Chapter 9: Web Application Hardening

In this laboratory exercise, the student will be introduced to common web application vulnerabilities and will mitigate against the vulnerabilities.

9.1 Laboratory Exercise

Web Application Hardening

9.1.1 Specifications

The variant of Linux being used for this laboratory is Ubuntu 16.04.6, an older version of the Ubuntu operating system. This machine has been configured with a vulnerable web application.

9.1.2 Learning Objectives

- Understand web applications
- Familiarize with web server applications such as Nginx
- Implement web applications using HTML, CSS, JavaScript, PHP, and MySQL
- Acknowledge common security vulnerabilities in web applications
- Mitigate against common security vulnerabilities in web applications

9.1.3 Mapping to NIST Nice Framework

This laboratory exercise is intended to increase the student's awareness of web application vulnerabilities. Additionally, the student will learn how to mitigate against these vulnerabilities. The student will have to perform small portions of programming in order to harden the vulnerable application. The student should have improved familiarity with HTML, MySQL, PHP, and Nginx upon successful completion of this exercise. This laboratory exercise maps to the following KSAs from the NIST NICE Framework:

- Cybersecurity and Privacy Principles (K0004)
- Cyber Threats and Vulnerabilities (K0005)
- Application Vulnerabilities (K0006)
- Data Administration (K0020)
- Database Management Systems (K0023)
- Programming Language Structures and Logic (K0068)
- Query Languages (K0069)
- Application Security Threats and Vulnerabilities (K0070)
- Secure Coding Techniques (K0140)
- Database Access Applicaiton Programming (K0197)
- Test and Evaluation Processes (K0250)
- Hacking Methodologies (K0310)
- Database Theory (K0420)
- Conducting Queries (S0013)
- Conducting Test Events (S0015)

- Designing Countermeasures to Identified Security Risks (S0022)
- Generate Queries (S0037)
- Writing Code (S0060)
- Recognize Types of Vulnerabilities (S0078)
- Applying Secure Coding Techniques (S0172)
- Develop Secure Software (A0047)
- Maintain Databases (A0176)

9.1.4 Necessary Background and Expected Completion Time

This laboratory exercise can be completed by students with varying background and experience. The following categories should help identify approximately how much time (in minutes) will be necessary to complete the laboratory exercise, for a student meeting the criteria for the respective experience level.

- Beginner: A student in this category has little to no experience with web application concepts or web development. This student should follow the walkthrough guide as well have instructor guidance.
- Intermediate: A student in this category is familiar with the concepts surrounding web applications and web development and may have some experience with web development. This student has little to no experience in web application hardening and security. This student should reference the walkthrough guide as needed.
- Advanced: A student in this category is experienced with web development and is familiar with web application hardening and security concepts. This student may need to reference the walkthrough to see how the vulnerable web application was implemented.

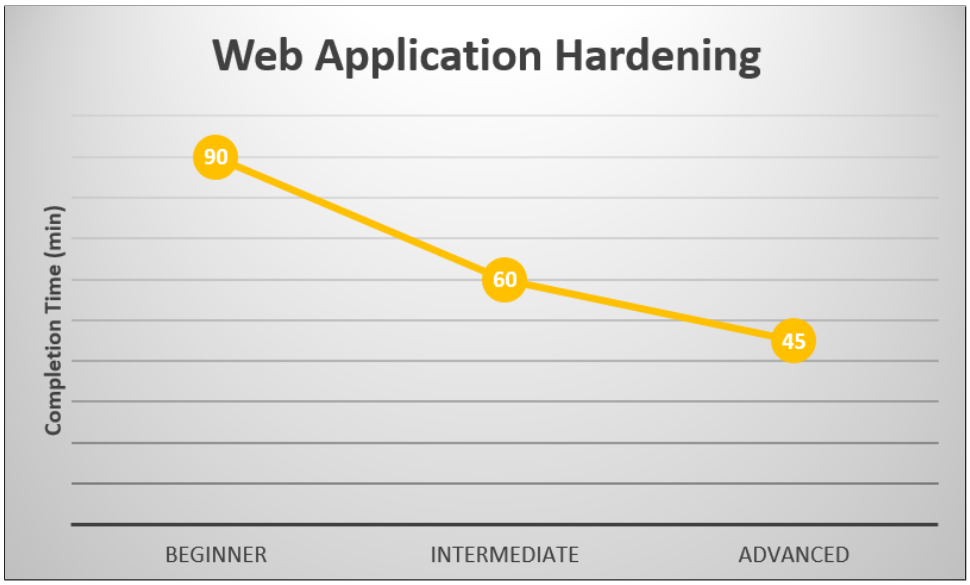

Figure 9.1: *Web Application Hardening Laboratory Exercise Expected Completion Time (min)*

9.1.5 Configuration and Setup

The machine used in this laboratory exercise is an installation of Ubuntu 16.04. The machine reaches out to a GitHub repo to clone files necessary to complete the exercises. Additionally, Nginx, PHP, and MySQL are installed. The initialization script creates and populates the MySQL databases and tables. Additionally, the script creates the vulnerable web application. The machine is configured using the initialization script, listing **9.1**.

Listing 9.1: *hw-initializationscript.sh*

```
#!/bin/bash

sudo dpkg --configure -a


#grab the github repo
sudo apt-get install git
sudo rm -r CYOTEE
sudo git clone https://github.com/CenterForSecureAndDependableSystems/CYOTEE.git

#install nginx
sudo apt-get install nginx -y

#install php
sudo apt-get install php-fpm php-mysql -y

#install MySQL
sudo apt-get install mysql-server -y

sudo cp CYOTEE/CYOTEE_Code/VulnerableCode/* /var/www/html/

sudo mv /var/www/html/default /etc/nginx/sites-available/default

sudo mysql -u root -e "DROP DATABASE test";
sudo mysql -u root -e "CREATE DATABASE test";

sudo mysql -u root -e "CREATE USER 'newuser'@'localhost' IDENTIFIED BY 'newpass'"
    ;
sudo mysql -u root -e "GRANT ALL PRIVILEGES ON test.* TO 'newuser'@'localhost'";

sudo mysql -u root test < CYOTEE/CYOTEE_Code/SQL/test.sql

sudo service nginx restart
```

9.1.6 Vulnerability Overview

1. SQL Injection
2. User Access not Controlled
3. Lack of Input Validation

9.1.7 Challenges

1. Implement and run the vulnerable web application
2. Identify where in the vulnerabilities exist
    - SQL Injection
    - Lack of Least Privilege Implementation

- Lack of Input Validation

3. Mitigate against vulnerabilities

    - Mitigate against SQL Injection
    - Implement a form of least privilege so that only necessary MySQL users have access to their respective pages
    - Perform input validation on the date field to ensure that a valid date is entered