message integrity, and linkage. SET uses public key encryption and signed certificates to establish the identity of everyone involved in the transaction and to allow every correspondence between them to be private.

Kerberos

Kerberos is a network authentication protocol designed to allow users, clients and servers, authenticate themselves to each other. This mutual authentication is done using secret-key cryptography with parties proving to each other their identity across an insecure network connection. Communication between the client and the server can be secure after the client and server have used Kerberos to prove their identity. From this point on, subsequent communication between the two can be encrypted to assure privacy and data integrity. Kerberos client/server authentication requirements are:

- Security—that Kerberos is strong enough to stop potential eavesdroppers from finding it to be a weak link.
- Reliability—that Kerberos is highly reliable employing a distributed server architecture where one server is able to back up another. This means that Kerberos systems are fail safe, meaning graceful degradation, if it happens.
- Transparency—that users are not aware that authentication is taking place beyond providing passwords.
- Scalability—that Kerberos systems accept and support new clients and servers.
To meet these requirements, Kerberos designers proposed a third-party trusted authentication service to arbitrate between the client and server in their mutual authentication.

3. SECURITY IN THE TRANSPORT LAYER

These protocols are at the level below the application layer. We discuss two: Secure Socket Layer (SSL) and Transport Layer Security (TLS). Currently, however, these two are no longer considered as two separate protocols but one under the name SSL/TLS, after the SSL standardization was passed over to IETF, by the Netscape consortium, and Internet Engineering Task Force (IETF) renamed it TLS.

Secure Socket Layer (SSL)

SSL is a widely used general purpose cryptographic system used in the two major Internet browsers: Netscape and Explorer. It provides an encrypted end-to-end data path between a client and a server regardless of platform or OS. Secure and authenticated services are provided through data encryption, server authentication, message integrity, and client authentication for a TCP connection through HTTP, LDAP or POP3 application layers. It rivals S-HTTP. These two Web giants had a lot in common, however, there are some differences in design goals, implementation, and acceptance. First S-HTTP was designed to work with only web protocols. Because SSL is at a lower level in the network stack than S-HTTP, it can work in many other network protocols. Secondly, in terms of implementation, since SSL is again at a lower level than S-HTTP, it is implemented as a replacement for the sockets API to be used by applications requiring secure communications. On the other hand, S-HTTP has its data passed in named text fields in the HTTP header. Finally, in terms of distribution and acceptance, history has not been so good to S-HTTP (See SSL protocol stack).

The SSL Handshake

Before any TCP connection between a client and a server, both running under SSL, is established, there must be almost a process similar to a three-way handshake. This get-to-know-you process, is similarly called the SSL handshake. During the handshake, the client and server perform the following tasks:

- Establish a cipher suite to use between them.
- Provide mandatory server authentication through the server sending its certificate to the client to verify that the server’s certificate was signed by a trusted CA.
- Provide optional client authentication, if required, through the client sending its own certificate to the server to verify that the client's certificate was signed by a trusted CA.

Exchange key information using public key cryptography, after mutual authentication, that leads to the client generating a session key (usually a random number) which, with the negotiated cipher, is used in all subsequent encryption or decryption. The customer encrypts the session key using the public key of the merchant server (from the merchant’s certificate). The server recovers the session key by decrypting it using its private key. This symmetric key which now both parties have is used in all subsequent communication.

Transport Layer Security (TLS)

TLS is the result of the 1996 Internet Engineering Task Force (IETF) attempt at standardization of a secure method to communicate over the Web. The 1999 outcome of that attempt was released as RFC 2246 spelling out a new protocol—the Transport Layer Security or TLS. TLS was charged with providing security and data integrity at the transport layer between two applications. TLS version 1.0 was an evolved SSL 3.0. Frequently, the new standard is referred to as SSL/TLS. Since then, however, the following additional features have been added:

- Interoperability - ability to exchange TLS parameters by either party, with no need for one party to know the other’s TLS implementation details.
- Expandability – to plan for future expansions and accommodation of new protocols.

4. SECURITY IN THE NETWORK LAYER

These protocols also address Internet communication security. These protocols include: IPSec and VPN technologies.

Internet Protocol Security (IPSec)
IPSec is a suite of authentication and encryption protocols developed by the Internet Engineering Task Force (IETF) and designed to address the inherent lack of security for IP-based networks. IPSec, has a very complex set of protocols described in a number of RFCs including RFC 2401 and 2411. Although it was designed to run in the new version of the Internet Protocol, IP Version 6 (IPv6), it has also successfully run in the older IPv4 as well. IPSec sets out to offer protection by providing the following services at the network layer:

- Access Control –to prevent an unauthorized access to the resource.
- Connectionless Integrity – to give an assurance that the traffic received has not been modified in any way.
- Confidentiality – to ensure that Internet traffic is not examined by non-authorized parties. This requires all IP datagrams to have their data field, TCP, UDP, ICMP or any other datagram data field segment, encrypted.
- Authentication – particularly source authentication so that when a destination host receives an IP datagram, with a particular IP source address, it is possible to be sure that the IP datagram was indeed generated by the host with the source IP address. This prevents spoofed IP addresses.
- Replay protection – to guarantee that each packet exchanged between two parties is different.

IPSec protocol achieves these objectives by dividing the protocol suite into two main protocols: Authentication Header (AH) protocol and the Encapsulating Security Payload (ESP) protocol. The AH protocol provides source authentication and data integrity but no confidentiality. The ESP protocol provides authentication, data integrity, and confidentiality. Any datagram from a source must be secured with either AH or ESP (See diagrams of these). IPSec operates in two modes: transport and tunnel:

- **Transport mode**
  
The Transport mode provides host-to-host protection to higher layer protocols in the communication between two hosts in both IPv4 and IPv6. In IPv4, this area is the area beyond the IP address. In IPv6, the new extensions to IPv4, the protection includes the upper protocols, the IP address and any IPv6 header extensions (see extensions).

- **Tunnel mode**
  
  Tunnel mode offers protection to the entire IP datagram both in AH and ESP between two IPSec gateways. This is possible because of the added new IP header in both IPv4 and IPv6. Between the two gateways, the datagram is secure and the original IP address is also secure. However, beyond the gateways, the datagram may not be secure. Such protection is created when the first IPSec gateway encapsulates the datagram including its IP address into a new shield datagram with a new IP address of the receiving IPSec gateway. At the receiving gateway, the new datagram is unwrapped and brought back to the original datagram.

**Virtual Private Networks (VPN)**

A VPN is a private data network that makes use of the public telecommunication infrastructure, such as the Internet, by adding security procedures over the unsecured communication channels. The security procedures that involve encryption are achieved through the use of a tunneling protocol. There are two types of VPNs: remote access which lets single users connect to the protected company network and site-to-site which supports connections between two protected company networks. In either mode, VPN technology gives companies the facilities of expensive private leased lines at much lower cost by using the shared public infrastructure like the Internet. The two components of a VPN are:

i. Two terminators which are either software or hardware. These perform encryption, decryption and authentication services. They also encapsulate the information.
ii. A tunnel – connecting the end-points. The tunnel is a secure communication link between the end-points and networks such as the Internet. In fact this tunnel is virtually created by the end-points.

VPN technology must do the following activities:

- IP encapsulation – this involves enclosing TCP/IP data packets within another packet with an IP-address of either a firewall or a server that acts as a VPN end-point. This encapsulation of host IP-address helps in hiding the host.
- Encryption – is done on the data part of the packet. Just like in SSL, the encryption can be done either in transport mode which encrypts its data at the time of generation, or tunnel mode which encrypts and decrypts data during transmission encrypting both data and header.
- Authentication – involves creating an encryption domain which includes authenticating computers and data packets by use for public encryption.

**Types of VPNs**

The security of VPN technologies falls into three types: trusted VPNs; secure VPNs; and hybrid VPNs.
Trusted VPNs: In these VPNs a customer trusted the VPN provider to safeguard his or her privacy and security by maintaining the integrity of the circuits. This security is based on trust.

Secure VPNs: Trusted VPN actually offers only virtual security, so security concerns in VPN are still there. To address these concerns, protocols that encrypt traffic at the edge of one network or at the originating computer, moved over the Internet like any other data, and then decrypt when it reaches the corporate network or a receiving computer are used. This way it looks like encrypted traffic has traveled through a tunnel between the two networks. Between the source and the destination points, although the data is in the clear and an attacker can see the traffic, still one cannot read it, and one cannot change the traffic without the changes being seen by the receiving party and, therefore, rejected. Networks that are constructed using encryption are called secure VPNs. Secure VPNs are more secure than trusted VPNs.

Hybrid VPNs: Hybrid VPN is the newest type of VPN technologies that substitutes the Internet for the telephone system as the underlying structure for communications. The trusted VPN components of the new VPN still do not offer security but they give customers a way to easily create network segments for wide area networks (WANs). On the other hand, the secure VPN components can be controlled from a single place and often come with guaranteed quality-of-service (QoS) from the provider.

5. SECURITY IN THE LINK LAYER AND OVER LANS

In the Data Link Layer, there are several protocols including : PPP, RADIUS and TACAS+.

Point-to-Point Protocol (PPP)

This is an old protocol because early Internet users used to dial into the Internet using a modem and PPP. It is a protocol limited to a single data link. Each call went directly to the remote access server (RAS) whose job was to authenticate the calls as they came in. A PPP communication begins with a handshake which involves a negotiation between the client and the RAS to settle the transmission and security issues before the transfer of data could begin. This negotiation is done using the Link Control Protocol (LCP). Since PPP does not require authentication, the negotiation may result in an agreement to authenticate or not to authenticate.

Remote Authentication Dial-In User Service (RADIUS)

RADIUS is a server for remote user authentication and accounting. It is one of a class of Internet dial-in security protocols that include Password Authentication Protocol (PAP) and Challenge-Handshake Authentication Protocol (CHAP). It is mainly used by Internet Service Providers (ISPs) to provide authentication and accounting for remote users. It can be used also in private networks to centralize authentication and accounting services on the network for all dial-in connections for service. It has two main components: authentication and accounting protocols.

Terminal Access Controller Access Control System (TACACS+)

This protocol, commonly referred to as “tac-plus”, is a commonly used method of authentication protocol. It is a strong protocol for dial-up and it offers:

- Authentication – arbitrary length and content authentication exchange which allows many authentication mechanisms to be used with it.
- Authorization
- Auditing – a recording of what a user has been doing and in TACACS+, it serves two purposes: To account for services used and to audit for security services.

SUMMARY

This paper reviews the different security protocols used at different levels to protect the network at one place. This will be helpful to all who are interested in network security protocol. In this paper I try to describe HOW, WHEN and WHERE these protocols are used. These protocols are necessary to understand the security mechanism of computer network and improve it.

REFERENCES


[12]. Information on the development of IPSEC standard can be found in the IPSEC home page, http://www.ietf.org/html.charters/ipsec-charter.html and the IPSEC mailing list ipsec@tis.com.


