Bruno Sousa, bmsousa@dei.uc.pt
2024/2025

Worksheet #1 – Web Application Vulnerabilities

Practical Lectures #1, #2

OBJECTIVE:

Understand the process of discovering, analyzing, exploiting, and mitigating common vulnerabilities in web applications. The process can be automated through the utilization of tools that have several steps pre-prepared for use in vulnerability discovery and exploitation.

CONTEXT:

WebGoat[1] is an OWASP project which comprises a deliberately insecure web application, designed to teach web application security lessons. In each lesson, users must demonstrate their understanding of security issues by exploiting a real vulnerability in the WebGoat application. This project uses a hands-on approach so students and professionals can understand the most common vulnerabilities by finding and exploiting them.

SETUP:

WebGoat can be installed using Docker **OR** the standalone version which is available for download on the website.

Docker

First install Docker following the steps available at: https://dockr.ly/3DuDnf4. After configuring it, run the following command to start the container and the image will automatically download, if not present.

```
$ docker run -p 8080:8080 -p 9090:9090 -p 80:8888 -e TZ=Europe/Amsterdam
webgoat/goatandwolf:latest
```

Standalone

Download the latest release of WebGoat from (https://github.com/WebGoat/WebGoat/releases) and run the following command.
Note: for this approach to work Java[2] is required to be installed in the system.

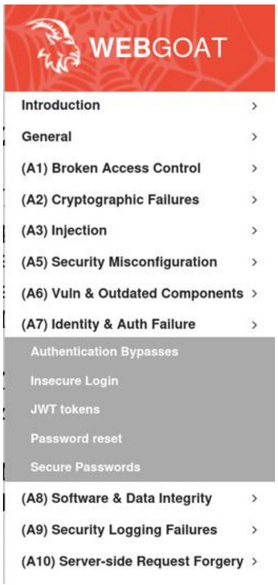
```
$ java -jar <path-to-webgoat-server-jar.jar>
```

After the setup process is complete, WebGoat should be available at http://localhost:8080/WebGoat.

[1] https://owasp.org/www-project-webgoat
[2] https://java.com/en/download/help/download_options.html

EXERCISES:

1. Exploring Web Application Vulnerabilities


WEBGOAT
Introduction
General
(A1) Broken Access Control
(A2) Cryptographic Failures
(A3) Injection
(A5) Security Misconfiguration
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
Authentication Bypasses
Insecure Login
JWT tokens
Password reset
Secure Passwords
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery

Explore the application manually. Many web applications have SQL injection vulnerabilities that allow users to provide SQL code through form fields and execute it in the backend database. Understand how to identify vulnerable fields exposed to users and how the vulnerabilities can be explored.

Note: Focus on lessons A3: Injections, A7: Identity & Auth Failure.

2. Exploring Web Application Vulnerabilities using Tools

The identification of some vulnerabilities is easily done manually, but the use of tools eases the process of exploring the requests formed and sent by the client to the server. Use proxy servers and/or web application vulnerability scanners to assist in the identification of vulnerable fields and their exploration, to automate repetitive processes. Use tools such as OWASP ZAP, Burp, Nikto, WebWolf, or others that you deem fit.

The General tab of WebGoat lessons explains how to set up ZAP to use it as a proxy for the application.

Note: Focus on lessons A3: Injection (Path Traversal), and A7: Identity & Auth Failure (Insecure Login).

3. Automated Discovery and Exploitation of Vulnerabilities

As discovering and exploiting vulnerabilities is a process with well-defined steps, it can be automated using tools. Several tools find vulnerabilities and explore them automatically. As you are already familiar with some vulnerabilities you can try to explore them using tools to understand how it eases the process but still requires you to understand what is occurring under the hood. Use tools such as SQLMap or others.

REPORT (WHAT TO DELIVER):

The report must include the following information:

1. Option chosen for installation (docker or standalone)
2. Demonstration of the A3 Injection (Path Traversal) exploitation
3. Demonstration of the A7 Identity & Auth Failure (Insecure login) exploitation
4. Demonstration of discovery (and possible exploitation) using automated tools

READINGS:

- OWASP, *“Top 10 Web Application Security Risks”* 2017. Available at: https://owasp.org/www-project-top-ten
- Weidman, Georgia. *“Penetration testing: a hands-on introduction to hacking”.* No Starch Press, 2014. [Chapter 14: Web Application Testing]