CSE 543: Computer Security

Fall 2021
Project 2: Web Security
Due: 11:59 pm (eastern time), October 25, 2021

September 30, 2021

1 Introduction

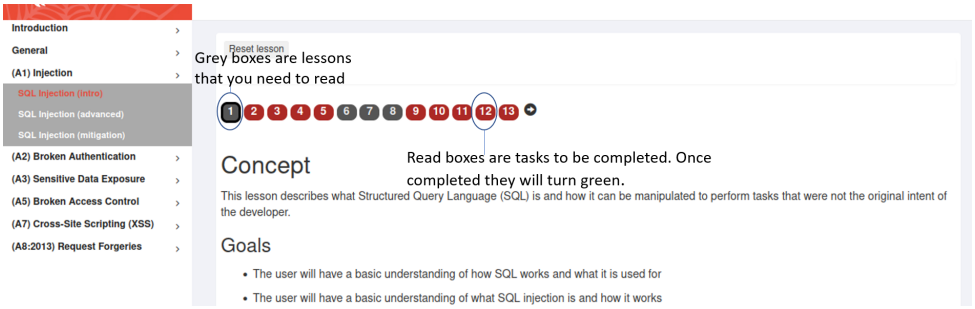
In this project, you will exploit a series of web vulnerabilities discussed in class. You will be given WebGoat[1], an insecure web application designed for educational purposes. **Note: This project is for educational purposes only, and should never be used outside of this class**. We recommend disconnecting the VM from the internet when using WebGoat. This will be a more practical and hands-on project. You will be using some tools that professional security engineers use (like Burp [2]). There is not much coding involved but that does not mean this would be an easy project. We highly recommend you start early on this project as it has more components and a longer report that you will have to submit.

If you are having trouble setting or running it up, please contact TA as soon as possible.

2 Getting Started

WebGoat is already installed on the VM that can be downloaded here in tar.gz or here in zip. Downlaod size is around 7 GB. Username for the VM is **cse** and password is **esc**. You would need around 14 GB of free space for the VM (you can delete the VM from project1). If you want to run it on Apple's latest M1 chip based machine, contact the TA

You can run the WebGoat by going into WebGoat folder on the desktop and executing: **sh ./start.sh**. Then you can go to Firefox with URL *http://localhost:8080/WebGoat/* to start. Your username for WebGoat is **student543** and the password is **student543**. Once you start WebGoat it will look like this:



3 Prerequisite

Before attempting this project, it is advisable to read and understand A1, A2, A3, A7, and A8 from https://owasp.org/www-project-top-ten/. Although not required, it is **highly recommended** that you finish tasks from the **General** category. They introduce you to some basic tools and concepts that will help you with this project.

Most of the tools that you might need are already installed on your VM: Burp community edition [2], FoxyProxy [3] extension for Firefox and WebWolf (comes with WebGoat see Introduction–¿WebWolf on left side menu when you log in to WebGoat). Developer's tool in Firefox browser will be of great help. If you want to use any other tool you are welcome to use the latest version of it, please let the TA know what tools you will be using. Wireshark[4] might be one of those tools. You can install it with *sudo apt-get install wireshark* **if needed**.

4 Exercise Tasks

WebGoat has the basics of each vulnerability, related concepts, and hints for each of the tasks. You can read the description for each on the respective page. For this project, you will have to solve the following tasks:

1. Injection
    - SQL injection intro
    - SQL injection advanced
    - SQL injection mitigation

2. Broken Authentication
    - Authentication Bypasses
    - Secure Passwords
3. Sensitive Data Exposure
    - Insecure Login
4. XSS
    - Cross Site Scripting
5. Request Forgeries
    - Cross-Site Request Forgeries
    - Client-Site Request Forgeries (**Optional**)

5 Deliverables

It is not necessary to submit any code for this exercise. Please submit a report in pdf format containing the following:

1. Your Report card
2. Screenshot of each completed task with (a) at least one paragraph on what you learned, (b)Step by step instruction to solve each task, (c) if you could not solve it, what you tried, what worked, and what did not.
3. If you exploited a vulnerability what can be the mitigation? How would the mitigation work?
4. Some tasks have associated questions with them, Answers those questions in pdf
5. Any outside resources you referred to. This can be a blog, StackOverflow, research paper, classmate, etc.

6 Grading

The assignment is worth 100 points (+10 bonus) total broken down as follows.
Injection 32%, Broken Authentication 10%, Sensitive Data Exposure 5%, XSS 25%, Request forgeries: Cross-site 28%, Client-Side 10% (optional bonus).

You will be evaluated on:

1. Completion of the tasks/sub-tasks
2. Explanation of tasks and clarity of steps in the solution
3. Answer to questions asked in the tasks.
4. Completeness of the resources used

Note: You might be asked to demonstrate the solution(s) of task(s) to award the final grade for the project.

7 Misc

- Make sure you are taking the time to learn. Your report should reflect your understanding.
- There might be more than one way of doing a task.
- Take advantage of Piazza.
- We expect you to use the Internet to figure out some features of burp [2], ZAP, wireshark[4], helpful encoding/decoding tools/websites, HTML basics, SQL, etc. If needed. Don't forget to cite the resource you use.
- We will check all submissions for plagiarism.
- We have noticed sometimes the WebGoat freezes or stops responding when you send some malformed requests to it or if you did not set up proxy in a certain way. Restarting WebGoat and sometimes VM fixes the issue.
- Since there might be various scenarios, components, tools, and variables associated with some tasks, don't expect the TA to have all answers right away.
- Some tasks are intentionally difficult. The difficult part might be setting up a tool or understanding what to look for or some term that you have not heard before. We have tried to attach resources to Webgoat pages for your help. If not, use the Internet/papers/TA's OH.
- If you find a particular resource helpful for understanding the tool or concept, you can share that on piazza. (But no solutions).

References

[1] "Owasp webgoat - learn the hack - stop the attack." [Online]. Available: https://owasp.org/www-project-webgoat/

[2] "Burp suite - application security testing software." [Online]. Available: https://portswigger.net/burp

[3] "Foxyproxy:an advanced proxy management tool for firefox," Apr 2006. [Online]. Available: https://addons.mozilla.org/en-US/firefox/addon/foxyproxy-standard/

[4] "Wireshark packet capture · go deep." [Online]. Available: https://www.wireshark.org/