Chapter 1
Introduction

Computer network is fundamental to the operations of most communities. A computer network that is functioning properly and efficiently is an invaluable asset. However, the need to secure one's network is also very fundamental and crucial. Security underside was fraught with increasing communications and productivities associated with access information, email, streaming media, instant message, etc. While several years ago most attacks exploited network-level vulnerabilities such as flaws in the TCP/IP protocol, today's hackers primarily exploit application-level bugs. The Protocol-layer firewalls and the Network-based Intrusion Detection Systems (NIDSes), provide protections and recovery capabilities for all priority information assets. Intrusion Detection and Prevention Systems (IPSes) are based on the typical detection technology used in NIDS, but actually sit in-line to the network. The premise behind an IPS is once an intrusion is recognized the IPS system filters malicious traffic and drops the connection.

NIDS / IPS have become indispensable to the network security over past several years. A major reason is, recognizing this reality, knowledgeable hackers have advanced well and have devised sophisticated attacks that are designed to circumvent. Except network-based threats [1], content-based threats are now becoming the norm with worms, viruses, Trojans, and backdoor threats. Slammer, Blaster and Sasser were a few examples of sophisticated worms and email viruses that made headlines in the last few years and have shown us how fast these types of threats can spread.

In terms of the "Defense In Depth [2]," that means having multiple layers of security. "Defense In Depth" technology has been evolving so as to tackle these newer

threats. The typical equipments include application firewalls [3], stateful inspection firewalls [4], content filtering engines [5], anti-virus inspectors, anti-spam gateways, and even SSL VPN, such as Figure 1.

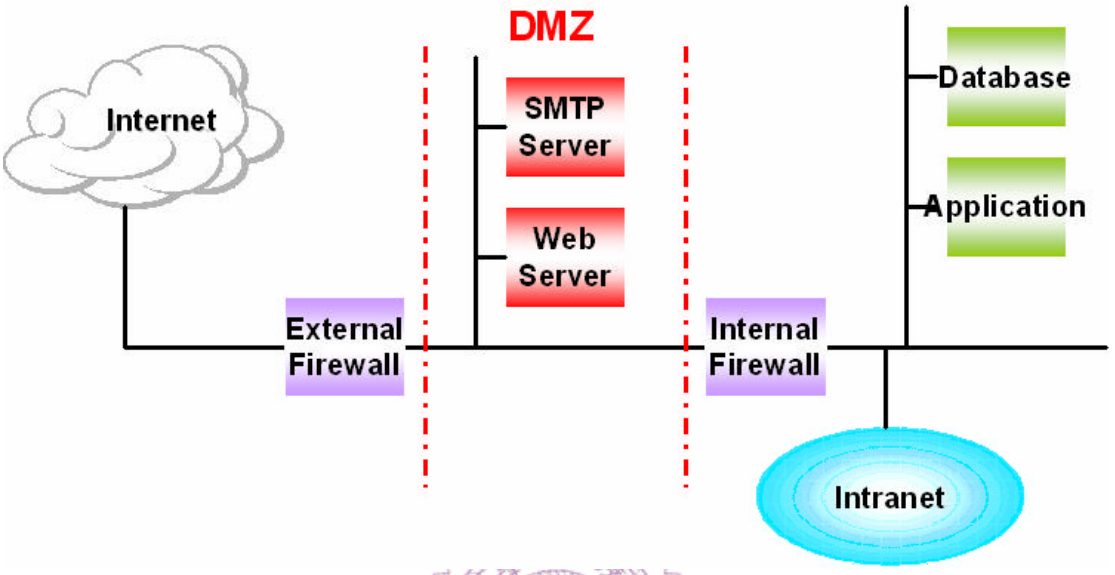

Figure 1. The topology of "Defense In Depth"

These platforms have been developed to detect the signature or monitor behavior from application layer [6]. The packet inspection engine receives incoming requests and attempts to match the header portions of packets (along with other possible feature sets) with known traffic signatures. If the traffic signatures match an "allowed" rule the packets are allowed to pass through the firewall. If the traffic signatures match "deny" rules, or they don't match "allowed" rules, they should be rejected or dropped. The capabilities of inspection engines can be further broken into two categories: stateful and non-stateful. A stateful packet inspection firewall learns a session characteristic when the initial session is built after it passes the rule base, and requires no return rule. The outbound and inbound rules must be programmed into a non-stateful packet inspection firewall.

Among all application services, World Wide Web (WWW) has become a popular and convenient application for all kinds of users. It offers global access to nearly any type of information. Its fundamental protocol, HTTP (TCP port 80) and HTTPS (TCP port 443) are particular interests to hackers, which are commonly open in many networks [7]. It has comes a security underside fraught with increasing communications and productivities associated with access information, email, streaming media, instant message, etc. However, data security plays an essential role in today's web service. Detecting and defending against attacks at the application layer is more difficult than at lower layers. Web applications pose unique security challenges to businesses and security professionals in that they expose the integrity of their data to the public.

In the most stringently secured systems, separate tiers differentiate between content presentation, security and control of the user session, and the downstream data storage services and protection. They avoid intercepted, tampered or forged by the man in the middle. The recent explosive growth of the Internet and the WWW has brought with it a need to securely protect sensitive communications sent over this open network. The SSL 3.0 protocol [8] [9] has become a de facto standard for cryptographic protection of web http/https traffic. The SSL protocol provides data stream exchanges to be confidentially, integrity and authentication. Hence, the web transactions could be safely processing without worrying about exposing clear text format in the communication. Consequently, many banks, shopping sites and businesses corporations have already installed a web server built in with SSL protocol. Secure HTTP (HTTPS) is one of the popular protocols to transfer sensitive data over the Internet.

However, the security would be not able to guarantee from above discussion even if both deep inspection and cryptographic protection are employed together. In

the area of web security, despite having strong encryption on the browser-server channel, users still have no assurance about what happens at the other end or at the middle of the transaction. For instance, suppose someone issues an attack command or publishes a virus program hidden as the result of encryption. The threat will be invisible by application firewalls. If the hacker appears to be a “normal user”, he can pass all the regular security checks and end up engaged at the application layer. Though the intention of the secure cryptographic protocol achieves message protect. The tricks include virus, inappropriate document or spam, could be evaded behind encryption as well as malicious code of any type can bypass firewalls without inspection by using the encrypted feature of SSL protocol. The encrypted application content layer should be inspected while building a solid security firewall.

There are various ways for a malicious party to steal this kind of information from the users’ personal computers, ranging from Trojan horses to JavaScript bug exploits. These malicious methods could be evaded under secure channel. When it comes to the crunch most used SSL protocol to perform encryption. The application firewall is shown as should be upgraded to provide with ability to cope with encryption traffic so as to ensure application layer security.

In this thesis, a framework of SSL Proxy Server is proposed. The objectives of SSL proxy server are to compute cryptographic functionalities, decrypt packets, and extract the origin contents of HTTPS traffic. It provides layer-7 application firewall with plain contents to inspect and identity the signatures. After performing in-depth intrusion detection analysis, the SSL proxy server, then forwards secure traffic to the origin web server. Besides, the proxy server also provides several application-layer services such as caching technique due to cipher-text traffic has been decrypted already. The potential features are made include security, scalability, flexibility and feasibility. The system can be incorporated with sophisticated content-based

applications such as IDS/IPS or proxy cache. In addition, since SSL computing is expensive, the performance is improved by having take advantage of the SSL characteristic of resume handshake. Moreover, the flexibility SSL cipher suites are provided benefits that help transferring data in diverse security level.

The thesis is organized as follows. Chapter 2 introduces the background on the types of application layer attacks and an introduction of SSL protocol. Next, Chapter 3 explains our design and implementation of the framework of SSL proxy server. Chapter 4 presents our results of evaluation. Chapter 5 discusses some security issues in details. Finally some conclusions are given in Chapter 6.