Technology Application Project

2024-HD06-COS80029

USER MANUAL: Penetration Testing Tool

by

Jenish Gautam (Team 7)

SID: 104223445
S.Name: Jenish Gautam
Lecturer: Dr. Ayesha Binte Ashfaq

Table of Contents

1. Introduction

Purpose of the Manual: This manual provides a comprehensive guide to setting up a contained environment for penetration testing, identifying and testing vulnerabilities in a web application, and using a custom-built tool to automate the testing process.

Overview of the Project: This project involved testing a web application called Juice-Shop for security vulnerabilities. The testing was conducted in a contained environment using Kali Linux and Burp Suite. Five key vulnerabilities were identified and tested, and a custom tool was created to automate the testing process.

2. Setting Up the Contained Environment

Step 1: Install Kali Linux

To conduct penetration testing, you need to set up a secure and reliable environment. Kali Linux is a preferred choice for such tasks due to its extensive suite of security tools.

Download Kali Linux:

Create a Bootable USB :

Boot from the USB :

Install Kali Linux:

Post-Installation Setup:

- Update your system and install any additional tools needed for penetration testing.

Step 2: Clone the Web Application (Juice-Shop)

Juice-Shop is an intentionally vulnerable web application that will be used for testing.

Clone the Repository:

- Open a terminal in Kali Linux and run the following command:

```
git clone https://github.com/bkimminich/juice-shop.git
```

Install Dependencies:

- Navigate to the cloned directory:
  ```
  git clone https://github.com/bkimminich/juice-shop.git
  ```

Install Dependencies:

- Navigate to the cloned directory:
  ```
  npm install
  ```

Run the Application:

- Start the application using:

  ```
  npm start
  ```

- Access Juice-Shop in your browser at http://localhost:3000.

Step 3: Set Up Burp Suite

Burp Suite is a powerful tool for web application security testing.

Download Burp Suite:

- Download Burp Suite Community Edition

Install and Configure Burp Suite:

- Install Burp Suite on Kali Linux

3. Vulnerability Scenarios and Testing

This section details the five key vulnerabilities identified in Juice-Shop, along with step-by-step instructions for testing each one.

Vulnerability 1: Cross-Site Scripting (XSS)

- Scenario 1: Changing Product Description
    1. Navigate to a product page in Juice-Shop.
    2. Modify the product description by injecting a script tag, such as <script>alert('XSS')</script>.
    3. Save the changes and reload the page to see the XSS alert pop up.

Outcome: XSS allows attackers to inject malicious scripts into web pages viewed by other users.

Vulnerability 2: Improper Input Validation

- Scenario 1: Admin Registration
    1. Use Burp Suite to intercept the registration request.
    2. Modify the intercepted data to bypass input validation, such as changing the user role to admin.
    3. Submit the modified request to see if the application accepts the forged data.

- Scenario 2: Forged Rating
    1. Submit a rating through the application.
    2. Intercept the rating request with Burp Suite and manipulate the input data.
    3. Submit the altered request and observe whether the rating is accepted.

Outcome: Improper input validation can lead to unauthorized actions, such as elevating user privileges or forging ratings.

Vulnerability 3: Sensitive Data Exposure

- Scenario 1: Confidential Document Access
    1. Use Burp Suite to identify endpoints that may expose confidential documents.
    2. Access these endpoints directly to retrieve sensitive information.
- Scenario 2: Forgotten Sales Backup
    1. Search for backup files left on the server, such as .bak or .sql files.
    2. Download and analyze these files to extract sensitive data.

Outcome: Sensitive data exposure occurs when an application fails to adequately protect sensitive information.

Vulnerability 4: Broken Access Control

- Scenario 1: Change Password of a User
    1. Attempt to change another user’s password by intercepting and modifying the password change request.
    2. Submit the request and verify if the password change was successful.
- Scenario 2: Feedback Deletion
    1. Identify vulnerabilities in the feedback management system.
    2. Attempt to delete feedback entries without admin privileges.

Outcome: Broken access controls allow unauthorized users to perform actions beyond their permissions.

Vulnerability 5: Injection

- Scenario: Admin Username and Login
  1. Attempt an SQL injection attack on the login form by entering malicious SQL commands.
  2. Use payloads like ' OR 1=1;-- to bypass authentication and gain admin access.

Outcome: SQL injection can lead to unauthorized access to database information, including sensitive user credentials.

4. Using the Custom-Built Testing Tool

Tool Overview

This tool automates the process of testing the five vulnerabilities identified in the Juice-Shop application.

Tool Installation and Setup

Install Dependencies:

- Ensure all necessary dependencies are installed, such as Python.

Set Up the Tool:

- Download and extract the tool's source code.
- Configure the tool to target the Juice-Shop application by setting the appropriate URLs and parameters. [Example url andparameter]

Running the Tool

Execution:

- Run the tool from the command line or interface. Use the following command structure:
  python <file_name>.py

Tool Output:

- The tool will generate a report highlighting the success or failure of each vulnerability test and also indicate whether the vulnerability exists in the application or not.
- Review the output to identify vulnerabilities that were successfully exploited.

Interpreting the Results

Result Analysis:

- Each test result is accompanied by a status (e.g., pass/fail) and detailed information about what was tested and what was found.
- Use the results to determine which vulnerabilities need to be addressed in the application.[Needed to be edited]

5. Conclusion

Summary of Findings

- The penetration testing process identified five key vulnerabilities in the Juice-Shop application: XSS, Improper Input Validation, Sensitive Data Exposure, Broken Access Control, and Injection.
- Each vulnerability was successfully tested and documented using the custom-built tool and Burp Suite tool.

6. Appendices

Appendix A: Tool Source Code

[Add source code file path or github link]

Appendix B: Additional Resources

- **OWASP Documentation:** https://owasp.org/
- **Kali Linux Documentation:** https://www.kali.org/docs/
- **Burp Suite Documentation:** https://portswigger.net/burp/documentation