A Study on Applying Software Security to Information Systems: E-Learning Portals

Chee Chern Lim and Jesse S Jin

School of Design, Communication and I.T, University of Newcastle, NSW 2308, AUSTRALIA

Summary

All people treasure privacy, linking it to their concept of personal freedom and well-being. Unfortunately, the Internet's great promise - that it facilitates the collection, re-use and instantaneous transmission of information - can, if not managed carefully, diminish personal privacy. It is therefore essential, in any application developed, to assure personal privacy in the networked environment, if people would like to feel free and comfortable to conduct business using that particular application. Thus, any Internet application must be secure and reliable. If Internet users do not have confidence that their communications and data are safe from unauthorized access or modification, they will be unlikely to use the Internet on a routine basis.

In this paper, we perform a case study on applying common security to an e-learning portal. First, the paper will explore the risks behind the Internet security and privacy fundamentals.. Next, the paper will review the common Internet application vulnerabilities. Then the paper will focus on securing such applications. In particular, we will provide details of the implementation of a strong authentication and authorization mechanism. We utilize the session to achieve secure environment for e-learning portals.

Key words:
Internet security, e-learning portal security.

Background

Increasingly, organizations are connecting to the Internet to establish a business and electronic commerce presence and to access information rapidly. When an organization's networks are connected to the Internet without adequate security measures in place, the organization will become vulnerable to attacks from external adversaries. These risks include:

- Loss of confidentiality of business information (such as financial records, strategic planning data, engineering models and prototypes, marketing plans, medical records), as well as inability to guarantee the integrity of such information;
- Loss of availability of mission-critical services such as EDI (electronic data interchange), ERP (enterprise resource planning), just-in-time inventory controls, and electronic mails;
- Exposure of critical data about your information infrastructure that can be used by your adversaries in planning their attacks;
- Legal liability, regulatory liability, or public loss of confidence when your adversaries use one of your computers to carry out attacks against other organizations;
- Vandalism of public information services (such as your public Web site) [1].

The online application or Internet portal must be secure and reliable. If Internet users do not have confidence that their communications and data are safe from unauthorized access or modification, they will be unlikely to use the Internet on a routine basis.. The key components to secure data information are:

- Secure and reliable telecommunications networks;
- An effective means for protecting the information systems attached to those networks;
- An efficient means for authenticating and ensuring confidentiality of electronic information to protect data from unauthorized use; and
- Well-trained portal users who understand how to protect their systems and their data..

Accomplishing these components requires effective use of a range of technologies such as encryption, authentication, password controls and firewalls as well as consistent use of those technologies, all supported globally by trustworthy key and security management infrastructures.

In this paper, first it reviews the common Internet application vulnerabilities.. After that, it will focus on securing such applications. In particular, it will provide details of the implementation of a strong authentication and authorization

Manuscript received February 10, 2006.
Manuscript revised February 27 , 2006.

mechanism which utilizes the session to achieve secure environment for e-learning portals.

2. Common Application Vulnerabilities

In the article *Common Security Vulnerabilities in e-commerce Systems*, Mookhey states that: *The tremendous increase in online transactions has been accompanied by an equal rise in the number and type of attacks against the security of online payment systems. Some of these attacks have utilized vulnerabilities that have been published in reusable third-party components utilized by websites, such as shopping cart software. Other attacks have used vulnerabilities that are common in any web application, such as SQL injection or cross-site scripting* [2]. Successful exploitation of these vulnerabilities can lead to a wide range of results. Information and path disclosure vulnerabilities will typically act as initial stages leading to further exploitation. SQL injection or price manipulation attacks could cripple the website, compromise confidentiality, and in the worst cases cause the e-commerce business to shut down completely [2]. These types of Vulnerabilities include:

- **SQL Injection**: SQL injection refers to the insertion of SQL meta-characters in user input, such that the attacker's queries are executed by the back-end database. Typically, attackers will first determine if a site is vulnerable to such an attack by sending in the single-quote (') character. The results from an SQL injection attack on a vulnerable site may range from a detailed error message, which discloses the back-end technology being used, to allowing the attacker to access restricted areas of the site because he manipulated the query to an always-true Boolean value. It may even allow the execution of commands at operating system levels;

- **URL Manipulation**: Attacks try to manipulate or access important information if the application implements GET request to send important parameters on the URL. The parameters can be manipulated to give undesired results. The best solution is to avoid sending critical parameters in a query string. In addition, the application should validate the parameters with a session token;

- **Data Manipulation** (such as price manipulation): This is a vulnerability that is almost completely unique to online shopping carts and payment gateways. In the most common occurrence of this vulnerability, the total payable price of the purchased goods is stored in a hidden HTML field of a dynamically generated web page. An attacker can use a web application proxy such as Achilles [3] to simply modify the amount that is payable when this information flows from the user's browser to the web server;

- **Buffer Overflows**: Buffer overflow vulnerabilities are not very common in shopping cart or other web applications using Perl, PHP, ASP, etc. However, sending in a large number of bytes to web applications that are not geared to deal with them can have unexpected consequences. In our penetration tests, we have shown that it is possible to disclose the path of the PHP functions used by sending in a very large value in the input fields;

- **Cross-site Scripting**: The Cross-site Scripting (XSS) [4] attack is primarily targeted against the end-user and leverages two factors:
    1. The lack of input and output validation being done by the web application;
    2. The trust placed by the end-user in a URL that carries the vulnerable website's name.

  The XSS attack requires a web form that takes in a user input, processes it, and prints out the results on a web page, which also contains the user's original input. It is the most commonly found in 'search' features, where the search logic will print out the results along with a line such as 'Results for <user_supplied_input>'.. In this case, if the user input is printed out without being parsed, then an attacker can embed JavaScript by supplying it as part of the input. By crafting a URL, which contains this JavaScript, a victim can be socially engineered into clicking on it, and the script executes on the victim's system;

- **Remote Command Execution**: The most devastating web application vulnerabilities occur when the CGI script allows an attacker to execute operating system commands due to inadequate input validation. This is the most common case with the use of the "system call" in Perl and PHP scripts. Using a command separator and other shell metacharacters, it is possible for the attacker to execute commands with the privileges of the web server;

- **Weak Authentication and Authorization**: Authentication mechanisms that do not prohibit multiple failed logins can be attacked using tools such as Brutus [5].. Similarly, if the web site uses HTTP basic authentication or does not pass session IDs over SSL (Secure Sockets Layer), an attacker can sniff the traffic to discover user's authentication and/or authorization credentials. Since HTTP is a stateless protocol, web applications commonly maintain state using session IDs or transaction IDs stored in a cookie on the user's system. Thus this session ID becomes the only way that the web application can determine the online identity of the user. If the session ID is stolen (say through XSS), or it can be predicted, then an attacker can take over a genuine user's online identity vis-à-vis the vulnerable web site. Where the algorithm used to generate the session ID is weak, it is trivial to write a Perl script to enumerate through the possible session ID space and break the application's authentication and authorization schemes.

3. Securing Internet Application

One of the leading institutions concerned with information systems security, the Carnegie-Mellon's CERT®, has contributed enormously to the development of security practices and frameworks. One of the most important methods they have created is the Security Knowledge in Practice method (or the SKiP method for short).. It consists of steps to secure network software, to harden a network (make it difficult to break into), to detect and respond to network intrusions, and then to improve the system based on a review of events [6]..

The following steps are included in the SKiP method:

1. Select system software from a vendor and customize it according to an organization's needs;
2. Harden and secure the system against known vulnerabilities;
3. Prepare the system so that anomalies may be noticed and analysed for potential problems;
4. Detect those anomalies and any other system changes that could indicate evidence of an intrusion;
5. Respond to intrusions when they occur;
6. Improve practices and procedures after updating the system;
7. Repeat the SKiP process as long as the organization needs to protect the system and its information assets.

Internet system offerings have proliferated over the last few years and today it is possible to find a suitable solution for any budget size. It is important do say, though, that not every solution is safe, and must be configured appropriately. One of the important steps is to identify tasks a system must perform and configure it to fulfil essential functions while eliminating those that are unnecessary or vulnerable [6]. Securing a system is a challenging task. It is often neglected, especially for a novice administrator.. SKiP has recommended the following measures for network administrators [6]:

- Eliminate services that are unneeded and insecurely configured;
- Restrict access to vulnerable files and directories;
- Turn off software "features" that introduce vulnerabilities;
- Mitigate vulnerabilities that intruders can use to break into systems.

4. Implement Strong Authentication and Authorization in Cyber Campus

Cyber Campus is a centralized learning management system which provides a user-friendly administrating, teaching and learning environment for instructors and learners [7]. Like Blackboard® [8], Cyber Campus consists of a web server and many client PCs as the interface from which instructors and learners interact with the server via WWW to offer e-learning services. Cyber Campus is an all-in-one web-based teaching and learning education system which integrates components including student management, course management, assessments, classroom allocation, communication, etc, into one complete package solution. It integrates web technology with database systems and provides a user-friendly administrating, teaching and learning environment. The system is designed by using the framework based on Web Content Component Model (WCCM), which is the suitable model of web application that supports maintenance of content oriented web applications [9].

Besides implementing mechanisms to prevent the common application vulnerabilities discussed in Section 2, Cyber Campus also implemented the following mechanisms to strengthen the security of the learning portal. These mechanisms include:

MySQL Access Privilege

The primary function of the MySQL access privilege is to authenticate a user connecting from a given host, and to associate that user with privileges on a database such as SELECT, INSERT, UPDATE, and DELETE. The MySQL privilege system ensures that all users may perform only the operations allowed to them. As a user, when he/she connects to a MySQL server, his/her identity is determined by the host from which the connection is made and the username is specified. The system grants privileges according to the identity and what the user wants to do. MySQL access control involves two stages [10]:

- *Stage 1*: The server checks whether the user is even allowed to connect;
- *Stage 2*: Assuming that the user can connect, the server checks each statement the user issues to see whether the user has sufficient privileges to perform it. For example, if the user tries to select rows from a table in a database or drop a table from the database, the server verifies that the user has the SELECT privilege for the table or the DROP privilege for the database.

Different access privileges should be setup for different users to provide better security for the online learning portal. Cyber Campus has classified three access privileges for three different access users (administrator, instructor and learner). These access privileges are defined depending on specific users and the users are identified by the location of the *logon* portal.

Secure Socket Layer and MD5 Encryption

In addition to the security protections via the database privilege settings and the system access restriction based on the IP setting, Cyber Campus is designed and implemented with a multi-layered security infrastructure. In order to guarantee a secure access to the system, Cyber Campus uses Secure Socket Layer (SSL) to transfer sensitive data such as login name and password over the Internet via HTTPS. Of particular importance is this final note: the importance of developing trusted certification services is that the service has to support digital signatures and permit users to authenticate the other side with whom they are communicating on the Internet. The use of cryptographic keys and the Public Key Infrastructure (PKI) is required to support such framework with digital signatures and confidentiality.

In case the unavailability of SSL, the system supports a client's side MD5 function in JavaScript. Thus, instead of sending raw password to the server, the user's password is encrypted at the client's side before transmitting to the server. Raw passwords are not stored in the database; only encrypted passwords are stored and retrieved for comparison during the user authentication. In addition, MD5 is one-way encryption which avoids the raw passwords being stolen from the database. However, the JavaScript MD5 function only works when the JavaScript is enabled in the client's browser.

Strong Session Tracking

Cyber Campus has implemented a strong session tracking procedure to authenticate authorized users to access the learning portal. Raw password is never stored at the database, the Cyber Campus authenticate logon user with the encrypted password. To avoid the case of weak authentication and authorization in Section 2, Cyber Campus implements new mechanism to strengthen session tracking. Session ID is still being used to determine the online identity of the user. However, the authentication will also use the user's IP in conjunction with the session to further ensure the identity of authorised user.

Once the user has been authorised and logon successfully to the learning portal, the unique login name will be stored in the session register, a key value will be calculated based on the unique login name and user's IP address. This key value will be registered in session for authenticating the user during an active session tracking process. Every time the user requests a page in the portal, the script will calculate a value based on the user's login name registered in the session and user's IP address determined by using PHP functions. Then system will verify the genuine identity by comparing the calculated key value with the key value (calculated when the user first logins) in the session register. This mechanism prevents the possible attack even if the session ID is stolen. The following PHP function is used to determine the IP address belong to the portal user.

```
<?php
  $ip = getenv("REMOTE_ADDR");  // get IP
?>
```

5. Conclusion

All the issues mentioned in this paper are important and necessary to create a secure online application. The paper is focus on issues that can be brought about by an attacker relatively easily, just by giving Internet application information the Web does not expect. Such attacks could be malicious, such as ..URL manipulation, SQL injection, etc, or accidental such as weak session tracking.. It should be noted that developers should follow all of these guidelines perfectly but they may still have an insecure application. Absolute 100% security is almost impossible, but these guidelines will provide an application as hack-resistant as possible.

References

[1] CERT Security Improvement Module Deploying Firewalls, http://www.cert.org/security-improvement/modules/m08.html, accessed on 20 February

[2] Common Security Vulnerabilities in e-commerce Systems, http://www.securityfocus.com/infocus/1775 accessed on 20 February 2006.

[3] Maven Security Consulting Inc - Info Security Consulting & Training Services – Achilles, http://www.mavensecurity.com/achilles accessed on 20 February 2006.

[4] The Cross Site Scripting FAQ, http://www.cgisecurity.com/articles/xss-faq.txt accessed on 20 February 2006.

[5] Brutus - The Remote Password Cracker, http://www.hoobie.net/brutus/ accessed on 20 February 2006.

[6] Securing Networks Systematically — the SKiP Method, http://www.cert.org/archive/pdf/SKiP.pdf accessed on 20 February 2006.

[7] C Lim, R Xu & J S Jin. E-learning via augmented reality on adaptive LMS.. *AACE E-Learn 2005 -World Conference on E-Learning in Corporate, Government, Healthcare, & Higher Education*, Canada, October 2005, pp..2199-2204.

[8] Blackboard Inc (http://www.blackboard.com) accessed on 20 February 2006.

[9] C Lim, M Yu, J S Jin. Generic e-learning data structure and Web teaching, *2005 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE 2005)*, Hong Kong, March 2005, pp 564-569.

[10] MySQL 3.23, 4.0, 4.1 Reference Manual :: 13.5.1.2 GRANT Syntax, vol. 2004, http://dev.mysql.com/doc/refman/4.1/en/grant.html, accessed on 23 February 2006..