SECURE ONLINE AUCTION WEBSITE

Subject: Network and Computer Security
Campus: Alameda
Group: A28

73216 Nuno Guilherme de Sousa Lobato da Graça

73422 Kévin David Esteves Santos

75657 Paulo Jorge Louseiro Gouveia

Abstract

Auction sites are a useful tool to promote the exchange of goods directly exploiting the demand and request needs of dynamic markets. The goal of this work is to create an online auction store, providing sellers and buyers a reliable website with all the mechanisms for the secure transactions of products and goods.
The site is to be secure against threats like XSS, SQL injection, impersonation and other attacks.

Problem

From the topics available our group chose to tackle the development of a secure auction website.

Such websites often deal with real money and goods, and as such, are subject to various attacks aiming to steal and/or destroy the money, the goods or even the information being shared. Our objective is to protect our website against these attacks by protecting the website infrastructures, as well as protecting the users against possible vulnerabilities and exploitations. Possible exploitations we have to protect against include Cross-site scripting, SQL Injection and others.

Because it is impossible to protect everything in a system, a model with the assumptions made about its environment are required. Ours are the following:

- Social engineering – it is assumed the user is aware of problems such as phishing and takes the necessary precautions to prevent information theft;
- Updated browser security – it is assumed the user is using the latest iteration of his/her browser of choice;
- Secure database – it is assumed the communication to the database is secure and that the database itself is also secure;
- Network problems – any problems with the communication between clients and the server are not ours to deal with, such problem are the responsibility of respective network links and nodes.

Requirements

Given the system vulnerabilities, the website will have to prevent several exploits:

- Code Injection – a malicious user may, for example, use text input fields like password boxes or comments to inject malicious code into the server. To prevent such exploits input sanitization is necessary;

- Packet capture – we also have to assume that all data travelling between the client and the server may be captured. As such, all sensitive information must be encrypted to ensure that only the client and/or the server can read that information;

- User impersonation – a malicious user can pretend to be another user. To prevent this a client authentication system must be implemented to assure that a user account is accessible only to the user.

Proposed Solution

Given the website and database reliant system structure, the proposed solution shall obviously cover the main vulnerabilities which are SQL injection and XSS, both mostly preventable by sanitization of the input and output in our application. For the storing of the login passwords a hash of them will be stored in their place.

A secure connection between the client and the server is still required and for that we will use SLL. As for the distribution of the respective certificates, a first implementation will be using self-signed certificates which will be improved in the advanced stage by the use of a certificate authority.

The project will be divided into three phases: basic, intermediate and advanced. This phases will respectively increment the level of security guaranteed upon the previous.

The specifications of each development stage are as follows:

Basic *November 10th 2016*	– Basic web interface with login – Webserver running *PHP* with input sanitization implemented – *MySQL* configured with the login tables and hashing of passwords
Intermediate *November 24th 2016*	– Fully defined website page tree – Webserver running *SSL* and certificates – Basic XSS prevention – Implementation of all required SQL tables
Advanced *December 9th 2016*	– Fully functional website – Comprehensive XSS prevention – Registration of certificates with a CA (that we might need to implement a simplified version of to simulate the real thing) – Optimized *MySQL* queries and tables

Table 1 – *Completion date and specifications of each development stage*

Tools

On the client's side, there will be a web application running industry standard components. The communication with the server will be using SSL which uses *asymmetric keys* and will work with *https* to assure a confidential channel between browser and server. The interface will be using *html*, *CSS* and *JavaScript*.

Although *Java* is considered more secure and reliable than *PHP*, we will use the latter due to its simplicity to both develop and setup. This will also allow us to use the website hosting service maintained and made available to students by DSI.

For the database we chose *MySQL* because of the knowledge we already have from previous classes and it still being a very widely accepted industry standard which assures proper testing. This will allow us to, once again, make use of the *MySQL* services provided by the DSI.

Work Plan

week	73216	75657
October 27 – November 2	- basic web interface	- *MySQL* login tables - *PHP* webserver
November 3 – November 9 <*basic*>	- input sanitization - login implementation	- password hash
November 10 – November 16	- product submission - simple XSS prevention - *MySQL* tables	- bidding - *MySQL* tables
November 17 – November 23 <*intermediate*>	- *SSL* connection	- certificates for server - *https* for login
November 24 – November 30	- product search - finalize XSS prevention	- finalize XSS prevention
December 1 – December 7 <*advanced*>	- interface touchups	- interface touchups - optimization of queries and tables
December 8 – December 10 <*final report*>	- write final report	- write final report

Table 2 – *Functionality to be implemented per week. Weeks are counted from Thursdays to Wednesdays to have the laboratory class in the beginning of the work cycle.*