Secure Data Exchange System : Minimizing Security Attack Risks while Preserving Bandwidth

Abdel-karim Al Tamimi **Khalid AlHokail**

aa7@cec.wustl.edu khalid@al-hokail.com

Abstract

In the world where Internet applications dominate data communications, a system to exchange information securely between Internet users is vital. Many solutions have been put forward to facilitate such exchange, these solutions have their own advantages and disadvantages. In this paper we introduce a thin secure layer that resides on top of the IP layer and supports encryption and compression of IP packets. Our solution provides the necessary security level to overcome most of the security risks without sacrificing performance and network bandwidth. By giving the option to choose between different levels of encryption and compression levels, the users can choose the level most suitable to their needs. In order to show the usage of our security layer, we implemented a simple chatting system that's capable of exchanging encrypted text messages and allows the clients to send encrypted and compressed files. The application also provides two ways to sniff network traffic showing the risks of exchanging information without imposing a proper security level.

Keywords:

Security Protocol, IPSec, Encryption, Rijndael, Compression, Bandwidth Consumption, Session Management, Cryptography, IP Layer, TCP/IP Suite.

See Also: IP Security : A Brief Survey Security in Wireless Data Networks : A Survey Paper

---

Table of Contents:

Back to Table of Contents

1. Introduction

The Internet has replaced many traditional communication systems because of its advantages in both its cost and usability. Using the Internet to share information on daily basis puts users in risk to be endangered by many Internet security attacks. Most of the data and money exchange is done these days using one of the many services provided to the users online. Such convenience comes with a high price where these communications are not always efficiently secure.

With the vast introduction of the wireless world, the exchanged information now is more prone to security risk than ever. One of the several security attacks is data sniffing, where the transmitted data is exposed to a third party and all the exchanged data is compromised. There are even commercial products that help network administrators or others to view, store and analyze exchanged data packets[EffeTech06].

The other common security attack is DoS (Denial of Service) attack, where the attacker overwhelms the victim's host with many resource requests. A more severe version of this attack is DDoS (Distributed DoS), where the attacker uses more than one host to attack the victim's host.

Many solutions provide the mechanism of encrypting the ongoing data exchange packets between two peers. Even when the packets are encrypted, the users are still prone to another security attack: Replay attack. Where the attacker uses pre-validated packets and sends them to one of the users to confuse and disrupt the communication.

In this project we introduce a simple to implement and easy to use infrastructure that can provide the necessary security level to exchange information between two nodes without the fear of being exposed to the sniffing attack. We also provide the necessary application level support to prevent replay attack. Because of the system's intended simplicity, it does not cover all the security risks out there. It does however, provide a base to overcome these hazards in the future.

This project report is divided into 6 sections: sections 1 and 2 cover the relevant and current solutions such as IPSec (IP Security) and SSL (Secure Socket Layer). Section 3 describes the system infrastructure, objectives, and design choices. Section 4 provides an overview of the software structure of the system. Section 5, provides a walk-through for the basic chat system introduced and the current status of the project. Finally, section 6 provides a summary and conclusion of the project.

Back to Table of Contents

2. Related Solutions

This section will illustrate two of the common solutions to facilitate exchanging data between users: IPSec and SSL. While IPSec operates on top of the IP layer, SSL operates on top of the TCP layer.

2.1 IPSec

IPSec is short for Internet Protocol Security and was developed by the IETF (Internet Engineering Task Force) to enable secure exchange of packets using the IP layer (layer 3). It is widely used in secure VPN (Virtual Private Network) communication. IPSec can work on two different encryption modes[WikiIPSec07]

1. Transport Mode

This mode only encrypts the payload (data) portion of the packet leaving the header unencrypted. This mode is mostly used in host-host communications.

2. Tunnel Mode

This mode the entire packet is encrypted and/or authenticated including the header. Which implies that another header has to be added to allow routing to work. This mode is used mainly in router-router communications.

IPSec provides two methods of securing the IP packet using one of the two protocols:

1. Authentication Header (AH)

This protocol provides integrity and data origin authentication. It can also protect against replay attacks ,repeating or delaying a previously valid packet, by using the sliding window technique. AH protects the IP header except for the mutable fields that have to change during transmission from source to destination such as the TTL field.

2. Encapsulating Security Payload (ESP)

This protocol ensures confidentiality, data origin authentication, connectionless integrity and anti-replay service. Unlike AH, ESP doesn't protect the IP header in any way but this can be protected by using the Tunnel Mode to protect the inner IP packet but the packet header will remain unprotected.

In order for IPSec to operate properly, both the sender and receiver will have to exchange public keys. Internet Key Exchange (IKE) protocols are used to help exchange public keys between the two nodes.

2.2 SSL

SSL is short for Secure Sockets Layer which is a cryptographic protocol that provides secure communication over the internet using popular applications such as web browsers, emails and instant messaging. It was developed by Netscape Communications Corporation in 1994 and in 1999, IETF established RFC 2246 that documented Transport Layer Security (TLS) that is based on SSL. Unlike IPSec (which is implemented at the kernel level), SSL is implemented at the user level and uses TCP for reliable communication so that SSL will not have to worry about delivering the packets. It is placed above the TCP/IP layer and below the high-level application protocols. SSL provides authentication for both the client and the server. There are two methods of authentication; the first is that only the server is authenticated to ensure its identity leaving the client unauthenticated. The other method is called mutual authentication where (in addition to authenticating the server) the client is also authenticated using either his certificate or a username and a password[WikiTLS07].

SSL supports the use of various types of encryption and hashing algorithms. This is decided when the client wants to communicate with the server by sending a Client Hello message to the server with all algorithms that the client supports (along with other information such as the session ID, a random number…etc) and the server will decide which algorithms to use by selecting the strongest algorithms that both can support and then notifies the client of the choices.

Many protocols are based on SSL but the most popular protocol is HTTPS. Many websites are based on HTTPS especially the ones that accepts confidential information such as credit cards or medical records. Another popular SSL-based protocol is FTPS which is a secure FTP protocol.

In this section we described the main features of both SSL and IPSec protocols. In the next section we will discuss the system infrastructure and design choices.

Back to Table of Contents
