Teaching Security Techniques in an E-Commerce Course
Liu, Chang;Mackie, Brian G
Journal of Information Systems Education; Spring 2006; 17, 1; Research Library
pg. 5

Teaching Tip

Teaching Security Techniques in an E-Commerce Course

Chang Liu
Brian G. Mackie
Department of Operations Management and Information Systems
College of Business
Northern Illinois University
De Kalb, IL 60115 USA
cliu@niu.edu, bmackie@niu.edu

ABSTRACT

Over the past few years, more and more companies have been investing in electronic commerce (EC) by developing and implementing web-based applications on the Internet. While EC can help improve business services and increase customer satisfaction, it also brings increased security risks to those companies implementing it. Developers of EC web sites have to incorporate ways to systematically identify and eliminate security vulnerabilities within their EC applications. This paper describes how Microsoft ASP.Net can be used to assist students in exploring ways to increase the security of EC applications. The hands-on component covers useful techniques for improving application robustness in the pre-sales, online-sales and after-sales phases of an EC application. The paper concludes with a discussion of "lessons learned" and suggestions for effectively teaching security in an EC design course.

Keywords: Electronic Commerce, Security, Application, Course Development, ASP.Net

1. INTRODUCTION

Electronic Commerce (EC) has allowed organizations to enhance their economic growth, reduce barriers to market entry, improve efficiency and effectiveness, keep inventories lean, and reduce costs (Hof and Hamm, 2002). Research indicates that EC will continue to grow and that it will change every kind of business, online as well as offline. In order to achieve the most benefit, businesses need to build security into their EC web sites (Gartner Group, 2005). Many security experts believe that implementing firewalls and Intrusion Detection Systems (IDS) alone are inadequate, as security is a continual process and it needs to be addressed across the computer network layer, the web host layer and the application layer (Main, 2004).

Recent studies show that the number of severe computer breaches of EC applications have grown steadily and the application layer is a frequent point of attack by intruders in recent years (Computer Security Report, 2002). Given the magnitude of real and potential losses, there is a need to build a systematic framework to address security issues in web-based EC applications. Although some universities have either started to expand their curriculum by developing security related courses or to integrate relevant security content into the technical courses in their IS degree programs, little has been done to emphasize security within EC application design courses offered at universities.

Internet intruders can create havoc and produce catastrophic results by exploiting weaknesses within EC applications. The intruder's best ally is poorly written or inadequately tested software. Therefore, students who take an EC web application design course need to be given ways to systematically identify and eliminate vulnerabilities within the code for EC applications to improve their security.

In the following sections we describe some teaching techniques that emphasize methods students can use to identify and eliminate vulnerabilities within their EC applications and improve application robustness by integrating many security features within the design process. More importantly, the hands-on approach described here enhances students' understanding of the security content and provides them with solid hands-on experience with web-based EC applications design.

2. CURRICULUM DEVELOPMENT

A semester long e-commerce course taught in the fall of 2004 at a large mid-western university was used for this teaching strategy. It emphasized web-based application

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

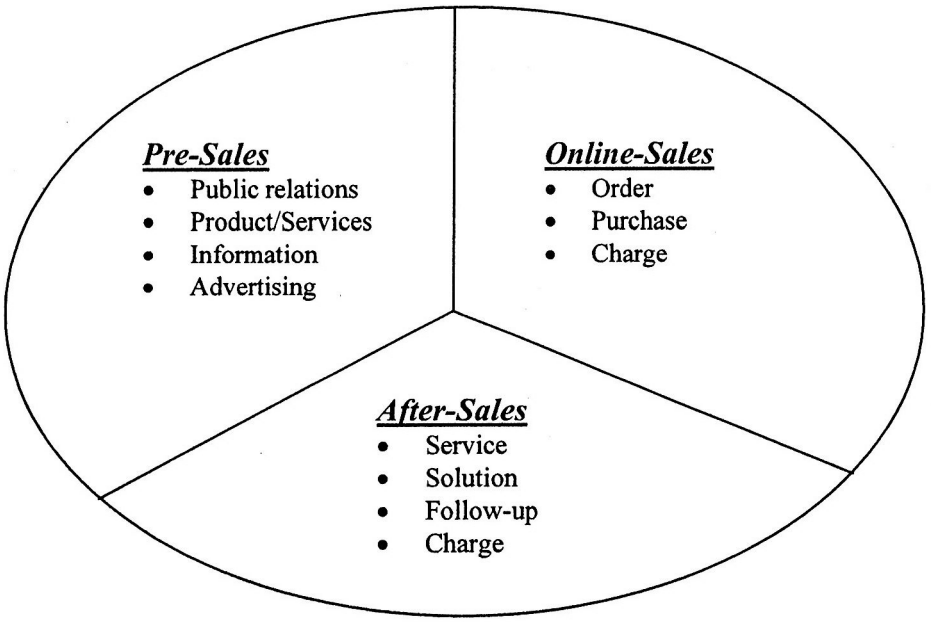


Figure 1: Three Phases of Marketing Activities For EC

design using Microsoft's ASP.Net technology. This allowed the students to explore the EC application development process by creating an electronic shopping mall to sell products or services. The students, who were seniors with an Information Systems major, covered a range of EC students in the course. Students were split into seven groups of three students per group. At the end of the semester, each group of students presented a final project, which incorporated security in each marketing phase of the EC web application, to the entire class. Each group of students chose their own project topic as long as it was different from the topic selected by other groups. Each project had to include an online transaction application for products, services, and/or information. This insured that the project presentation was interesting to the students and it concentrated on EC application design. Moreover, it prompted healthy competitions among different project groups. The students were also required to write a final report on "lessons learned" and "future improvements" for the final project. The final topics were agreed upon on the first class after the middle of the semester thus giving the groups enough time to complete the project by the end of the semester.

3. THE HANDS-ON ENVIRONMENT

A dedicated web server with Windows 2003 Enterprise edition was used for the class and implementation of the students' projects. This server included the following applications: Microsoft Visual Studio.Net (VS.Net) and Microsoft SQL Server 2000. The Information Technology Services (ITS) department on campus was responsible for creating a web folder for each student. The students used the remote desktop connection, available in Windows XP, to access the web server. One of the main advantages of this design environment was that there was no need to install a development topics, including web site design, shopping cart design, input validation, web database integration, order confirmation, and incorporating security features throughout. This allowed the authors the freedom to integrate security features within the design process. There were twenty-one copy of VS.Net on each computer a student might use while developing an EC application. Therefore the only requirement the student needed to work on their application was an Internet connection and the remote desktop application which is included as part of Windows XP. This design environment also insured that each student could only access their own EC application.

4. THE TEACHING APPROACH

The paper explains a unique teaching approach to cover EC application security based on the three phases of EC marketing activities: *pre-sales*, *online-sales*, and *after-sales* as shown in Figure 1.

All EC activities fall into at least one of the three classifications (Liu, Arnett, Capella, and Beatty, 1997). The pre-sales phase includes a company's efforts to attract customers by advertising, public relations, new product or service announcements, and other related activities. Customers' electronic purchasing activities occur in the online-sales phase where orders and charges are placed electronically through the web. Also included in this phase if permitted would be off-line orders and charges, such as those from a mail or telephone medium. The after-sales phase includes customer service, problem resolution, and other issues such as handling product defects and returns, as well as follow-up surveys to maintain and increase customer satisfaction.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation defaultLanguage="vb" debug="true" />

    <!--  CUSTOM ERROR MESSAGES
          Set customErrors mode="On" or "RemoteOnly" to enable custom error messages,
          Add <error> tags for each of the errors you want to handle.

          "On" Always display custom (friendly) messages.
          "Off" Always display detailed ASP.NET error information.
          "RemoteOnly" Display custom (friendly) messages only to users not running
           on the local Web server. This setting is recommended for security purposes,
           that you do not display application detail information to remote clients.
    -->
    <customErrors mode="On" defaultRedirect="error.aspx" >
        <error statusCode="403" redirect="authorizationfailed.aspx"/>
        <error statusCode="404" redirect="notAvailable.aspx"/>
        <error statusCode="500" redirect="error500.aspx?code=500"/>
    </customErrors>
```

Figure 2: The Web.Config File for Error Handling

EC Activities	Security Topics	Hands-on Activities
Pre-Sales Phase	• Document and research security related issues • Handling "unexpected" errors to increase trust	• Examine *xml* technique • Work on the *web.config* file to avoid vulnerabilities
Online-Sales Phase	• Privacy policy and privacy seal protection • Input validation • Data encryption • Secure data connection	• Develop a privacy policy • Establish a secure connection using *https* • Compute a hash value for a credit card number • Exercise validation tools in ASP.Net
After-sales Phase	• Authentication • Authorization	• Examine *stored procedure* technique • Examine *SQL Injection attack* • Work on the *web.config* file to add authentication and authorization components

Table 1: Security Content Breakdown for Three-Phase of Marketing Activities in EC

In this course, the students were asked to examine how to enhance EC application security for each of the three phases of marketing activities. Table 1 shows the security content breakdown and hands-on activities within the course.

4.1 Security in the Pre-Sales Phase

When discussing how to build security in the pre-sales phase of an EC web site, emphasis was given to ways to prevent unexpected errors, thus building robustness into the site to increase customers' trust level and confidence while browsing the site. There is no doubt that errors may occur in any EC application. When creating code in ASP.Net, the students tried to trap errors using TRY-CATCH statement blocks. Although the statement blocks are helpful, it was shown that this technique can not foresee every possible exception. For example, when a customer tries to access a non-existent page from a site developed with ASP.Net, he or she would see a typical error message of "*the resource cannot be found.*" Moreover, if an unexpected error occurs, the error message displayed in a web browser could reveal information which includes the business logic behind the site design. It was explained that this can be quite dangerous since it can give intruders more information to use in trying to stop the application, gain access to sensitive information, and/or execute malicious code.

In the class, the students were asked to examine two different methods in ASP.Net for handling unexpected errors: using a feature called *customerErrors*, which is a section within the *Web.Config* file, and using a subroutine called *Application_Error* within the *Global.asax* file. Since the *Web.Config* is written in XML format, a hands-on lab was given so that the students learned the basics of XML and how to generate an XML file to represent a data structure. Figure 2 presents a lab exercise of using the *customerErrors* section in the *Web.Config* to handle unexpected errors and specified errors such as those defined in HTTP 403, 404, and 500 codes. These hands-on exercises demonstrated to the students that all unexpected errors could be handled in a certain way to increase security and robustness of the pre-sales phase of an EC application.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

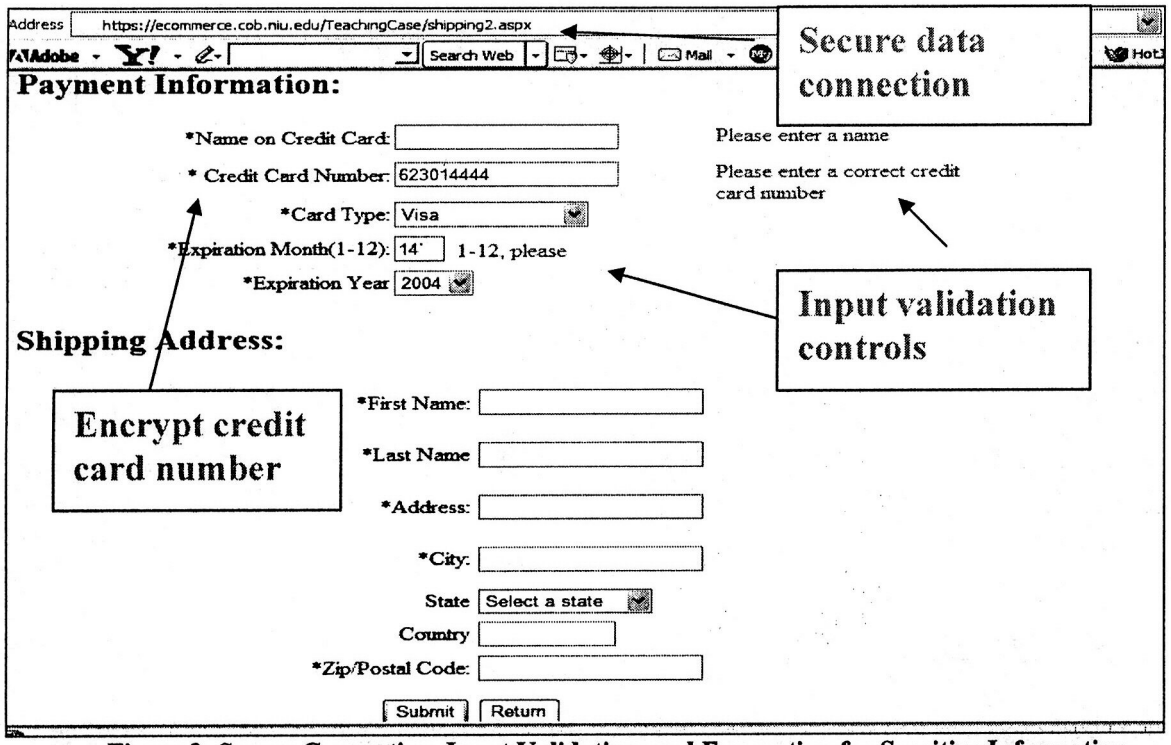


Figure 3: Secure Connection, Input Validation, and Encryption for Sensitive Information

4.2 Security in the Online Sales Phase

The next area to be handled was security within the online sales phase. Research was presented that showed that as the number of businesses using EC applications has increased there has been an increase in the level of concern about consumer privacy. This concern will become even more heightened as more customers engage in EC activities which collect personal and financial information. According to the Federal Trade Commission (FTC), protecting consumers' privacy is an important aspect of ensuring data security in online sales activities (FTC Congress Report, 2000). To show the importance of addressing privacy concerns in an EC application design, the students were asked to research and write their own privacy policies that they felt could ease customers' privacy concerns. The students also explored several seal programs such as TRUSTe (http://www.truste.org/) and BBBOnLine (http://www.bbbonline.org/). They found that the seal programs require their licensees to abide by posted privacy policies and various types of compliance monitoring in order to be allowed to display a seal of trust on their web sites. For example, all privacy seal programs require posting *notice* and *disclosure* of collection and use of personally identifiable information. In addition, websites should give customers *choice* and *consent* over how their information could be used and shared. It is very important to incorporate these privacy dimensions into an EC application design in the online sales phase.

It was shown how to use ASP.Net for input validation, data encryption, and secure data connection within the online sales phase of an EC application. Figure 3 shows a web page used in the course for the secure data connection, data encryption, and input validation hands-on activities.

When customers submit their financial and personal information to a web site, the data is transmitted from a browser to the company's web server. As the data moves through the Internet, it could be intercepted and read by unauthorized persons. The proper solution is to encrypt the data before it is sent through the Internet. One hands-on exercise in the course was to use the MD5 Hash algorithm to encrypt credit card information before it was passed from the browser to the server and then decrypt it before it was stored in a database table. In addition, the students learned how the Secure Socket Layer (SSL) could be used to encrypt sensitive information. Because this course dealt with application design, the authors presented the process to install Certificate Services in a Microsoft Windows 2003 Server, generate a Certificate Request file, issue a Certificate, and then install a server-side Certificate by using Microsoft Internet Information Manager.

4.3 Security in the After-Sales Phase

In the after-sales phase, the course focused on secure access to the data collected from customers for an EC application. Hands-on activities centered on authentication and authorization techniques to allow customers to securely view or update their personal and financial information submitted to a corporate web site. For example, the course explored the SQL Injection attack as shown in Figure 4. In this type of attack the intruders attempt to pass malicious SQL code into an application in an attempt to determine rights, passwords and/or information about the data and the backend database

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

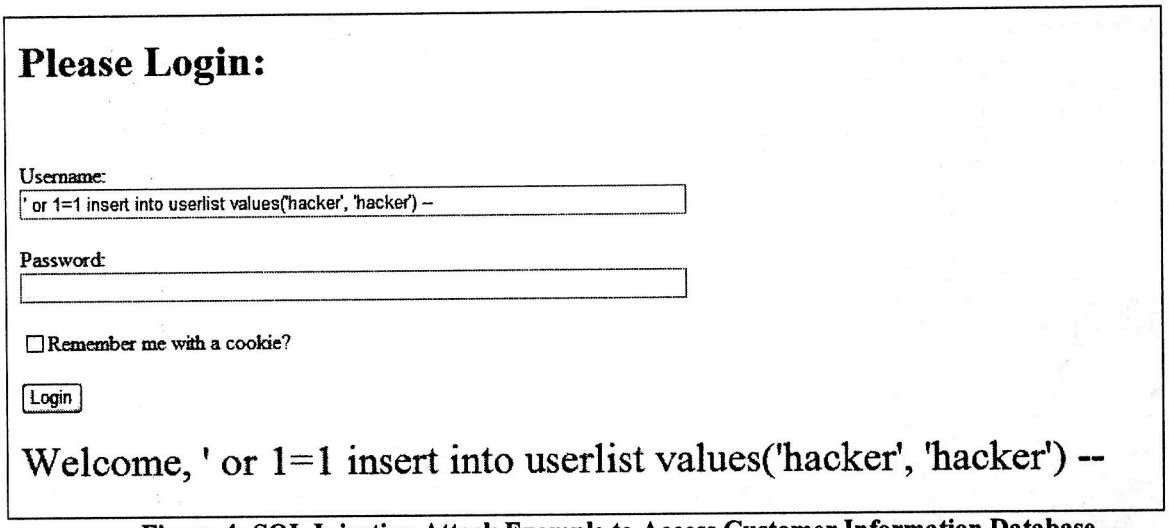


Figure 4: SQL Injection Attack Example to Access Customer Information Database

design. The students learned about using validation controls to constrain certain characters such as " ' " and "--" and implementing SQL stored procedures to avoid this type of attack.

Another technique demonstrated in the course was to use separate *Web.Config* files in subdirectories of an EC application. These *Web.Config* files were used to limit user access to ensure security in the after-sales phase of EC activities. A scenario was developed in which the students had to create a *Member Only* directory within their EC application to serve returning customers in the after-sales phase. The *Member Only* directory had several subdirectories such as *Special Deals* and *Award Services*. Each directory had its own authorization rules declared in the *Web.Config* file residing in that directory folder. Therefore, access was determined by a user's identity which enhanced the security of the EC application.

5. CONCLUSIONS

To evaluate student expectations and reactions, the authors developed a post-course evaluation survey. This evaluation survey was in addition to the normal university course evaluation. Initial results indicated that ALL students rated the hands-on exercises on security through the three phases of EC activities very helpful and applicable to real business situations. The students believed that being "forced" to examine security issues based on *pre-sales*, *online-sales*, and *after-sales* phases was important to helping them with content understanding and classification. The students walked away surprised that there were so many security issues involved in an EC application design. Many were excited that they could protect an application against these security vulnerabilities. Interestingly, several students went to another faculty member and showed him that his online application was vulnerable to a SQL injection attack (one of the examples used in the class). As a result of lessons learned, the following are some suggestions for other faculty incorporating security issues in their EC application design class:

- Each student should be required to sign a letter promising to be a good citizen, by not using the skills and knowledge learned in the class to harm or explore vulnerabilities of web sites. This would be a protection for both faculty who show these techniques and the students who participate in this type of class.
- Prerequisites for students include knowledge of the VB.Net programming language, HTML, database concepts, and a good understanding of networking fundamentals. Students can then learn quickly and cover the topics in more depth if they have the above skills.
- The faculty member should be given permissions to assign web folder configurations. Hands-on exercises in the course often required a group of students to develop and test security issues together. With the current settings, each student could only access his or own folder within the web server. It would be very useful if the students could all have access to a given project folder instead of one of the students having to give his login and password to all members of his group.
- Students should be taught that security is no easy fix. They should continue to document, search solutions, review, and refine security issues in the application design process.

REFERENCES

FTC Report to Congress: Privacy online: fair information practices in the electronic marketplace, http://www.ftc.gov/os/2000/05/index.htm#22.

Hof, R.D. and Hamm, S. (2002), "How e-biz rose, fell, and will rise anew", *BusinessWeek*, May 13, pp. 64-72.

Gartner Group report (2005), "B2B spending to reach $8.5 trillioon."

Liu, C., Arnett, K.P., Capella, L., and Beatty, R.C. (1997), "Web Sites of the Fortune 500: Facing Customers through Home Pages," *Information & Management*, 31(1), pp. 335-345.

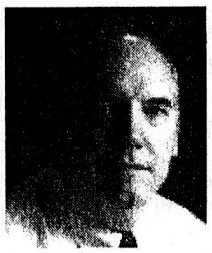
Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Main, A. (2004), "Application Security: building security into the development stage", *Information Systems Security*, 31(7), pp. 51-53.

The 2002 Computer Security Institute (CSI) report, "Cyber crime bleeds U.S. corporations", http://www.gocsi.com/press/20020407.html.

AUTHOR BIOGRAPHIES

Chang Liu received a Ph.D. in Business Administration from Mississippi State University in 1997. Currently, he teaches database and electronic commerce classes at Northern Illinois University. His research works published at *Information & Management, International Journal of Electronic Commerce and Business Media, Journal of Global Information Management, Journal of Internet Research, Journal of Computer Information Systems, Mid-American Journal of Business, International Journal of Mobile Communications, Journal of International Technology and Information Management*, and *Journal of Informatics Education Research*.



Brian Mackie received a Ph.D. in Management Information Systems from the University of Iowa in 1999. Since 2000, he has been a member of the Operations Management and Information Systems Department at Northern Illinois University. His interests are in networking, databases and security. Dr. Mackie is leading a group of researchers in developing secure collaboration techniques within online communities.

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

Information Systems & Computing
Academic Professionals

STATEMENT OF PEER REVIEW INTEGRITY

All papers published in the Journal of Information Systems Education have undergone rigorous peer review. This includes an initial editor screening and double-blind refereeing by three or more expert referees.

Copyright ©2006 by the Information Systems & Computing Academic Professionals, Inc. (ISCAP). Permission to make digital or hard copies of all or part of this journal for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial use. All copies must bear this notice and full citation. Permission from the Editor is required to post to servers, redistribute to lists, or utilize in a for-profit or commercial use. Permission requests should be sent to the Editor-in-Chief, Journal of Information Systems Education, editor@jise.org.

ISSN 1055-3096