Chapter 5
Security Considerations

For security's sake, SSL proxy server and inspection engine are separately deployed apart from web server. Consequently, it equivalently takes one-more hop overhead. For avoid the penalty, in some situation another solution is provided to inspect the encrypted web connections. It is supposed to develop the cryptographic functionalities binding with web server. In practical, it is accomplishable for some deployment demands. Nevertheless, the drawbacks are stated as below. First, modify kernel code is difficult and complex, including that intercepts the packets before HTTP stack seeing it. Second, the framework that is built in kernel is incompatible with other operating systems. A related work named as CodeSeeker [38] examines HTTP requests on malicious traffic. It is installed on the Web server and intercepts HTTP traffic off the stack immediately after it is decrypted by SSL if it's HTTPS. Unfortunately, CodeSeeker only sits in a passive mode simply alerting attacks rather than blocking illegitimate traffic. Third, the methodology in this thesis that SSL proxy server apart from web server has another advantage for providing more application-layer services such as caching, content adaptation or language translation. In contrast, that is a major limitation for obvious flexibility problem if the function of SSL proxy server built on web server.

The SSL proxy server was provided with strong sense of confidentiality, message integrity. Therefore, strict protection and management of SSL proxy server are fundamental requirements. SSL proxy server plays an important role in today's network security environment. Hence, a network administrator has to take care for keeping the contents and operations secure and especially taking detection and

reaction to any inappropriate, incorrect or anomalous activity.

In the following paragraph we are going to discuss an exceptional security issue. The SSL proxy server sits intervening between client and server, the relationship is similar to the man-in-the-middle attack [39]. The attack is a manner used by hackers to compromise the SSL-encrypted connections. The major idea is that the client connects not directly to the server, but to a computer inserted by the hacker in the middle of their network connection. Hence the nodes relationship of communication scheme looks like this: client-hacker-server. It is completely transparent to both the client and the server, but the hacker is able to steal the glance at all decrypted traffic. The novel manner was manipulated as Phishing [40].

As mentioned before, the SSL proxy server belongs to a component of intranet defense, sits co-located with origin web server and operates in coordination with application firewall. Inevitability, network administrator has to take control strictly against compromised security. Next, SSL proxy server and web server must accomplish authentication, identify and acknowledge to each other [41]. Following the granted, SSL proxy server should be trusted with origin web server. Therefore, SSL proxy server provides the legitimate certificate to outside clients in place of web server.

Due to the asymmetric nature of the mechanism used by SSL, the hacker won't be able to imitate the original web server even if he gets the public key. To verify the connection the private key is also necessary, which is stored only on the server and never be transmitted out.

Another consideration is based on how a client can establish trust in the computation and data storage at servers of otherwise unknown credibility. This oversight forces clients to trust the good intentions and competence of the server operator—but gives clients no basis for that trust. In the dominant SSL framework of

server-side authentication, the user trusts a CA to bind a server identity to a public key. The server has a private-public key pair, and a certificate from some CA attesting to something vague about the entity owning this public key. The client browser has some notion of which CA root keys it recognizes as valid identity. When a client opens an SSL connection, it verifies that the certificate from the server has correctly signed by a CA root, the client's browser currently recognizes as legitimate. The client and server then carry out a key generation/exchange protocol that ensures that the client, and a party which knows the private key matching the server's public key, share a symmetric key—that is (theoretically) shared by no one else, not even an adversary observing the messages between the client and server.

Based on the "defense-in-depth" principle, besides the effort of server side, and most important of all, client users also have to pay attention to check the certificate before transferring the confidential data to or from a security web site. With the aid of the certificate, the client verifies the server's authenticity. If the certificate is valid, the next step can be taken, otherwise the user needs to be consulted. Most users never notice the warning message in the act of connecting to a new secure web server. There is always possibilities that we are connected to some strange box who wants to intercept our sessions. To place emphasis on defense in depth, therefore, responsibility should be taken for every user.