System Security Group Project #2

(10% of class grade, due 2/23 at 11:59pm)

Go through interactive broken web application tutorials. Explain what you learn! This project is worth 10% of your class grade.

What you will need:

You will need access to the class vSphere environment. You will be using the following virtual machines:

- Kali Linux with WebGoat pre-installed

Directions:

Setup

Follow the project videos for accessing and running the WebGoat and the Kali Linux virtual machines. If you need a reminder of how to access our cloud environment, please see the user manual.

Start the kali virtual machine image (if not already started). Start and log into your Kali Linux virtual machine (username kali, password kali).

Open up the terminal and run the following command to start WebGoat from your kali home directory:

```
java -jar webgoat-2025.3.jar
```

Once WebGoat starts, open up Firefox in the Kali Linux machine. Put the following URL in your web browser:

http://127.0.0.1:8000/WebGoat/

Create a username and password for yourself, and **be sure to put it in your group documentation.** You don't want to lose your work if you forget your password. I can't reset it for you…

What is WebGoat? (From the OWASP website)

"WebGoat is a deliberately insecure web application maintained by OWASP designed to teach web application security lessons. In each lesson, users must demonstrate their understanding of a security issue by exploiting a real vulnerability in the WebGoat applications. For example, in one of the lessons the user must use SQL injection to steal fake credit card numbers. The application aims to provide a realistic teaching environment, providing users with hints and code to further explain the lesson." – WebGoat wiki at https://owasp.org/www-project-webgoat/ .

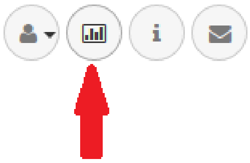
On the left, click on the Introduction link. It will expand. Click on and read the Webgoat lesson. WebWolf is also installed. Please use your WebGoat username and password to log into WebWolf while completing that lesson.

Lessons in WebGoat

Start with the Introduction and General lessons. Everyone must complete all of the following assignments within those lessons. We will do most of these in class together.

- **Webgoat** and **Webwolf**
- **HTTP Basics** (You probably want to do HTTP Proxies first, then this lesson second)
- **HTTP Proxies** (Note, you may have to do this lesson first to complete the HTTP Basics lesson. ZAP is already installed on your Kali Linux machine, so follow the directions to hook it up to your web browser.)
- **Developer Tools**

You'll notice that there are ***lessons*** and ***assignments*** in WebGoat. Lessons are the links on the left-hand menu. Assignments are usually numbered in red and require you to solve the puzzle. There are 90 total assignments in this version of WebGoat. You can see how many lessons are completed by clicking on the report card icon in the upper right:



At the top of the report card will be a summary of what you have completed. However, in previous versions of WebGoat, the scorecard was sometimes buggy. To prove that you have completed a lesson, you'll want to take screenshots of the lesson completed message.

Grading

The scorecard is broken up into lessons and individual assignments (red numbers) within the lessons. Not counting the 7 assignments in Webgoat, Webwolf, HTTP Basics, HTTP Proxies, and Developer Tools, each group needs to complete the following number of assignments:

For groups of 3:

- Completing 54 of the assignments is a middle A (95%)
- Completing 45 of the assignments is a middle B (85%)
- Completing 36 of the assignments is a middle C (75%)

For groups of 2:

- Completing 36 of the assignments is a middle A (95%)
- Completing 30 of the assignments is a middle B (85%)
- Completing 24 of the assignments is a middle C (75%)

Note that only unique assignments completed count towards the total. So, if Alice finishes the CIA Triad assignment and Bob completes the same assignment, that is only 1 unique assignment completed.

There are a few assignments that are definitely easier than other assignments. Some assignments don't make much sense for this class (such as the assignment embedded in the lesson "Writing new lesson").

If you are stuck

You have a couple of options. Sometimes there is a "hints" button you can click at the top of the assignment in webgoat for hints. Other times, you may want to check out the OWASP blog for tips: https://github.com/WebGoat/WebGoat/wiki/Main-Exploits (Note – the blog is not complete.)

There are also walkthroughs on youtube. Make sure the version of webgoat looks close to your own. (Webgoat has been around for a long time, and older versions were very different.)

The professor will give advice on particular lessons you may want to avoid and other lesson-specific hints.

The deliverable

I am expecting 2 parts to the deliverable.

1. **Paragraphs:** For each ***assignment you complete***, take a screenshot (pic) of successfully completing the assignment. Then type up at least one paragraph on what you learned. Don't just copy and paste text from the lesson plans (I will be checking). Also, if applicable, let me know what was easy, what was hard, and what you couldn't get to work quite right. We will use class periods as work periods to ask questions about things you can't quite get and get advice from other teams. (All advice from other teams should get a citation/shout out in your paragraph.)

2. **Assignment Completion Table:** Create a table listing the lessons completed and who completed them. At the bottom of the table, state the total number of lessons completed by each group member. (Note: It might be tempting to copy/paste the webgoat report card here. Unfortunately, the report card can be buggy in some lessons and not indicate things were solved when they were actually solved in a minority of cases.)

Your deliverable should be in your team google folder. Create a new google doc called "Project 2" directly under your main team folder and place your paragraphs and score cards in there.

In addition, you will be required to fill out an internal peer review before you receive a grade. (You cannot skip this part.)

Other Things

- This is the newest version of Webgoat being tested on you. Expect some bugs.
- I'm expecting things to sometimes be difficult, even with the hints and solution videos available. Sometimes the difficulty will be in getting things set up. Other times it might be in figuring out what you should be watching or modifying. Sometimes a new term will come up that you need to look up.
- If you don't know what something is, Google it to find out!
- I expect you to have to use Google to figure out some features of ZAP/Burp, wireshark, helpful encoding/decoding tools/websites, etc.

- I know other walkthroughs exist on the Internet. It's ok to learn from those. Please cite your source in the paragraph by providing the link to the help. (There is no penalty for this.)
- Make sure you are taking the time to learn. If your paragraphs don't convince me you really understand what's going on, I will award less points.
- Doing some things in the "Challenges" lesson may earn you extra credit, but be careful! (E.g. some students have accidentally reset the WebGoat VM in the past by doing some things, so take a working VM snapshot of the machine before you play around in here.)
- I don't expect you to become masters of this material. Instead, the point of this project is to expose you to classes of web problems at an introductory level by doing things. This knowledge will be useful later in the class.