CPS353: Internet Programming
Homework Assignment 7: Web Security
Due Wednesday, November 20 by the start of class

This assignment gives you the chance to learn about some common web security vulnerabilities by exploiting them against an unprotected web application. You will use the OWASP WebGoat project to gain first-hand experience of website (in)security.

DISCLAIMER: This assignment neither encourages nor condones the exploitation and hacking of real-world websites. Such practices are Not a Good Idea ™ as they are immoral, unethical, illegal, and can be prosecuted under criminal law. The sole purpose of the exercises below is to give you insight into how attackers think and work so that you can write applications that better defend against them. **Do not use these or any other web security exploits on actual production websites that you do not own!**

Setting up WebGoat

To begin with, you need to download and install the WebGoat application on your personal desktop or laptop computer. You can do this as follows:

1. Download the WebGoat archive from the Homework 7 Resources area on the ips.cs.gordon.edu site.
2. Detailed installation instructions are available for both Windows and Unix-based systems (including Macs). On OSX or Linux, run the following commands from a terminal window to unpack and launch WebGoat on port 8080 of your local machine.

   ```
   unzip WebGoat-OWASP_Standard-5.2.zip
   cd WebGoat-5.2
   sh webgoat.sh start8080
   ```

   (Similar commands involving the webgoat.bat script can be executed to launch WebGoat on Windows. See the instructions referenced above for details.)
3. Browse to the WebGoat homepage at http://localhoast:8080/WebGoat/attack.
4. Login as the guest user (with the credentials guest/guest), click the start button and you're ready to go.

WebGoat Lessons

A collection of WebGoat lesson categories and plans is given in a multi-tiered list on the left side of the page. You can view the first page of a lesson by clicking on its link. For this assignment, you must **complete the following 10 WebGoat Lessons:**

1. **Access Control Flaws – Bypass a Path Based Access Control Scheme**
2. **Authentication Flaws – Forgot Password**
3. **Code Quality – Discover Clues in the HTML**
4. **Cross-Site Scripting – Stored XSS Attacks**
5. **Cross-Site Scripting – Cross Site Request Forgery (CSRF)**
6. **Cross-Site Scripting – Reflected XSS Attacks**
7. **Injection Flaws – Blind SQL Injection**
8. **Injection Flaws – Numeric SQL Injection**
9. **Injection Flaws – String SQL Injection**
10. **Parameter Tampering – Exploit Hidden Fields**

You may also complete up to 10 additional lessons for extra credit. Each successfully completed lesson beyond those listed above is worth 5 bonus points.

Tools and Help

WebGoat is equipped with several tools to help you move through its lessons. The navigation controls at the top of each page include several utilities that you will find useful.

- **Hints** provides you with one or more hints on the current lesson. Each hint you view is recorded on your Report Card page (see below).
- **Show Params** displays the parameters for the last HTTP request submitted to WebGoat.
- **Show Cookies** displays the cookies for the previous HTTP request to and response from WebGoat.
- **Lesson** Plan redisplays the instructions for the lesson in a screen overlay (i.e. "light box").
- **Show Java** pops up another browser window containing the Java source code associated with the current lesson.
- **Solution** opens a new window containing a written synopsis of the lesson and its solution.
- **Solution Videos** links to a set of screencasts providing walkthroughs for various WebGoat lesson.
- **Restart this Lesson** returns you to the first page of the current lesson.

You are free to make use of any or all of these resources as you work through the lessons. However, you should first attempt to solve each lesson on your own prior to viewing hints or solutions for that lesson. Doing so will provide you with the best experience and understanding of the web security concepts illustrated in WebGoat.

Modifying HTTP Requests

Several WebGoat lessons require you to customize various properties of HTTP requests in order to complete them. The easiest way to do this is to use a tool that intercepts these requests after your browser submits them and allows you to make modifications before sending them onto the WebGoat server. A browser extension such as Firefox's Tamper Data add-on works well for this task. (WebGoat recommends the OWASP WebScarab tool, but this can be unwieldy to install and use.)

Once you install the Tamper Data add-on to Firefox, you can activate it via the "Tools – Tamper Data" menu item and then clicking the "Start Tamper" option in the window that appears. At this point, Tamper Data should intercept each HTTP request from your browser and ask you whether you want to submit it, abort it, or tamper with it. Tampering with a request allows you to modify its headers and parameters before forwarding it onto the server.

You may also find the Firebug Firefox extension (or its cousins for other browsers) useful for this assignment. Your browser's "View Source" command might also come in handy.

Report Card Page

Once you have completed the WebGoat lessons above, proceed to the Report Card page (under "Admin Functions – Report Card" in the lessons menu). This page displays details on the lessons you have worked on with completed lessons having a green background.

Save a copy of your Report Card page (HTML only) and submit it to the instructor via email. NO CREDIT WILL BE GIVEN WITHOUT THIS REPORT CARD PAGE.

Turn In…

Submit the following materials to the instructor via email.

- **Your WebGoat Report Card HTML page.**