Design and Implementation of Secure 3-Tier Web Application with Open Source Software*

Kim, Chang Su** · Low, Hooi Yin*** · Lee, Yong Ju****

〈Abstract〉

Providing a secure 3-tier Web application has become a high priority for companies as e-businesses have increased the amount and the sensitivity of corporate information that can be accessed through the web. Web applications become more difficult to secure with this very increase in online traffic and transactions. This paper first reviews the 3-tier of web application, types of attacks that can threaten web application services and security principles. We then are designing and implementing a secure web application with open source software that able to mitigate the web application vulnerable to attack.

Key Words : Security, 3-tier web application, Open source software, Design and implementation

Ⅰ. Introduction

Traditional applications were fully installed on a local computer or accessed with a locally installed client that interacted with remote server software. But with the enormous success of the World Wide Web (WWW), many applications are developed for the web. This has lead to the development of web applications in different area such as banking, finance, e-commerce, and education. In the web application development, 3-tier architecture is the typical one[1]. In 3-tier architecture, the application process is being separated into three specific tiers. At the presentation tier, the user can see displayed information via a web browser and content served from a web application server. The occurred business logic processing will be performed at the middle tier. There is a data tier at the back end which is responsible to handle the database processing and access to the data.

Web applications are able to handle many kinds of transactions and provide important functions to people and business such as online bookstores and banking systems which moving millions of dollars. These applications rely on complex interactions with database servers, web application servers, and host, and it has generated wholly new security challenges such as information security and data integrity (encryption), and uncertainty about the true identity of

* This work was supported by the Kyungpook National University Research Grant, 2008

** 영남대학교 경영학부 교수
*** 영남대학교 대학원 경영학과
**** 경북대학교 컴퓨터정보학부 교수(교신저자)

communicating parties (user authentication)[2].

In order to provide a secure web environment to users while using a web application, we need to make sure that there are a reasonable assurance web application that users' personal information will be kept secure from unauthorized attempts to access it during transaction. Implementing security technologies into the web application are critical to provide a secure web application service.

Cost-effectiveness is an important issue in developing a secure web application. It is very costly in developing a holistic secure web application and its need a huge amount of maintenance fees to maintenance a secure web application. According to a survey released by security firm Mcfee, small businesses have, for the most part, frozen spending on security in 2009, despite an increase in perceived threats[3]. Due to this issue, open source software is the best solution in helping small business company to develop a simple, intelligent, and secure web application with minimum cost. Besides, open source software also provides us a simple license management, public collaboration and abundant online document support. The main goal of this paper is to implement a secure web application with open source software which is cost-effectiveness and able to mitigate web application vulnerable to attack.

Our web application architecture is designed with depth security where it has multiple layered of protections and defense through diversification. We use SSL to protect the process of exchanging sensitive data between clients browse and web application server using HTTPS connection. VPN tunnel is developed to protect the data transfer between web application server and database server to prevent attacks such as eavesdropping attack and denial of service. We used network firewall and web application firewall to provide a protective layer between the resources of a private network and users from different network. Password encryption is developed to protect the sensitive data store in database server.

Ⅱ. Review of 3-tier Secure Applications

2.1. 3-tier of Web Applications

3-tier architecture[4] is a client-server architecture in which the user interface (presentation tier), business logic (middle tier) and data storage or access (data tier) are developed and maintained as independent modules or on separate platforms. Thus, the module for a tier can be changed and relocated without affecting other tiers. Therefore, with 3-tier architecture, an enterprise or software packager can easily evolve an application when new needs and opportunity arise.

The presentation tier or user interface is what the user sees when they open a web page in the browser [5]. This web browser (client) interacts with web application server through HTML over HTTP. In web application the languages used in this tier are typically HTML, DHTML, CSS and javascript. Presentation layer is independent with the other tiers in such a way that it can be changed without doing any changes to the business or the database layer. The presentation tier does all its work through calls to the middle tier.

The middle tier contains a web application server. It is where the business logic is located. The main task of middle layer is processing data, business validation and business workflow[5]. The middle tier will validate

the data input conditions before calling a method from data tier and to ensure the outputs is correct. This validation of input is called business rules. These rules also apply to the calculations or other action that takes place in the middle tier[6]. Business workflow is sequence steps that involve state transformation. The middle layer would be implemented as reusable components and server pages. This tier is where the developers define the classes, functions, sub procedures, and properties[7]. Middle tier is responsible for processing the data retrieved from data tier and sent the data to the presentation layer

The data tier contains a database server. It includes data and data store software; relational databases, email stores, message queues, and directory service[5]. The data tier is a separate component and its main responsibility is to serve up the data from the database and return it to the caller. The data in data tier can be logically reused. The data tier hides the database engine that's working behind the screens. This tier also kept data neutral and independent from web application servers or business logic. Only the data tier will get influences if there is change to the database. Developers define queries and stored procedures in this layer[7].

2.2 Types of Attacks

In a web application, a flow of data is started when a user makes a request at the web interface. As data flows through 3-tiers of the web application, each tier will becoming a potential attack point for an attacker if the request is not properly handled. In this section we will provide a brief review of some types of attacks that can threaten a web application.

(1) *Denial of Service (DoS) and Distributed Denial of Service (DDoS)*: DoS is an attack technique with the intent of preventing a web site from serving normal user activity. The user or organization will be deprived of the services of a resource to which they would normally expect to have access when DoS occurred. This loss of service might affect a particular network service such as e-mail, access to web sites, online account (banking) or it might involve the temporary loss of all network connectivity and services[8-9].

DoS at the application layer may target on each independent component at 3-tier web applications which are a web application user, a web application server, and a database server. DoS attack on a specific user of the web application occurs when an intruder repeatedly attempts to login to a web site as user with invalid password. This process will eventually lock out the legitimate user. DoS attack on the web application server occurs when an intruder uses buffer overflow techniques to send crafted request which will crash the web application process and cause the system to be inaccessible to the normal user activity. DoS attack on the database server occurs when an intruder uses SQL (Structured Query Language) Injection techniques to modify the database and make the system to become unusable.

In a DDoS attack, an attacker scans systems to attack. By taking advantage of security vulnerabilities or weaknesses of a system, the attacker will install handler system software on the client system to scan for, infect or compromise other agent systems[8-9]. The agent systems are loaded with remote control attack software and respond to the client when it issues commands to handler systems that control agents in a mass attack. The attack is "distributed" because the

attacker is using multiple computers or systems to launch the DoS attack. DDoS attack has overwhelmed several high-powered Internet commerce sites such as Amazon, Yahoo! and eBay early in 2000.

(2) *Man in the Middle (MITM)*: In MITM attack, the attacker situated between the web application user and the real web application server, and proxies all communications between the systems[10]. Therefore, with this vantage point, all transactions are observed and recorded by the attacker. The attacker directs the web application user to their proxy server instead of the real web application server. The attack host then acts as the real web application server by establishing a connection with the user on one side and another connection with the real web application server, relaying the traffic back and forth. The attacker relays the traffic between the user and the web application server, making them believe that they are having direct private connection to each other.

(3) *Viruses, Trojans, and Worms*: Viruses are classified according to their mode of infection, the path used to replicate the virus, and the type of system infected [9]. Virus can replicate itself by requires some form of carrier or 'host'. A virus can damage the business operation or system it 'infects' either accidentally or deliberately. Virus can occur in PC's software as boot blocks, in file allocation tables, in. EXE and. COM files, or as functional files. These programs seek out unused resources and use them to resolve master program problems or tasks. Trojan programs are designed to hide themselves inside apparently harmless applications until triggered[9]. Worms propagate and exist independently. Worms do not have to attach themselves to another program or part of the operating system[9]. Morris worm spread by finding IP addresses on the machine. Slammer worm sent UDP packets to cause buffer overflow.

(4) *Cross-Site Scripting (XSS)*: XSS is an attack technique that forces a web site to echo attacker-supplied executable code, which loads in a user's browser[11]. The code itself is usually written in HTML/JavaScript or any other browser-supported technology.

XSS attack occurs when an attacker allows to capture private session information by introducing malicious codes to a dynamic page. An attacker could capture the session information, peer into private user details such as ID, passwords, credit card information, home address, telephone number, and social security/tax ID using the malicious codes. There needs a way that attacker can make the victim's browser execute the script that he wants it to execute no matter what the malicious codes is.

There are two types of XSS attacks, non-persistent and persistent. Non-persistent attacks require a user to visit a specially crafted link laced with malicious code. Persistent attacks occur when the malicious code is submitted to a web site where it's stored for a period of time.

(5) *SQL Injection*: SQL is a textual language used to interact with relational databases. SQL injection occurs when un-trusted values are used to construct SQL commands, resulting in the execution of arbitrary SQL commands given by an attacker[11]. SQL injection is a security vulnerability that occurs in the database layer of an application. The vulnerability is present when user input is incorrectly filtered for string literal escape characters embedded in SQL statements or user input is not strongly typed and thereby unexpectedly executed. The impact of this attack can allow attackers

to gain total control of the database or even execute commands on the system.

(6) *Buffer Overflow:* Buffer Overflow is a common software flaw that results in an error condition which occurs when data written to memory exceed the allocated size of the buffer. As the buffer is overflowed, adjacent memory addresses are overwritten and this causes the software to fault or crash[11]. When memory is corrupted, a buffer overflow can be used as DoS attack. Buffer Overflow vulnerabilities have been used to overwrite stack pointers and redirect the program to execute malicious instructions and also to change program variables.

Buffer Overflow vulnerabilities most commonly occur in programming languages such as C and C++. In the case of C and C++ programming languages, when input values are larger than expected, the buffer overflow can crash the application and the operating system[12].

(7) *Summary:* In this section, we have a review on some types of attacks that can threaten a web application. In order to provide further insight of the type of attacks occur at each tier, here we attempt to organize the above review into <Table 1>.

<Table 1> also listed three attack points in the web application where an attacker can abuse the applications and resources. As show in <Table 1>, XSS will only attack at user interface tier. Attacker can abuse the applications and resources at web application server tier and database server tier by type of attack such as DoS, MITM, SQL injection, and buffer overflow. Viruses, Trojans and Worms can damage all three tiers such as database server, web application server, and user interface. Therefore we can see that at each tier there is a strong potential that an attacker can either directly or indirectly cause a site outage by taking down the server or the application. The situation will become worse if there is a skilled attacker who used method that can not normally detected to access data and resources outside of the application scope, which means they would neither be stopped nor logged.

2.3 Security Principles

"What is security?" Security is fundamentally about

<Table 1> Types of Attacks and Attack Points

Type of Attack	Description	Consequence	Attack Points		
			UI	WA	DS
Denial of Service	Invalid data input	Crash server and application		YES	YES
Man in the Middle	Relay message and connection	Access sensitive data		YES	YES
Viruses, Trojans and Worms	Malicious code	Damage web application	YES	YES	YES
Cross-Site Scripting	Use URL meta code to insert Trojan code	Server-side exploitation, access sensitive data	YES		
SQL Injection	Use SQL statement to modify database	Crash application and data, access sensitive data		YES	YES
Buffer Overflow	Overflow field input	Crash site and application, access sensitive data		YES	YES

Ref. UI: User Interface (Client), WA: Web Application Server, DS: Database Server

protecting assets[13]. Assets may be tangible items such as web page or customer database, or less tangible items such as company's reputation. The security of a system is an assessment of the extent that the system protects itself from external attacks that may be accidental or deliberate. This means that security is not just important because of attacks which are planned but also of accidentals. Therefore security plays an important role in establish and maintain the authenticity (correct attribution) of data we create, send, and receive across the web application system. In e-business fields, security is important in ensuring a company's reputation that the company's reputation will not be tarnished by a security breach leaking their customer's information. Thus, e-businesses need to be on guard against information theft, espionage, and liability. In this section we will provide a brief review of the security principles.

(1) *Authentication*: Authentication is ensuring that the identity of a subject or resource is the one claimed [9]. This process ensures the claimed identity of an individual by confirming their identity to an appropriate level of confidence, after that hand the individual to an appropriate task. In the web application, authentication means the process of uniquely identifying the clients of the applications and services. These might be end users, other services, processes, or computers[13]. Authentication is an important step in building secure web applications, where it will define who can run the application, and then ensure that users are who they claim to be.

(2) *Authorization*: Authorization is the process of checking the authentication of an individual or resource to establish and confirm their authorized use of, or access to information or other assets[14]. In the web application, authorization means that only authorized users should be able to gain access to the web applications, data, and services, no matter where they are located. Each web page should check if the user accessing the web page has the appropriate permissions or not.

(3) *Availability*: Availability means that systems remain available for legitimate users [13]. A system's authorized users have timely and uninterrupted access to the information in the system and to the network. Information that is not available when and as required is not information at all but irrelevant data. There are several important things can do to ensure information availability. First of all, is to keep complete and current backups which can be useful in recover information from an intrusion or any other disaster. Second is to have no single point of failure means when equipment fails (firewall, server, or network link), a backup system needs to be ready to take over.

(4) *Confidentiality and Privacy*: Confidentiality and privacy are concerned with preventing the unauthorized disclosure of sensitive information [15]. The disclosure could be intentional, such as breaking a cipher and reading the information, or it could be unintentional, due to carelessness or incompetence of individuals handling the information. Information will often be applicable only to a limited number of individuals because of its nature, its content or because its wider distribution will results in undesired effects including legal or financial penalties, or embarrassment to one party or another.

(5) *Integrity*: Integrity is the guarantee that data is protected from accidental or deliberate (malicious) modification [13]. Useful information must be complete and accurate. The information must not be

manipulated in any way, either through electronic errors or human intention. Integrity must contain accurate, complete, and consistent data.

(6) *Accountability*: Accountability means the responsibility for actions and processes [14]. When any action is carried out on an information system, an individual needs to be accountable or responsible for that action. The person who has the accountability may delegate the actual work to someone else but they would still retain the accountability.

(7) *Auditing*: Auditing is the formal checking of the records of a system to ensure that the activities that were anticipated have taken place and have actually happened [14]. There are two main purposes of auditing. First is to provide the ability to determine whether, how, and when a system was attacked and what was attacked. Second is to provide the ability to troubleshoot security issues (Howard, 2002).

(8) *Non-repudiation*: Non-repudiation is the concept of ensuring that the validity of a statement or contract cannot be repudiates or refutes by a party in dispute [16]. Non-repudiation guarantees that the message sender is the same as the creator of the message. Digital signature is used to guarantee that a message has been sent by one specific entity digital certificate. In the world of e-business, the goal of non-repudiation is to prevent repudiation of valid transactions, which is critical to the success of e-business.

(9) *Summary*: In this section, we have a brief review about common security principles. Each of these principles might be more important than the others upon the application and context. For example, an organization would encrypt an electronically transmitted important document to prevent an unauthorized person from reading its contents. Thus, authorization, authentication and confidentiality of the information are paramount. If an individual succeeds in breaking the encryption cipher and then, retransmits a modified encrypted version, the integrity of the message and non-repudiation is compromised. On the other hand, an e-commerce company such as Amazon. com would be severely damaged if its network was out of commission for an extended period of time. Thus, availability and auditing is a key concern of such e-commerce companies. When an action is carried out on an information system, an individual needs to be accountable for that action. Lastly, security matter in e-business organization is important in ensuring a system is secure when the system needs to block all attempts to access corporate resources.

Ⅲ. Requirement Analysis

3.1 Overview

The objective of this stage is to understand clearly the data and design, and create a best solution by clarify all of the research requirements. All the requirements that are collected will be re-examined in this stage and they will be incorporated into the design stage. In this paper, we will suggest two kind of approach which is useful in gathering requirements.

The first approach is getting ideas from existing web applications. The vast number of currently running web applications could provide developers the inspiration for gathering requirements. This approach may also provide developers with the clues for sustainable requirements. This is because the deployed web sites have evolved overtime been massively tested

with real users. Therefore, developers might consult similar existing web sites to learn the essential content that should be included in the application. If the web site developer consulted is stable, they could identify relatively complete requirements and the requirements that have been widely accepted. These refers may give many idea of doing the better thing of the web site development. Then it may lead to the better function or performance of the web application.

Another approach is model-based approach. Its goal is to construct application models that show the structure, processes, and resources of a business as simply and directly as possible. This approach allows developers to gather requirements by applying the conventional information gathering techniques to those who are directly involved in the business processes and practices. The first approach can be augmented by the second approach.

3.2 Analysis

Analysis is the process of examining the requirements and developing the blueprint of an application to be built, which manifests higher global and local coherence.

(1) *Security Policies and Standards*: A security policy enables the technology to protect the information, data, and transactions over the internet, and safeguard the trust relationship. This also defines restrictions to determine what applications and users are not allowed to do[17]. It is critically important that a web application developed is done by following established policies and standards and is compliant with audit requirements. For example, if a web application authentication standard lists the need to have multi-factor authentication, then the web application build should be compliant to that standard.

(2) *Network Infrastructure Components*: The web application's developers have to identify and understand the network structure provided by the target environment and the baseline security requirements of the network[17]. These security requirements include filtering rules, port restrictions, and supported protocols. Besides, the developers also have to identify the role of firewalls and firewall policies which are likely to affect the application's design. The possible communication ports and authentication options from the web application server to remote application and database servers can be affect if the internet-facing application is separated from internal network by firewalls, or there may be firewalls in front of the database server. It is important to consider the kind of protocols, ports, and services that are allowed to access internal resources from the web application servers in perimeter network. The assumptions that made about network and application layer security and the component to be handled should be record to prevent security controls from being missed. The implications of a change in network must be considered carefully. It is important in identifying the lost of security in the application if specific network change is implemented.

(3) *Web Application Topology and Environment*: Application's deployment topology and the remote application tier (if occurred) are a key consideration that must be incorporated in order to develop a secure web application in the web environment[17]. If there is remote application tier in the application topology, it is important to consider the effective way to secure the network between servers to address the network

eavesdropping threat and to provide privacy and integrity for sensitive data. It is important to consider identity flow and identify the accounts that will be used for network authentication when an application connects to remote servers. A common approach is to use a least privileged process account and create a duplicate account on the remote server with the same password.

The web application is situated within a complex environment which can add vulnerability threats to the web application. In order to deploying the application securely within this web environment, it is important to separate server running production of the application from the rest of the internal servers (typically in a DMZ). This server should not run any other software that might disrupt the web application and should never be used to develop the application code. Before every new release the server running production version should maintain a sterile environment cleaned. The application can be copied into the sterile production environment by using an internal computer. This will ensure that only the minimal needed parts of the application actually reside in the production environment.

In order to prevent an abuse of the management application, the production site should never be administrated from outside the organization. It is best not to use remote administration even from within the organization. Administration should optimally be executed locally on the production computer.

(4) *Web Application Vulnerability and Practice*: The following is a set of common application vulnerability categories which are helpful in designing and building a secure web application and it is also helpful in evaluating the security of an existing application.

Input Validation: Input validation refers to the way an application filters, scrubs, or rejects input before additional processing[17]. It is a challenging issue because there has no single answer for constitutes valid input across or within application and there is no single definition of malicious input. For example, it is difficult to determine from where an application consume data, because an application can consume data within or across application. However, proper input validation is one of the strongest measures of defense against application attacks. Proper input validation is consider as an effective countermeasure that help in prevent input attacks such as XSS, SQL injection, and buffer overflows. Input validation of a web application can be improved by practices such as assume all input is malicious, centralize approach, do not rely on client-side validation, be aware with canonical issues, and lastly constrain, reject, and sanitize input.

Authentication: Authentication is the process of verifying caller identity[17]. There are three aspects to consider. First aspect is to identify that in application where should authentication be required. Generally authentication is required whenever a trust boundary is crossed. Trust boundaries usually include assemblies, processes, and hosts. Second aspect is to validate caller's identity with usernames and passwords. Third aspect is to identify the user on subsequent requests and an authentication token is required. This verifies that the information will not fall into wrong hands. Authentication of a web application can be improved by practices such as separate public and restricted areas, use account lockout policies for end-user accounts, support password expiration periods, able to disable accounts, do not store passwords in user stores, required strong passwords, do not send

passwords over the wire in plain text, and protect authentication cookies.

Authorization: Authorization is the process that an application uses to control access to resources and operation[17]. Security issues such as information disclosure and data tampering happened due to improper authorization[17]. Therefore, defense in depth always be consideredas an important security principle for application's authorization strategy. Authorization of a web application can be improved by practices such as use multiple gatekeepers, restrict user access to system-level resources, and consider authorization granularity.

Configuration Management: Web application's configuration management refers to the way an application handles operational issues such as application type, an application's connected database, and administration of an application[17]. Web application configuration management functionality should be considered carefully. Most applications require interfaces that allow content developers, operators, and administrators to configure the application and manage items such as web page content, user accounts, user profile information, and database connection strings. An administration interface can be severe due to consequences of a security breach. This is because the attacker has ends up running with administrator privileges and has direct access to the entire site. Configuration management of a web application can be improved by practices such as secure administration interfaces, secure configuration store, maintain separate administration privileges, and use least privileged process and service accounts

Sensitive Data: Sensitive data refers to the information that must be protected in memory, over the wire or in persistent store[17]. Applications that deal with private user information such as credit card numbers, addresses, and medical records, should always make sure that these sensitive data remain private and unaltered. In addition, strong secrets that used for the application's implementation such as passwords, database connection strings and credit card must be secured. Handling of sensitive data of a web application can be improve by practices such as try to avoid from storing secrets, do not store secrets in code, do not store database connections, passwords, or keys in plain text, and avoid storing secrets in the LSA (Local Security Authority).

Session Management: A session is a series of related interactions between a user and a web application. Session management refers to the way an application manages and protects these interactions[17]. It is an application-level responsibility, because web applications are built on the stateless HTTP protocol. Session security is critical to the overall security of an application. Session management of a web application can be improve by practices such as use SSL (Secure Sockets Layer) to protect session authentication cookies, encrypt the contents of the authentication cookies, limit the session life time, and protect session state from unauthorized access.

Cryptography: Cryptography provides few kind of service. First is the privacy service which keeps a secret such as passwords, database connection strings, and credit card numbers confidential. Non-reputation service will make sure that a user can not deny sending a particular message. Integrity is use to prevent data from altered. Lastly, authentication service is use to confirm a message sender's identity.

Web applications frequently use cryptography to secure data in persistent stores or as it is transmitted across networks. Practices such as do not develop own cryptography, keep not encrypted data close to algorithm, use correct algorithm and correct key size, and secure encryption key, is useful in improving web application's security.

Parameter Manipulation: Parameter manipulation refers to the way an application safeguards tampering application parameters such as query string arguments, form fields, and cookies values and the way an application processes input parameters[17]. It is crucial to protect parameters enforcing their validity and fit with the application logic. An attacker may modify the value of any parameters that sent between the client and web application with parameter manipulation attacks. Therefore, all input must be check for maximum number of character before any use of the parameters. It is important to decide which parameters might receive special characters and make them exception where only the specific characters are allowed and potentially dangerous sequences are eliminated. Parameter manipulation of a web application can be improved by practices such as encrypt sensitive cookie state, make sure that users do not bypass checks, validate all values sent from the client via scripting language, and do not trust HTTP header information.

Exception Management: Secure exception handling is helpful in preventing certain application-level DoS attacks and it can also helpful preventing valuable system-level information such as database schema details, operating system versions, stack traces, file names, and path information which is useful to attackers from being returned to the client. A centralized exception management and logging solution, as well as consider providing hooks into exception management system is a good approach to support instrumentation and centralized monitoring to help system administrators. Exception management of a web application can be improved by practices such as do not leak information to the client, log detailed error messages, and catch exceptions.

Auditing and Logging: Auditing and logging refer to the way an application records security related events [17]. Audit and log activity should be implemented across the tiers of application. Suspicious-looking activity can be detected by using logs. This frequently provides early indications of a full-blown attack and the repudiation threat where users deny their actions are addressed by the help of the logs. Legal proceedings require log files to prove the wrong doing of individuals. If the audits are generated at the precise time of resource access and by the same routines that access the resource, then auditing is considered to be most authoritative[17]. Web application's security can be improved by practices such as audit and log access across application tiers, consider identity flow, log key events, secure log files, and back up and analyze log files regularly.

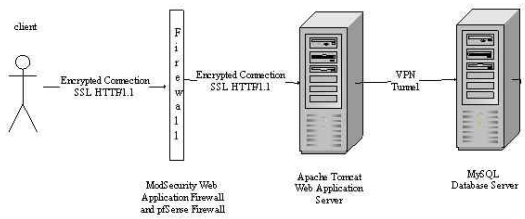
3.3 Summary

In this chapter, we have analyzed requirements need in order to develop a secure web application for e-business organizations. Every organization has own policy for the web application. Therefore, the web application should be developed by following the policies and standards which defined by the organization. Developers have to identify the network

infrastructure components and web application topology in order to build a secure web application. Security should be a focal point of web application design. Evaluating the web vulnerability of existing web application will give developers information on how to develop a secure web application. A secure web application must have a design of a solid authentication and authorization state. Majority of application level attacks rely on maliciously formed input data and poor application input validation. Therefore, input validation is being considered as the most important security issue for web applications.

Ⅳ. Logical & Physical Design

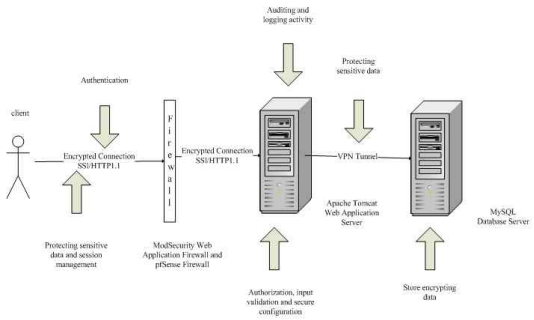
4.1 Logical Design



<Fig. 1> Logical design architecture of 3-tier web application

<Fig. 1> shows a web application architecture consists of 3-tiers which are client (presentation tier), web application server (business tier), and database server (data tier). This web application design is based on two objectives: 1) web application architecture with depth security, 2) use open source software to develop the secure web application.

(1) *Web Application Architecture with Depth Security*: Depth security approach always centers on the idea of multiple layers of security to protect vital assets[15]. Rules of the depth security approach are layered protections, defense in multiple, and defense through diversification. Layers of security include authentication, authorization, session management, input validation, secure server configuration, web application firewalls, auditing and logging activity, and data encryption.



<Fig. 2> Logical design architecture with layer of security

Authentication: Only specific individuals and the users are allowed to access restricted area after their identity is authenticated[18]. In Tomcat, the Servlet specification provides integration with the Java Authentication and Authorization Service (JAAS) API. In our web application, we more focused on the form authentication mechanism provided by Tomcat. Form authentication is an authentication method where web applications can use the application layer authentication to validate user credentials.

Protecting sensitive data and session management: Sensitive data should be sent over secure channels that are designed to prevent unauthorized interception of that information while in transit. In our web application, the complete transaction will be encrypted using SSL for all private areas of the application. SSL

provides a method for transparently encrypting data packets during a session. The web application server will need to be equipped with and operate under the HTTPS protocol, which will ensure all levels of protection. Thus, the data in transit can be considered fully secure.

Encrypting sensitive data: Encryption is a key aspect in providing security to the web application[18]. Password encryption in our web application is implemented using one way hash encryption because it is a collision free mechanism that guarantees no two different values will produce the same digest. By using this algorithm, there is no fear of losing encryption key and reduce the maintenance cost of cycle the encryption key periodically.

Authorization: The web application server will authorize user to access resources (using access control list) that they have restricted on[18]. In Tomcat, the access to the web application is controlled via a security realm. In our web application, we are using a user database realm for simplicity.

Input validation: Lack of proper input validation is the number-one cause of web application security issues [19]. In our web application, we perform input validation using JavaScript which has several properties. The first property is character-set, where only accept data which contains characters. Second is to ensure that the data falls within a restricted minimum and maximum number of bytes. This is to ensure the structure of the data is consistent with what is expected. Whenever possible, the user will not have to enter data himself, but be able to select from radio buttons, drop-down lists, or check boxes.

Secure configuration: In order to secure web application configuration, it is important that configuration management can only be accessible by authorized operators and administrators[18]. A key part to secure configuration is to enforce strong authentication over administration interfaces by using certificates, and limit the use of remote administration and require administrators to log on locally. These processes have limited the damage that can be done when an attacker manages to take control of a process.

Auditing and logging activity: Log files will secure by using ACLs (Access Control Lists) and restrict access to the log files. This will make it more difficult for attackers to tamper with log files to cover their tracks. The log files only will be accessible by administrators and will be analyzed regularly. Log files will be retrieved and moved to offline servers for analysis.

2) *Open Source Software that use in Web Application*: Cost-effectiveness is an important issue in developing a secure web application. According to a survey released by security firm Mcfee, small businesses have, for the most part, frozen spending on security in 2009, despite an increase in perceived threats. Due to this issue, open source software is the best solution in helping small business company to develop a simple, intelligent, and secure web application with minimum cost. Below is the open source software that we use in implementing our web application.

- Web application server: Apache Tomcat Server (Java-based) version 5.0, JDK version 1.5 [20]
- Database system: MySQL Server version 5.0 [21]
- SSL/HTTPS encrypted connection: OpenSSL version 0.9.7 [22]
- VPN (Virtual Private Network) tunnel: OpenVPN version 2.0.9 [23]
- Web application firewall: ModSecurity version 2.1.5 [24]

- Firewall between client and web application server: pfSense Firewall version 1.0.1 [25]

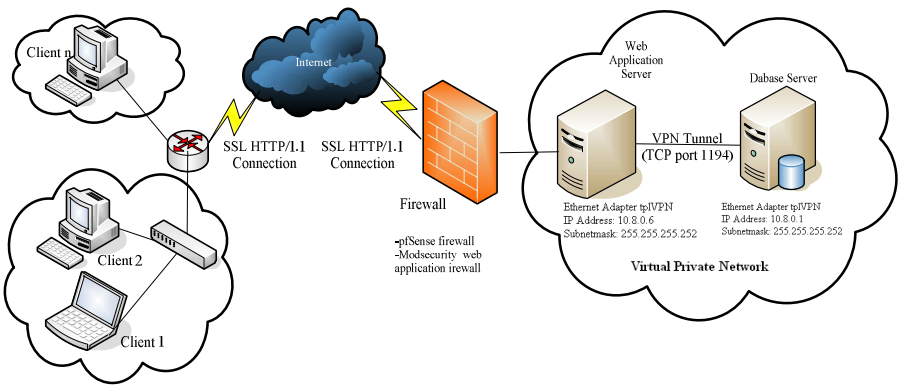
4.2 Physical Design

<Fig. 3> shows the physical design of the web application. In 3-tier architecture, it can support multiple clients at a same time. The clients are able to display information related to such services as browsing ordering or purchasing contents. The clients can be any client at the intranet or internet. The web application server will act as the mediator for converting the client requests into database understandable formats and send the requests to database server. The database server will retrieve the data according to the command given by the client and convert those results in the client understandable formats.

The web application server and database server are connected through a VPN tunnel at TCP port with port number 1194 to transfer sensitive data. The application server will act as the VPN client with ip address 10.8.0.6, while the database server will act as the VPN server with ip address 10.8.0.1. Encrypted connection SSL/HTTPS is use to protect sensitive data such as username, password, and credit card number that send between client and web application server through port 443. SSL/HTTPS and VPN tunnel also used in preventing session hijacking.

All users' identity will be authenticated before they can gain access to resource. Web application server will authorize user, validating the input data from user, and providing secure configuration. User authentication, authorization, and data input validation process is examined using Tomcat features and JavaScript input validation technique. The password will be encrypted using hash algorithm and stored in database.

Firewall is placed in front of web application server to provide a protective layer between the resources of a private network and users from other networks. All traffic moving from inside the corporate network to outside the network and vice versa must pass through the firewall. Only authorized traffic is allowed to pass through it. A network firewall provides traffic filtering at the IP and transport layer, while a web application firewall (WAF) is an appliance or server application



<Fig. 3> Physical design

that watches HTTP/HTTPS conversations between a client browser and web application server at application layer.

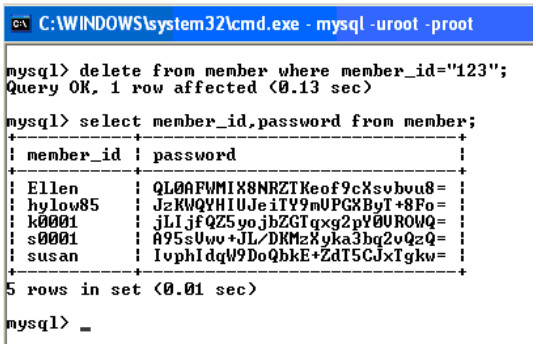
V. Implementation

(1) *Password Encryption*: Password encryption is developed to protect user data by using Java encryption methods. There is a registration module in our web system where a user is asked to choose a username and password, fill in some personal details. This data later get stored in the database. In order to let web page user peace of mind while using the web site, password encryption is developed that will be kept well-protected (read encrypted) member password (see <Fig. 5>).

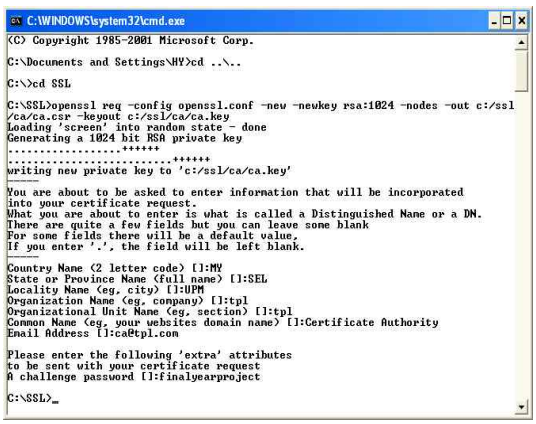
```
public synchronized String encrypt(String plaintext)
        {
          MessageDigest md = null;
          try
          {
            md = MessageDigest.getInstance("SHA");        }
          catch(NoSuchAlgorithmException e)
          {
            e.printStackTrace();
             }
          try
          {
            md.update(plaintext.getBytes("UTF-8"));
          }
          catch(UnsupportedEncodingException e)
          {
            e.printStackTrace();
          }
          byte raw[] = md.digest();
          String hash = (new BASE64Encoder()).encode(raw);
          return hash;
        }
```

<Fig. 4> Sample code for password encryption

(2) *Secure Sockets Layer (SSL)*: SSL is used to ensure secure transactions between web application servers and browsers. The SSL is implemented using OpenSSL. OpenSSL is open source that can be used to set up own certificate authority and sign the certificates created by the CA (Certificate Authority) (see <Fig. 6>).

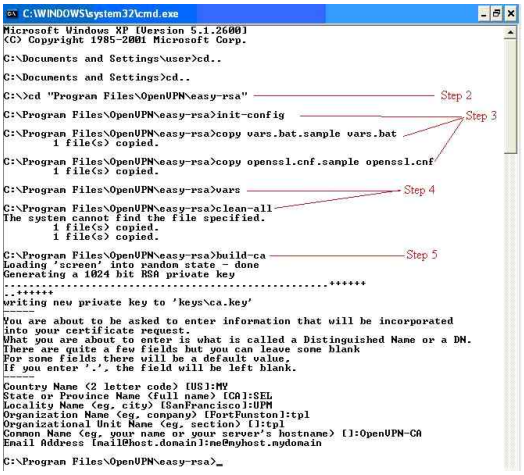


<Fig. 5> Encrypted password in database



<Fig. 6> Create private key and certificate request

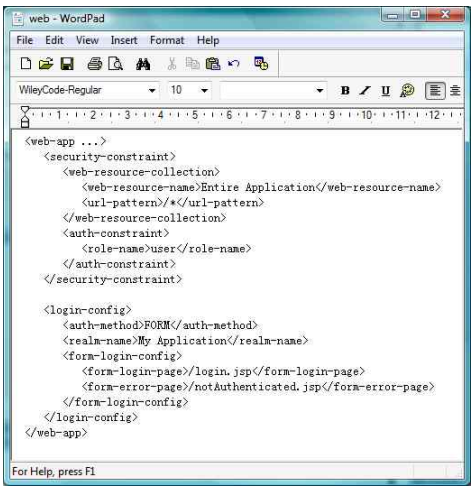
(3) *Virtual Private Network (VPN)*: VPN is used to ensure secure data transmission between web application server and database server. With establish of VPN tunneling all communications between the client machine and the VPN server are encrypted. The VPNis implemented using OpenVPN (see <Fig. 7>). OpenVPN supports bidirectional authentication based on certificates, meaning that the client must authenticate the server certificate and the server must authenticate the client certificate before mutual trust is established.



<Fig. 7> Generate the master CA certificate & key

(4) *User Authentication*. In order for a Web application to use one of the authentication mechanisms, it must be configured to do so inside its deployment descriptor (web. xml file). This is accomplished by adding <security-constraint> and <login-config> elements to the <web-app> element.

<Fig. 8> shows code excerpt, the <security-constraint> element is used to define a portion of the application that is restricted to users belonging to a specific role. The <url-pattern> element uses URL pattern matching to determine the protected portion of the application (in this case, the entire application), and the <role-name> element is used to restrict that portion of the application to authenticated users who belong to the "user" role. The <login-config> element is used to specify how users authenticate with the Web application. <auth-method> determines which of the authentication mechanisms described here is used. Possible values include BASIC, DIGEST, FORM, and CLIENT-CERT. We have chosen FORM, the <form-login-config> element must be nested in the <login-config> element. <form-login-config> identifies which page in the web application is used to authenticate the user (/login. jsp) and which page is displayed when authentication fails (/notAuthenticated. jsp). No page is configured to be displayed when authentication succeeds. Instead, the user is presented with the URL that triggered the authentication in the first place.



<Fig. 8> User authentication code

(5) *Firewall*. The firewall is implemented using pfSense. pfSense is installed to hard drive run option 99 from the shell menu now. The configuration will be transferred to the hard drive by the installer.

The web application firewall is implemented using ModSecurity. The following are the steps to implement web application firewall.

Step 1: Download the source code from CVS repository

$ cvs -d:pserver:anonymous@cvs.sourceforge.net:/cvsroot/mod-security login

$ cvs -z3 -d:pserver:anonymous@cvs.sourceforge. n

et:/cvsroot/mod-security

> co mod_security

Step 2: Installing as DSO

<apache-home>/bin/apxs -cia mod_security.c

Step 3: Stop and then start Apache

<apache-home>/bin/apachectl stop

<apache-home>/bin/apachectl start

Step 4: ModSecurity configuration directives are added to configuration file (typically httpd.conf) directly.

IfModule mod_security. c>

mod_security configuration directives

#...

</IfModule>

Step 5: Since Apache allows configuration data to exist in more than one file it is possible to group ModSecurity configuration directives in a single file (e. g. modsecurity. conf) and include it from httpd. conf with the Include directive:

Include conf/modsecurity. conf

Step 6: Turning filtering on

SecFilterEngine On

Step 7: POST scanning

SecFilterScanPOST On

Step 8: URL Encoding Validation

SecFilterCheckURLEncoding On

Step 9: Unicode Encoding Validation

SecFilterCheckUnicodeEncoding On

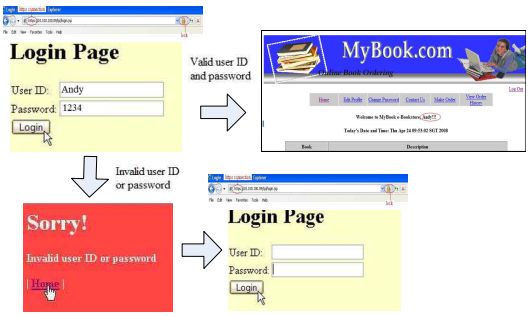
Step 10: Byte range check

SecFilterForceByteRange 32 126

Step 11: Advanced filtering

SecFilterSelective "REMOTE_ADDR|REMOTE_HOST" KEYWORD

(6) *System Layout for Web Application with Open Source Software*: <Fig. 9> shows the form authentication process of the web application. If proceed by clicking the login button without entering a user ID and password or wrong user ID or wrong password, the web application defines an error page for form-based authentication. The web container automatically invokes this page if the user ID and password that the user supplied in the login form are not found in the server's principal registry. This internal container action happens in response to an HTTP error status 401. Our error page has been coded to silently redirect to the login page. When a valid user ID and password are entered, the user is allowed enter into her own homepage. The homepage shows the user name that we were authenticated by form-based authentication. The connection between web browser and web application server is secured by SSL on port 443.

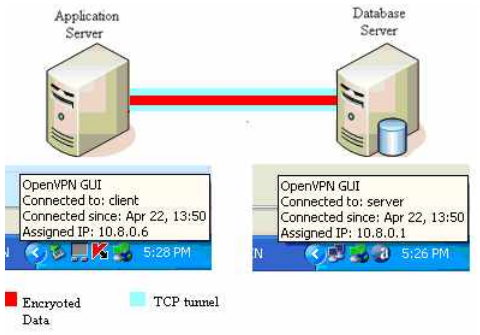


<Fig. 9> Form authentication

<Fig. 10> shows VPN tunneling (encrypted connection) established between web application server and database server. The web application server is configured to be a VPN client (IP address 10.8.0.6). The database server is configured to be a VPN server (IP address 10.8.0.1). <Fig. 11> shows the passwords stored database that are encrypted using SHA-1 algorithms.

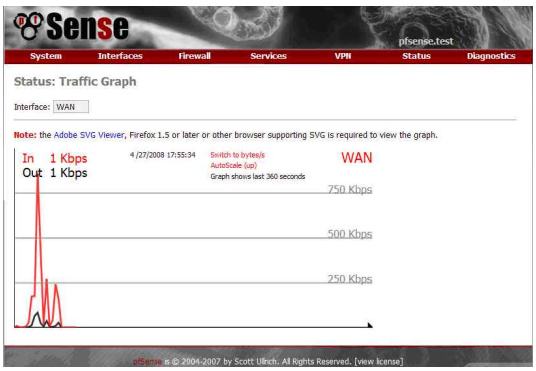
The ModSecurity debug log will, when configured

properly, provide a ton of information for every HTTP request. There can be pages of text for each request, especially if a file is being uploaded. The ModSecurity audit log will, when enabled, contain the complete request and the headers from the response. <Fig. 12> shows the pfSense firewall traffic graph at WAN interface.



<Fig. 10> VPN tunnel

<Fig. 11> Encrypted password in database



<Fig. 12> pfSense Traffic graph

(7) *Discussion*. There are several reasons to support us to choose open source software to develop our web application. The first reason is open source software can solve problems (bugs or errors) with more flexibility than commercial software. The development cost of web applications using open source software is less than those web applications using commercial software and we are allowed to modify everything that we find appropriate to suit our web application needs. In case of a failure, open source software used in our web application has the capability to re-establish its level of performance and recover the data directly affected in a timely manner. Even though the complexity of open source software is less than commercial software, open source software has the capability to provide appropriate response and processing times, as well as throughput rates when performing its function, under stated condition, and the development speed of open source software is faster than commercial software. In terms of maintainability, open source software has the capability to enable a specified modification to be implemented, avoid unexpected effects from software modifications and enable modified software to be validated. Besides, the used ofopen source software to develop our web application give us a simple license management where we had free from monitoring and counting license compliance. We can install open source software several times and used from any location without worrying license problems. Another benefit of open source software is the public collaboration where the software packages are developed by thousand of talented developers and there are many existing online forum which support independent peer reviews and has rapid evolution of source code[26].

Most of the web application systems these days are backed by databases. Web applications of any significant size typically have the database on a different host. Therefore, most of the web applications may focus more on how to protect the data that save in the database server and the accesses to the database server, than the confidential of the data that transfer between web application server and database server. This may give the attackers an opportunity to eavesdrop the data transfer between web application server and database server. Due to this issue, we have developed a VPN tunnel which can transfer the data between the web application server and database server in encrypted form and disabled remote access to public interface on database server. Even though VPN tunnel is normally used by most web application systems to protect exchanging data between client browse and web application server in a private network and we developed the web application server and database server at the same network but in real world the servers are most properly located in different network. Therefore VPN tunnel is useful to prevent eavesdropping attack when web application server and database server is located at different network. VPN is used to transfer critical important information, therefore, it is important to ensure that the performance provide by the network is constant over time[27].

Most of the web application systems focus more on network firewalls and have less consider on the web application firewalls. The web application firewall is the technologies that use to block application layer attacks such as input validation and SQL injection. Normally there are not many web application systems adding security technologies at the application layer. This is because adding security technologies at the application layer will slow down the system performance. However in our web application, we have added web application firewall which can filter the traffic at the application layer and prevent from application layer attacks such as SQL injection and input validation.

In our web application system, we established SSL connection between web application server and client browser. SSL is used to avoid that sniffing of data and session IDs when we have some application where security is critical[28]. In our web application system, all users'password is encrypted using one way hash algorithm while most of the other web application systems used standard ciphers for password encryption such as the Data Encryption Standard (DES) or RC2 which require an encryption key. These standard ciphers will need quite an amount of money for recycling the encryption key periodically and has to always keep track on the encryption key. If the encryption key is compromised, the entire database is compromised, and user privacy is no longer being protected. Due to this issue, we had chose one way hash algorithm to encrypted users'password because it did not require a key and produced an irreversibly encrypted cipher-text. The use of this algorithm has helped us to meet our first goal which is cost-effectiveness and solved the problems of user privacy. Even if the password in database were compromised, it would still be hard for an intruder to recover the original passwords, and user privacy is protected.

VI. Conclusion

We use open source software to develop security features that can provide security service to the web application. For example, we used OpenSSL to develop SSL, OpenVPN to develop VPN tunnel, pfSense to develop network-layer firewall and ModSecurity to develop web application firewall. The established SSL enables the exchanging sensitive information between the client browse and the web application server using HTTPS connection. This makes the exchanging sensitive information is secure. VPN tunnel is established between the web application server and the database server and has made the user data transmitted through a secure tunnel. Password Encryption is developed using one-way hash algorithm which can provide well-protected of member password because all the passwords are store in encrypted format. Network layer firewall provides a protective layer between the resources of a private network and users from other networks. The network-layer firewall does not protect at the application layer, therefore the web application firewall is the only viable option. The web application firewall is used to identify website vulnerabilities and block all known and emerging web and web service application attacks. These security features that we implemented using open source software can mitigate the security risks and provide secure, manageable, available, and scalable solutions to protect the web application system.

Related to the future research, there are several suggestions. First, N-tier web application architecture can be the future work to make client and server works more independent and scalable, and improve the performance. Second, there may be the future enhancement to develop a bridge VPN because our VPN is a routing VPN. By having the bridge VPN, windows files can share across the VPN without a Samba server. Finally, this paper lacks detailed on threat modeling and testing. The future enhancement on these issues can provide a consistent and more manageable way for the secure web application.

References

[1] Newfoundland and Labrador, Enterprise Architecture Guidelines and Best Practices, Version 3. 8, 2009.

[2] Desmet, L., Jacobs, B., Piessens, F., Joosen, W., "A Generic Architecture for Web Applications to Support Threat Analysis of Infrastructure components," DistriNet Research Group, Katholieke Universiteit Leuven, Belgium, 2005.

[3] http://www.securityfocus.com/brief/1029, Small, medium firms cut security budgets

[4] Petersen, J., Benefits of using the n-tired approach for web applications, Adobe Systems Inc. http://www.adobe.com/devnet/coldfusion/articles/ntier.html, 2009.

[5] Shamsaie, A., Habibi, J., Ghassemi, F., Tierpeer: A three-tier framework for P2P, IJCSNS International Journal of Computer Science and Network Security, VOL. 7 No. 2, pp292-301, 2007.

[6] Mains, B., Introduction to 3-Tier Architecture, DotNet- Slackers.com, http://dotnetslackers.com/articles/net/IntroductionTo3TierArchitecture.aspx, 2008.

[7] Thacker, N., 3-Tier Web Application Development. http://weblogs.asp.net/nannettet

hacker/archive/2008/03/05/3-tier-web-application-development.aspx, 2008.

[8] Zafar, M, F, Naheed, F, Ahmad, Z, and Anwar, M, M, Network Security: A survey of Modern approaches, The Nuclues, A Quarterly Scientific Journal of Pakistan, 2008, pp11-31.

[9] Lawrence, E, Newton, S., Corbit, B., Braithwaite, R., Parker, C., Technology of internet business, John Wiley & Sons Australian, Ltd, 2002, pp243-273.

[10] http://it.toolbox.com/wiki/index.php, Man in the Middle Attack

[11] Web Application Security Consortium, Web Application Security Consortium: Threat Classification, 2004, pp10-62.

[12] Pettit, S., "Anatomy of Web Application: Security Considerations," White Paper, Sanctum Inc., 2001.

[13] Meier, J. D., Mackman, A., Dunner, M., Vasireddy, S., Escamilla, R., Murukan, A., Web Application Security Fundamentals, Microsoft Corporation, 2003.

[14] Taylor, A., Alexander, D., Finch, A., Sutton, D., Information Security Management Principles, The British Computer Society, 2008.

[15] Viega, J., McGraw, G., Building Secure Software - How to avoid security problems the right way, Addision-Wesley, 2002.

[16] Satani, G., "Top 10 Web Service Security Requirements," http://builder.com.com/article.html, 2002.

[17] Curphey, M., Scambray, J., Olson, E., "Improving Web Application Security: Threats and Counter-measures Patterns & Practices," Microsoft Corporation, 2003.

[18] Curphey, M., et al., "A Guide to Building Secure Web Applications (OWASP Guide)," Creative Commons Attribution ShareAlike 3.0, 2002.

[19] White Hat Security, "Web Application Security 101: Real-world examples, tools and techniques for securing websites," White Paper, White Hat Security, Inc, 2005.

[20] The Apache Tomcat, http://tomcat.apache.org/

[21] MySQL Database Server 5.1.36, http://download.cnet.com/MySQL-Database-Server/

[22] Open SSL, http://www.openssl.org/

[23] OpenVPN, http://www.openvpn.net/

[24] ModSecurity, http://www.techiecorner.com/

[25] pfSense firewall, http://www.linux.com/

[26] Cheliotis, G, From open source to open content: Organization, licensing and decision processes in open cultural production, Decision Support Systems Volume 47, Issue 3, Elsevier Ltd, 2009, pp229-244.

[27] Forte, D, SSL VPN and return on investment: A possible combination, Network Security, Volume 2009, Issue 10, Elsevier Ltd, 2009, pp17-19.

[28] Bock, J, Session-Cookies and SSL, study research project at the EISS, University of Karlsruhe, 2008.

■ 저자소개 ■

김 창 수
Kim, Chang Su

2004년 3월~현재 영남대학교 경영학부 교수
2010년 3월~현재
미국 Carnegie Mellon University, School of Computer Science, 객원교수
2006년 7월~2007년 8월
영국 런던대학교 (University of London) 객원교수
2005년 12월~2006년 2월
미국 University of Texas at Austin, 객원교수
2008년 4월 영국 런던대학교 컴퓨터과학과 (컴퓨터과학 박사수료: MPhil)
2003년 2월 영국 런던정경대 (London School of Economics: LSE) 정보시스템학과(정보시스템박사)
1999년 8월 중앙대학교 경영학과 (경영학박사)

관심분야 : e-비즈니스, 유비쿼터스 컴퓨팅, 정보시스템 분석 및 설계
E-mail : c.kim@ynu.ac.kr

유 혜 인
Low, Hooi Yin

2009년 9월~현재
영남대학교 대학원 경영학과 경영학석사 과정
2008년 8월 말레이시아 푸트라 말레이시아 대학교 (University Putra Malaysia) 컴퓨터공학과 (컴퓨터공학사)

관심분야 : e-비즈니스, 정보시스템분석 및 설계

이 용 주
Lee, Yong Ju

2008년 1월~현재
경북대학교 컴퓨터정보학부 교수
1998년 8월~2007년 12월
상주대학교 컴퓨터공학과 교수
1989년 3월~1994년 7월
삼보컴퓨터 근무
1985년 3월~1989년 2월
KIST시스템공학센터 연구원
1997년 8월 한국과학기술원 컴퓨터공학전공(공학박사)
1985년 2월 한국과학기술원 정보검색전공 (공학석사)

관심분야 : 웹데이터베이스, 정보검색
E-mail : yongju@knu.ac.kr

논문접수일 : 2009년 1월 23일
수 정 일 : 2010년 2월 16일
게재확정일 : 2010년 2월 25일