Jong-Hyeon Henry Lee <jhlee@acm.org>

Project: XecureWeb/XecureNet - Channel Encryption
Employer: SoftForum Inc., Korea
Term: April 1996-February 1999

1) Application of theory

a) Analysis of requirements

This project was designed to implement an alternative to the Secure Socket Layer (SSL). Owing to the US export control on cryptographic software in 1996, web browsers could not support encryption stronger than 40-bit, and the 40-bit encryption was not secure enough to deal with important data exchange, such as financial transactions.

Financial institutions recognised that the Internet is a good medium to provide financial services to their customers because of its universal availability, but there is no assurance of its security and, in fact, there is a potential danger of attacks from the unknown. At the time, banking machines were using DES algorithm, a symmetric encryption algorithm with a 56-bit key, and the banking industry recognised that it is not so secure any more. Obviously, the financial industry demanded stronger encryption than the 40-bit encryption on the web.

b) Design and construction

We designed a protocol that provides an encrypted data channel between a web browser and a web server with a customisable key length so that customers can control the strength of security. This is a client-server protocol and the encryption key is transferred from the server through a secure exchange and the key resides on a digital certificate.

Widely used web browsers like Microsoft's Internet Explorer or Netscape, recognise certificates issued by some certificate issuers whose root certificates are embedded in these browsers. Otherwise, web browsers require some more steps to recognise certificates, which are issued by other certificate servers. We needed to provide client software running on the client's PC with a mechanism to deal with certificates issued by third-party issuers as well as with channel encryption software.

c) Quality control

Basically, we tested the software with both black box and white box tests. The software consists of a server part and a client part. Especially, the client part needed to support all available operating system versions on client PCs via two major browsers, Netscape and Microsoft Internet Explorer. We tested it with the various combinations of underlying operating systems and browsers.

The server part should be running on various UNIX machines including Solaris, HP-UX, and IBM AIX, and we tested it in those server environments. We needed to control the versions for each operating system. Since we implemented it in C/C++, there were many system-dependent parts in the server. It took us a long time to port the product on other platforms.

2) Practical experience

This product has been widely used in the Korean financial sector and the clients include majority of online stock brokerage service companies and major domestic and international banks. Since each company has its own configuration and computing environment, a large amount of customisation work was involved in the initial period, and we then were able to make the product more flexible by incorporating most of customisation points inside the product as configurable options.

3) Communication skills

The most important communication skill in this project was to understand users' workflow and their computing environment. We studied various configurations and prepared for possible customisation requests from users.

4) Social implication of engineering

This solution enhances the communication security and reduces the possibility of malicious attacks. Since it can reduce the danger of network hacking and packet sniffing, it is now widespread in various types of transactions over the Internet including banking, stock trading, and shopping. The digital certificate legislation empowered this type of software and boosted its use.