Chapter 6
Conclusions

The threats include any kind of malicious codes, such as viruses, worms, Trojan horses and injection attacks. They are coming from both outside (hackers, competitors), and inside (mistakes made by employees, worms brought in on laptops) the walls of the organization. They could bring about malicious accesses by hackers and the stealing of sensitive information by competitors. The Internet is a major source of security attacks, especially with the number of Web attacks doubling year over year. Because of the wide variety of threats and the complexity of today's IT networks, implement a security solution is a major challenge and require extensive expertise in a variety of highly specialized security disciplines.

Consider the network security for encrypted application layer. In this thesis a SSL proxy server has been proposed and implemented. The objective of this framework is to establish an SSL inspection engine that can provide inspection application firewall plain-text traffic with deep inspection. The most meaningful intension once the inspection engine is possessed of ability to scan the content stream so that it can give in-depth protection against threats posed by harmful content entering and ensures no sensitive materials out. The SSL proxy server has been proposed in conjunction with inspection engine. Therefore, the inspection engine is now able to inspect and analyze every inbound and outbound packet in an unencrypted form. In addition, the system supported proxy cache due to restore the packet contents even it comes from encryption traffic.

Compared to similar approaches such as SSL VPN, our implementation has a little difference in the design motivation. SSL VPN is an alone device and is designed

to protect intranet resources, while our framework is based on existing equipment such as application firewall. Our framework has no requirement of alteration for almost network settings.

The potential features of SSL proxy server also established as ensure the certain level of security between end-to-end communications. It makes scalability and compatibility with existing secure infrastructures. Not only hold the principle of simple is good, but also to be a modular architecture in its service. It simplifies the implementation, reliability, and scalability of value-added services, but provides flexibility with customizing the security level by administrator and offloads computing from origin web server. In additional, leveraging available communication methods and building under a common, open protocol for edge-based appliance communication to handle these value-added services. By adding on pieces of components on the application firewall we can make sure that out network traffic is highly secure.

This thesis presents the design, implementation, and experimental evaluation of an SSL proxy server. The server intercepts the initial connection from the client and establishes another connection with the Web server. The functionalities of proxy caching and inspection engine combination are also integrated. Therefore, the system can mitigate the server's loadings and inspect content as a consequence of the integrity combination of decryption, proxy cache, inspection and encryption. Moreover with the potential feature of SSL resume handshake, which speeds up establishing connection between proxy and web server. It has a positive impact overall performance by reducing response times and bandwidth requirements. In addition, a lightweight cipher suite, called NULL-encryption, is suggested that offloads the symmetric cryptographic computing from the origin web server and saving traffic bandwidth.

Although the experiments are limited by equipment resources, resume handshake technique were evaluated in our experiments. The evaluation result shows that we can have at most 164 % improvement. A conclusion is given that the files with smaller size can be coped with quickly in given testing duration. Thus, there are more connections for small size files as well as the number of handshakes. Consequently, the file with a smaller size obtains better improvement than larger one. We also evaluated and presented the improvement of using NULL-encryption technique in the selectivity content framework. At most approximately 13.8% are obtained in the experiments. Overall performance also had a great impact for larger files due to saving symmetric cryptographic computing in exchanging data.

These observations are useful for designing servers used in the e-commerce environment and also for supporting SSL transactions more efficiently. However, HTTPS needs more CPU intensive than ordinary HTTP communications. In particular, there has been a considerable amount of work on the enhancement of system performance through the addition of cryptographic hardware. More specifically, easing off SSL overhead would be the next work of this framework. For example, two new mechanisms were proposed by [42] for caching handshake information on TLS clients and reducing both network traffic and the number of round trips.

SSL has become the universal standard for authenticating the identity of a critical host, and for encrypting communications between client and server. More systems are interconnected together through Internet and also require more precious security level. For example, Wireless Transport Layer Security (WTLS) [43] is based on the TLS v1.0 security layer, provides privacy, data integrity and authentication for WAP services. Our future work also focuses on developing a more scalable framework which can support all kinds of secure protocol applications, not only web application but also like SSH, even IM. Because of SSL protocol provides secure channel for

end-to-end transaction. All these applications exploited by SSL protocol would be able to be integrated and become components of our framework. By implementing the integrated SSL proxy server in next work, the objective of system is provided for full range of secure protocol applications to prevent all attacks behind secure channel, while significantly improving the overall security of the IT systems.