EncryptEdge Labs

Jr. Penetration Tester Internship

Task Report

[shahtareq1216@gmail.com]

Task No: [10]

Credit: Offensive Security

EncryptEdge Labs

Table of Contents

1.0 EncryptEdge Labs Internship Task Report

1.1 Introduction

Web application penetration testing is an essential aspect of cybersecurity, providing critical insights into the vulnerabilities that may exist within an application or its underlying infrastructure. The process involves simulating an attack on a system to uncover security weaknesses before malicious actors can exploit them. This report covers the process of setting up a controlled penetration testing environment using Kali Linux as the attacker machine and Metasploitable as the target machine, which hosts intentionally vulnerable web applications like WebGoat, DVWA, and Mutillidae II.

The goal of this task was to gain hands-on experience in exploiting common web vulnerabilities, such as SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF), within a safe and controlled environment. This report discusses the installation and configuration of these applications, the identification and exploitation of vulnerabilities, and a reflection on the learning experience gained throughout the process.

1.2 Objective

The primary objectives of this task include:

- To understand why a controlled test environment is crucial for conducting web application penetration testing.
- To set up a web application penetration testing environment on Metasploitable and install OWASP WebGoat, DVWA, and OWASP Mutillidae II.
- To explore vulnerabilities in OWASP WebGoat, DVWA, and Mutillidae II, and perform attacks such as XSS, SQL Injection, and CSRF.

1.3 Requirements

- **Virtualization Software:** Virtualization software enables a single physical machine to run multiple virtual machines (VMs), each with its own OS and resources. It's ideal for testing, development, training, and especially useful in cybersecurity practices like penetration testing. Tools like VirtualBox, VMware, and Hyper-V are commonly used. Virtualization boosts hardware efficiency, lowers costs, and ensures environment isolation, so issues in one VM don't impact others. For this task, we'll be using VirtualBox.

- **WebGoat:** WebGoat is an open-source, deliberately vulnerable web app by OWASP designed for hands-on learning of web security flaws like SQL injection, XSS, and more. It provides guided lessons and is ideal for students and professionals to practice safe exploitation and prevention. Easily run via Java or Docker, it's a key tool for secure coding and penetration testing training.

- **DVWA:** Damn Vulnerable Web Application (DVWA) is an intentionally insecure PHP/MySQL web app designed for practicing web security testing. It includes common vulnerabilities like SQL injection, XSS, and CSRF, enabling users to safely learn and test exploitation techniques. DVWA offers adjustable security levels to show how different protections impact attacks, making it an ideal tool for cybersecurity training and penetration testing in a controlled environment.

- **Mutillidae II:** Mutillidae II is an open-source, vulnerable web application by OWASP designed for practicing web security testing. It features over 40 vulnerabilities from the OWASP Top Ten, making it ideal for hands-on learning, CTFs, and tool testing. It is user-friendly with built-in hints and can be run on platforms like XAMPP, LAMP, and Docker.

- **TryHackMe Lab:** Practical experience in TryHackMe labs provides hands-on, interactive learning experience by exploring vulnerabilities like SQL injection and XSS in a deliberately insecure app, following structured tasks. It encourages the use of security tools such as Burp Suite and provides a solid foundation for both beginners and professionals. With community insights and practical challenges, it's an excellent resource for improving web security skills.

2.0 Penetration Testing Environment

2.1 Concept

Penetration testing, especially for web applications, is a critical part of maintaining a secure digital infrastructure. However, such testing must be done in a safe and controlled environment to avoid unintended consequences. Using a dedicated test environment ensures that security assessments are conducted legally, ethically, and without risk to production systems.

2.2 Key Benefits of a Dedicated Test Environment

Reduced Risk of System Damage

- Performing penetration testing on live systems carries the risk of accidentally crashing services, corrupting data, or causing downtime.
- A dedicated test environment isolates these risks, allowing testers to safely simulate real-world attack scenarios without affecting business operations.

Safe Exploration of Vulnerabilities

- Testers can freely explore vulnerabilities like SQL Injection, XSS, and CSRF without fear of breaking critical services or violating service agreements.
- Security professionals can run automated scanners, fuzzing tools, and exploit scripts in a sandboxed setup.

Legal and Ethical Compliance

- Testing real-world systems without explicit permission is illegal and unethical.
- A controlled lab environment, often populated with intentionally vulnerable applications (e.g., DVWA, WebGoat, Mutillidae II), ensures testers operate within legal and ethical boundaries.

Reproducibility and Training

- A test environment allows repeatable testing, making it ideal for training, learning, and simulating attacks in a predictable way.
- Scenarios can be reset and replayed for educational purposes or for validating the effectiveness of new defenses.

2.3 Consequences of Testing on Live Systems

- **Data Breach Risks:** A poorly executed test might unintentionally expose sensitive data.
- **Service Disruption:** Unintended Denial of Service (DoS) may interrupt business-critical operations.
- **Reputation Damage:** Testing mishaps may lead to customer distrust if outages or data leaks occur.
- **Legal Action:** Unauthorized testing may result in criminal charges or civil lawsuits under laws like the Computer Fraud and Abuse Act (CFAA).

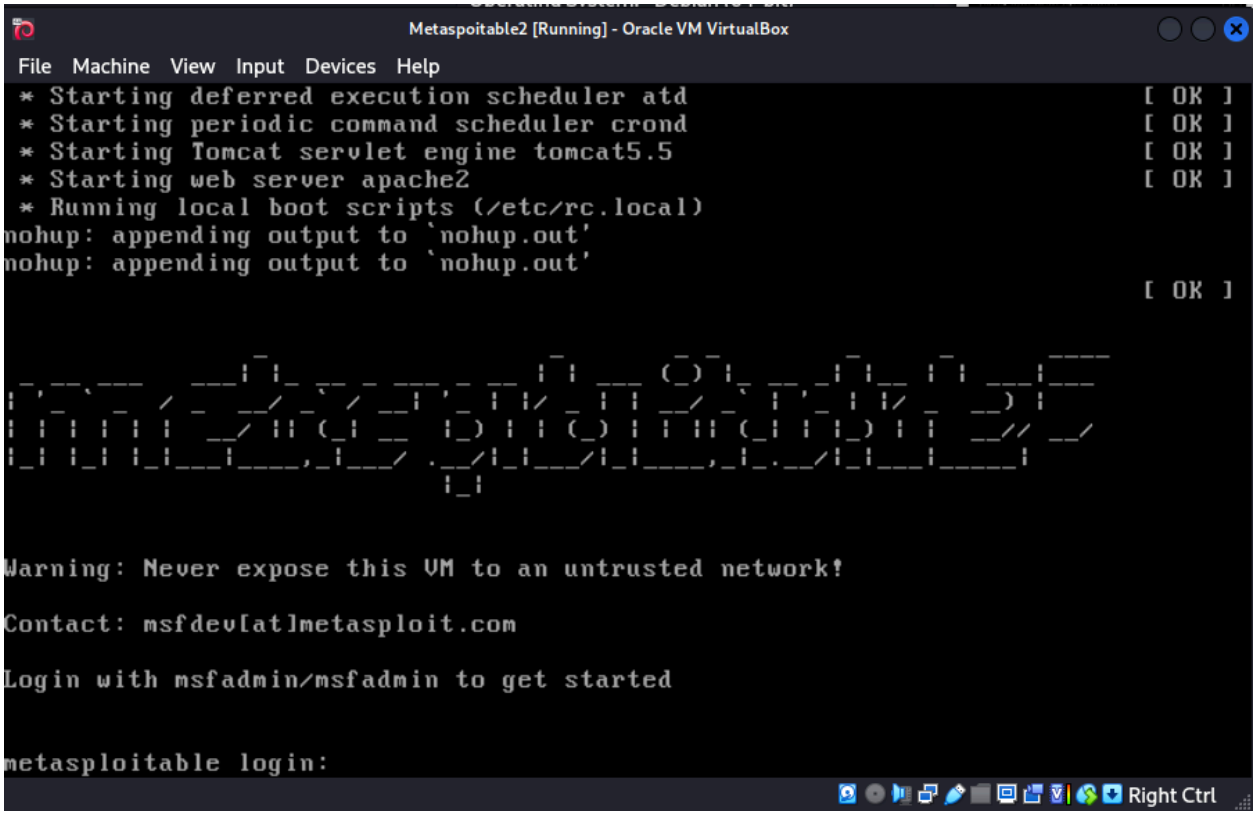
3.0 Installation and Configuration

3.1 Metasploitable 2

Metasploitable 2 is a deliberately vulnerable Linux-based virtual machine developed by Rapid7, designed for practicing penetration testing and ethical hacking in a safe and controlled environment. It is based on an outdated version of Ubuntu and comes pre-configured with numerous insecure services, such as vulnerable versions of Apache, MySQL, Samba, and FTP, making it ideal for learning how to exploit common security flaws. Metasploitable 2 also includes intentionally insecure web applications like DVWA and Mutillidae, which are widely used in cybersecurity training and Capture the Flag (CTF) challenges. It is most commonly used with tools like Metasploit to simulate real-world attack scenarios. For safety, it should only be run in isolated virtual environments to avoid any risk to actual networks.

Installation Steps

- Download the Metasploitable 2 zip file from Rapid7 or sourceforge
- Extract the VM Files
- Set up the VM on virtualbox. This includes
    - Selecting type and version of the VM
    - Allocating memory
    - Choosing virtual hard disk file
    - Configure network as Bridge Adapter or NAT.
- Start the VM and log in to this using msfadmin both as username and password.



3.2 DVWA and Mutillidae II

DVWA (Damn Vulnerable Web Application) is a deliberately insecure web application developed using PHP and MySQL, designed for educational purposes in web security. It provides a platform for security enthusiasts, students, and professionals to practice identifying and exploiting common web vulnerabilities in a safe, legal environment. DVWA includes challenges related to SQL Injection, Cross-Site Scripting (XSS), Command Injection, CSRF, and more. It also features multiple security levels—low, medium, high, and impossible—allowing users to gradually improve their skills. DVWA is widely used in penetration testing labs like Metasploitable for hands-on learning.

Mutillidae II is an intentionally vulnerable web application developed for learning and practicing web application security. Written in PHP and backed by a MySQL database, it simulates a real-world environment where users can explore and exploit a wide range of web vulnerabilities. Mutillidae II covers issues like SQL Injection, Cross-Site Scripting (XSS), Cross-Site Request Forgery (CSRF), Insecure Direct Object References, and many others based on the OWASP Top 10. It also includes built-in hints, tutorials, and different difficulty levels, making it a valuable tool for beginners and advanced learners alike in ethical hacking and secure coding practices.

Installation

Both DVWA and Mutillidae II come pre-installed on Metasploitable 2, providing ready-to-use platforms for practicing web application security and testing various vulnerabilities in a controlled environment.

N.B.: I have been facing difficulties installing WebGoat on Metasploitable and struggled to find effective resources to help me resolve the issue, which ultimately led to my failure in getting it set up. I need more time to figure out a solution.

4.0 Exploring and Exploiting Vulnerabilities

4.1 Concept

Exploring and exploiting vulnerabilities in DVWA (Damn Vulnerable Web Application) and Mutillidae II provides hands-on experience in identifying and taking advantage of common web security flaws. Both applications are designed with intentional vulnerabilities that simulate real-world security issues, making them ideal for learning web application security.

In DVWA, users can test vulnerabilities such as SQL Injection, Cross-Site Scripting (XSS), Command Injection, and File Inclusion, with multiple difficulty levels to help users gradually build their skills. Mutillidae II offers a similar learning environment but includes a broader range of vulnerabilities based on the OWASP Top 10, including Broken Authentication, Cross-Site Request Forgery (CSRF), and Insecure Direct Object References (IDOR).

4.2 DVWA (Damn Vulnerable Web Application)

DVWA is an intentionally vulnerable web application designed for testing and learning about security. It provides varying security levels (low, medium, high) to simulate different scenarios.

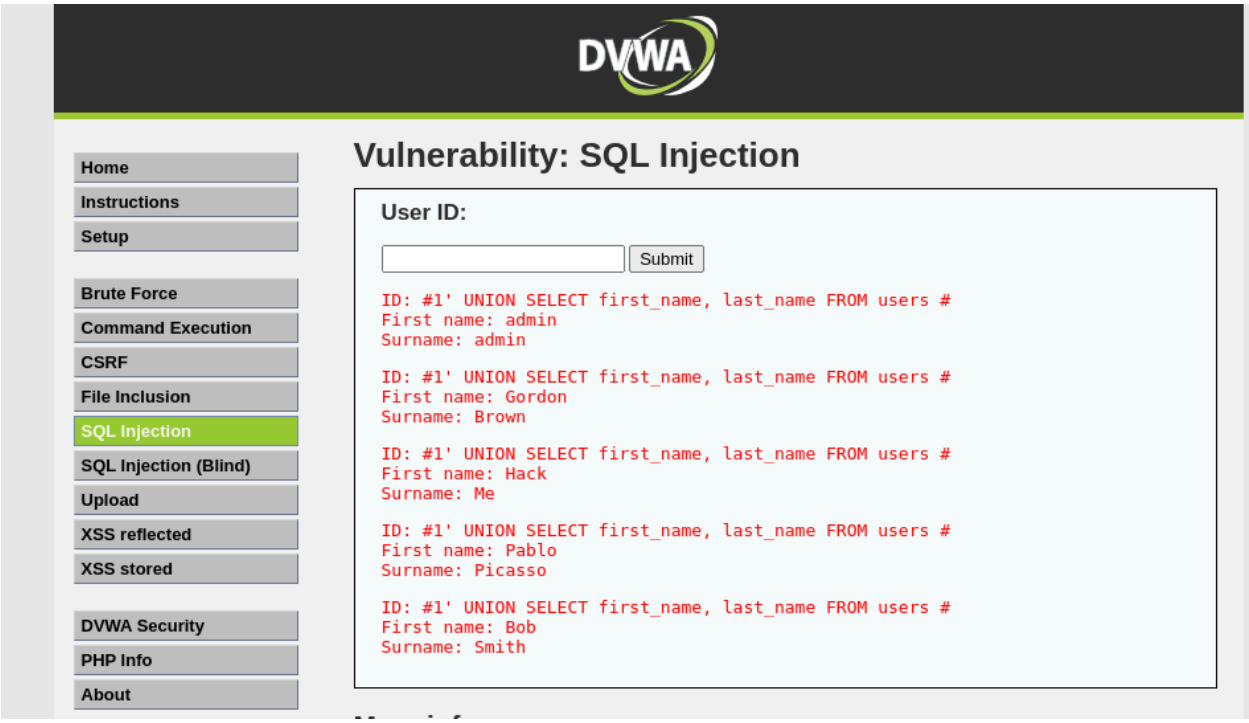
1. **Accessing DVWA:**

- Navigate to the DVWA application: `http://<Metasploitable_IP>/dvwa`.
- Log in with default credentials.

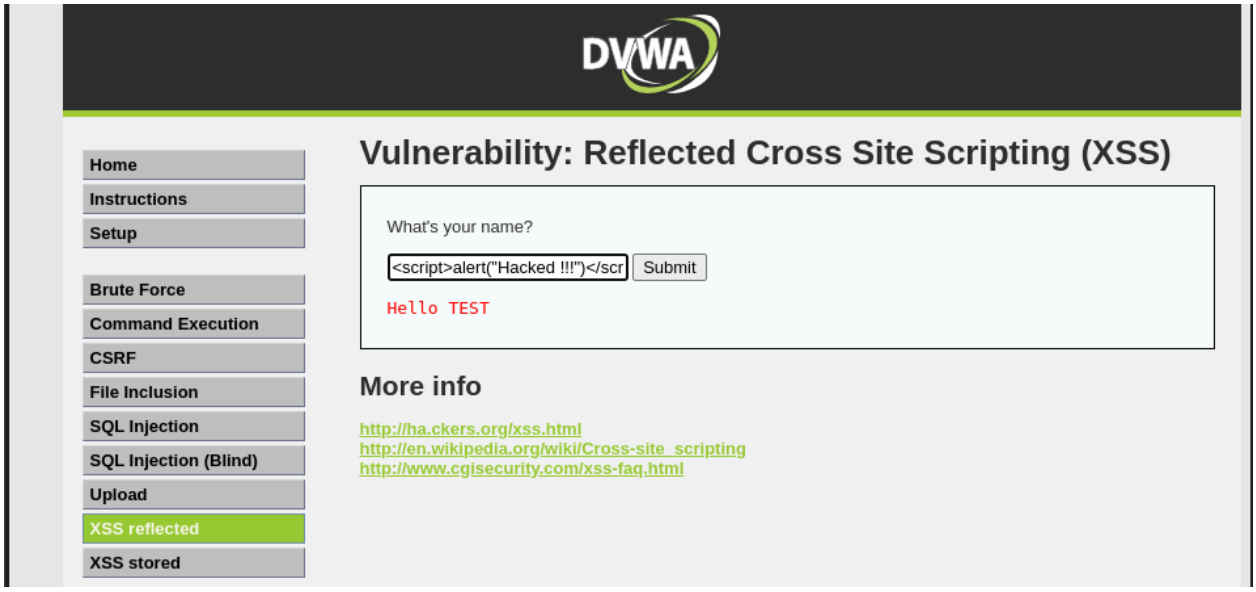
2. **Identify and Exploit Vulnerabilities:** Set the security level to low (to make exploitation easier for testing).

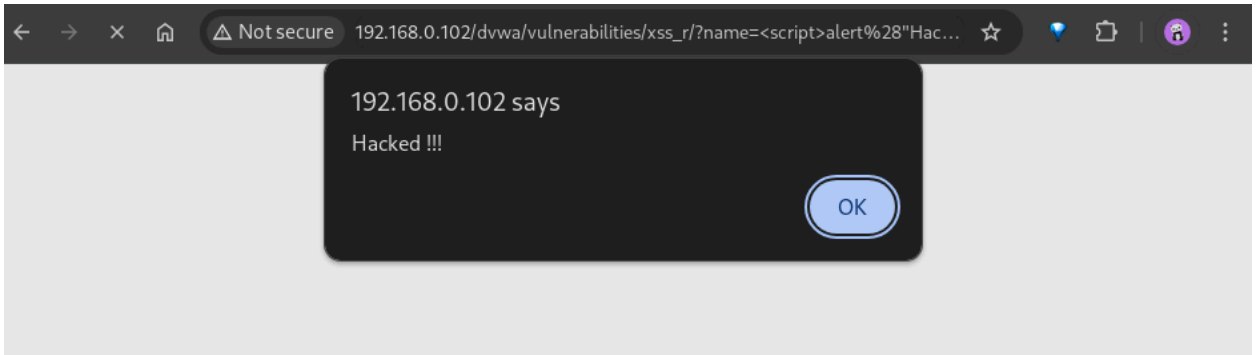
- **SQL Injection:**

- Go to the SQL Injection section.
- Enter a simple SQL Injection payload (e.g., #1' UNION SELECT first_name, last_name FROM users #) in the form of retrieving unauthorized data.

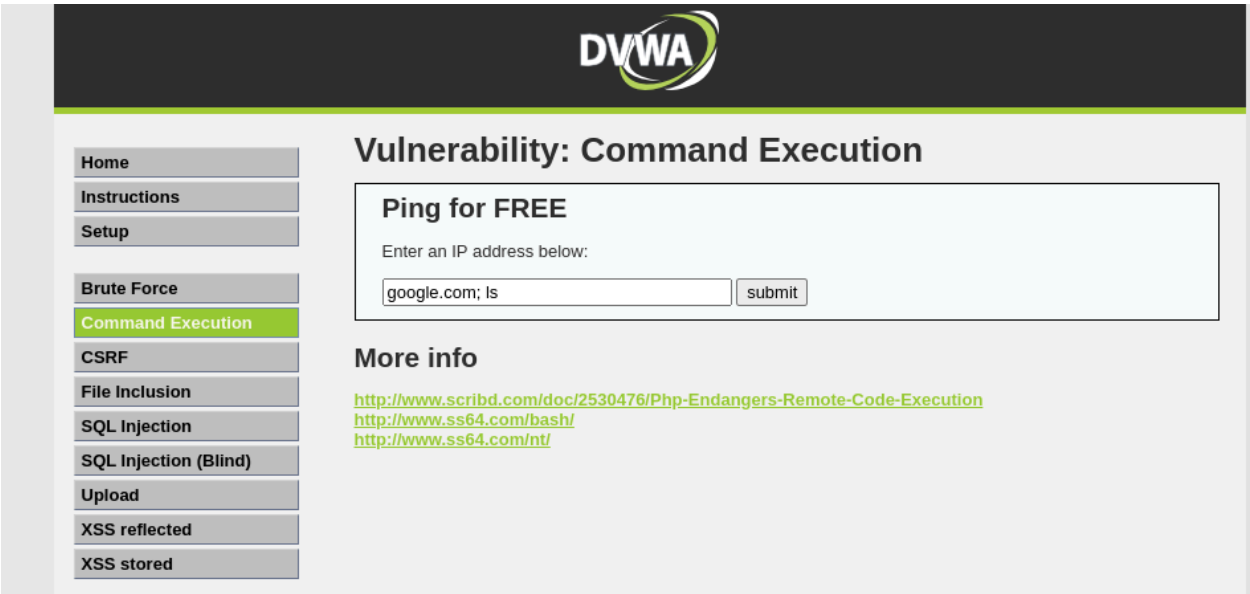


- **Cross-Site Scripting (XSS):**
  - Navigate to the XSS section.
  - Inject the payload `<script>alert('XSS')</script>` into a form field.





- **Command Injection:**
  - Go to the Command Injection section.
  - Inject a simple system command, such as ; ls, into the input field to execute it on the server.


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
Vulnerability: Command Execution
Ping for FREE
Enter an IP address below:
google.com; ls
submit
More info
http://www.scribd.com/doc/2530476/Php-Endangers-Remote-Code-Execution
http://www.ss64.com/bash/
http://www.ss64.com/nt/


DVWA
Home
Instructions
Setup
Brute Force
Command Execution
CSRF
File Inclusion
SQL Injection
SQL Injection (Blind)
Upload
XSS reflected
XSS stored
Vulnerability: Command Execution
Ping for FREE
Enter an IP address below:
submit
PING google.com (142.250.194.78) 56(84) bytes of data.
64 bytes from del12s03-in-f14.1e100.net (142.250.194.78): icmp_seq=1 ttl=58 time=42.
64 bytes from del12s03-in-f14.1e100.net (142.250.194.78): icmp_seq=2 ttl=58 time=53.
64 bytes from del12s03-in-f14.1e100.net (142.250.194.78): icmp_seq=3 ttl=58 time=40.
--- google.com ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2015ms
rtt min/avg/max/mdev = 40.735/45.621/53.881/5.878 ms
help
index.php
source

4.3 Mutillidae II

It is a deliberately insecure web application that covers a wide range of vulnerabilities. It allows testers to experiment with various types of attacks.

1. Accessing Mutillidae II:

- Navigate to the DVWA application: `http://<Metasploitable_IP>/mutillidae`.
- Log in with default credentials.

2. Identify and Exploit Vulnerabilities:

- **SQL Injection:**
  - Go to the SQL Injection section.
  - Input a malicious SQL query (e.g., `' OR 1=1 --`) to bypass authentication or retrieve data from the database.

- **Cross-Site Scripting (XSS):**
  - Find the XSS section.
  - Inject the payload `<script>alert('XSS')</script>` into a form field.

Mutillidae: Born to be Hacked

Version: 2.1.19 | Security Level: 0 (Hosed) | Hints: Disabled (0 - I try harder) | Not Logged In

Home | Login/Register | Toggle Hints | Toggle Security | Reset DB | View Log | View Captured Data

- Core Controls
- OWASP Top 10
- Others
- Documentation
- Resources

Site hacked...err...quality-tested with Samurai WTF, Backtrack, Firefox, Burp-Suite, Netcat, and these Mozilla Add-ons

@webpwnized

Mutillidae Channel

Developed by Adrian "Irongeek" Crenshaw

View your details

Back

Please enter username and password to view account details

Name: <script>alert('XSS');</script>

Password:

View Account Details

Dont have an account? Please register here

Error: Failure is always an option and this situation proves it

Line	126
Code	0
File	/var/www/mutillidae/user-info.php
Message	Error executing query: You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near 'XSS');' AND password='' at line 1
Trace	#0 /var/www/mutillidae/index.php(469): include() #1 {main}
Diagnotic Information	SELECT * FROM accounts WHERE username='' AND password=''

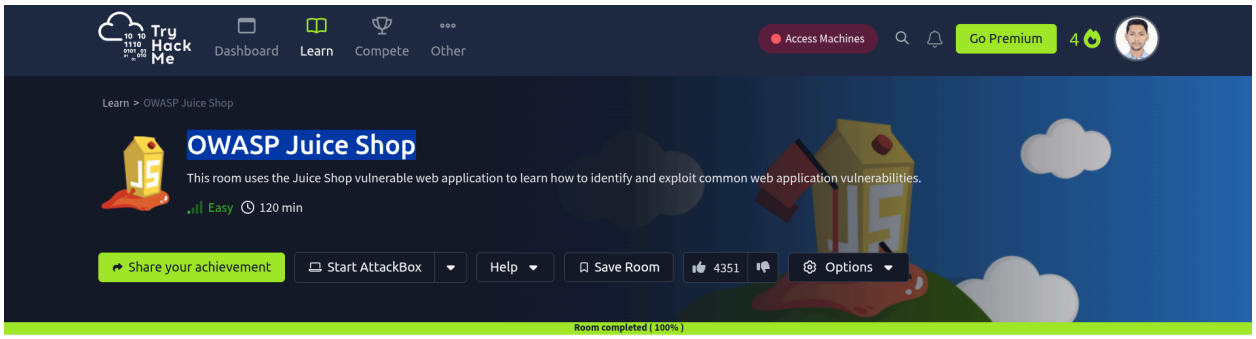
Did you setup/reset the DB?


Not secure
192.168.0.103/mutillidae/index.php?page=user-info.php&userna...
192.168.0.103 says
XSS
OK

5.0 Hands-On Labs

5.1 OWASP Juice Shop

The OWASP Juice Shop TryHackMe lab offers a practical and engaging environment for learning web application security through hands-on challenges. Built around a vulnerable e-commerce platform, it effectively covers key OWASP Top 10 issues such as Broken Access Control, Cross-Site Scripting (XSS), SQL Injection, and Authentication Flaws. Users explore how unauthorized access can be gained due to poor privilege enforcement, craft XSS payloads to exploit input vulnerabilities, perform SQL injections to manipulate database queries, and uncover authentication weaknesses like insecure login mechanisms and weak password policies. The lab is well-structured, beginner-friendly, and highly relevant, making it an excellent resource for building real-world penetration testing and secure coding skills.

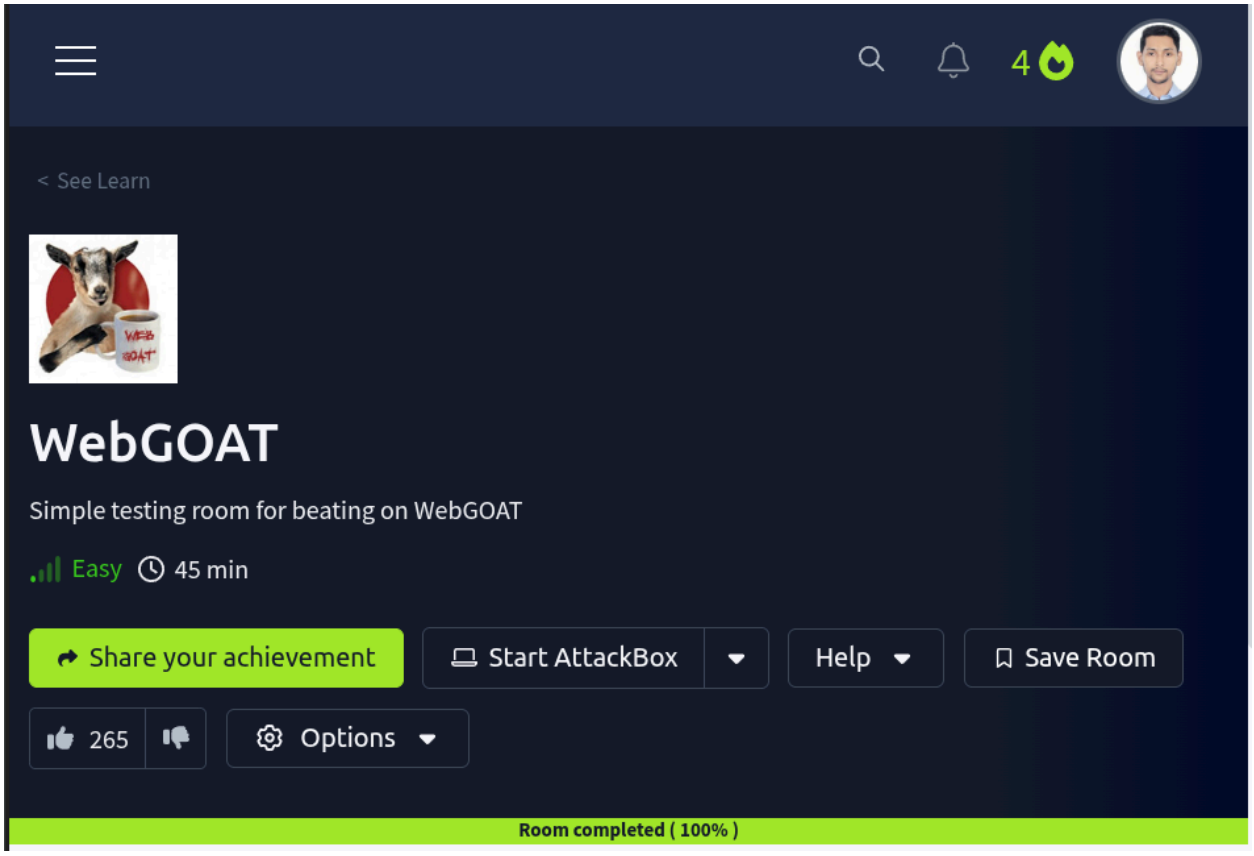


5.2 Mutillidae II

The OWASP Mutillidae II TryHackMe lab offers an interactive and practical environment for learning web application security. With step-by-step guidance, it's a valuable resource for beginners to build hands-on ethical hacking skills.

5.3 WebGoat

The OWASP WebGoat TryHackMe room provides a structured and hands-on approach to learning web application security. With clear guidance and interactive tasks, it's a great resource for beginners to build ethical hacking and secure coding skills.


4
< See Learn
WebGOAT
Simple testing room for beating on WebGOAT
Easy
45 min
Share your achievement
Start AttackBox
Help
Save Room
265
Options
Room completed (100%)

6.0 Reflection

6.1 Challenges Faced

One of the main challenges faced during the setup of the web application penetration testing environment was ensuring proper configuration of the vulnerable applications (WebGoat, DVWA, and Mutillidae II) on the Metasploitable virtual machine. Initially, some of the web applications had compatibility issues, which delayed the configuration process. Additionally, troubleshooting networking problems between Kali Linux (attacker machine) and Metasploitable (target machine) posed some difficulties, as both VMs needed to be on the same network to communicate effectively. Once resolved, it became clear that consistent networking settings are essential for the successful communication between VMs. Moreover, allocating storage to the VM resulted in several issues due to insufficient storage capacity.

Another challenge was identifying and exploiting vulnerabilities within the test applications. Although the documentation provided detailed steps, some of the vulnerabilities required further exploration and experimentation to successfully exploit. This hands-on experience proved to be invaluable in reinforcing the need for a deeper understanding of web vulnerabilities and attack vectors.

6.2 Lesson Learned

The task reinforced the importance of patience and persistence when configuring a controlled testing environment. The step-by-step process of setting up vulnerable web applications provided deeper insights into the complexity of penetration testing environments. I also learned that vulnerability exploitation is not always straightforward—some vulnerabilities required an understanding of various attack vectors, and different applications presented different levels of difficulty in exploiting them.

The process highlighted the significance of a well-configured environment for penetration testing, as issues in the setup can lead to inaccurate results or wasted time. Additionally, hands-on practice with real vulnerabilities, like SQL Injection and Cross-Site Scripting (XSS), helped solidify the theoretical knowledge gained from online resources.

6.3 Reflection on the Analysis

Setting up a controlled penetration testing environment provides significant value for learning about and practicing real-world web application vulnerabilities. By isolating these activities from production environments, the process ensures both safety and ethical standards. Hands-on experience in identifying and exploiting vulnerabilities helped deepen my understanding of web application security and the tools used by attackers.

This exercise also reinforced the importance of continuously testing web applications in a controlled manner to identify vulnerabilities before they are exploited in the wild. In a production environment, vulnerabilities can remain undetected, leading to potential data breaches and other security incidents. Therefore, penetration testing must be conducted regularly to maintain security hygiene and ensure systems remain resilient against emerging threats.

7 Conclusion

In conclusion, setting up a web application penetration testing environment provided valuable hands-on experience with common vulnerabilities like SQL Injection, XSS, and CSRF. Installing vulnerable applications such as WebGoat, DVWA, and Mutillidae II on Metasploitable enhanced my technical skills and deepened my understanding of web application security. The challenges faced during the setup reinforced the importance of a controlled test environment for safe and ethical testing. This task highlighted the need for regular vulnerability testing and solidified my understanding of various attack techniques, ultimately strengthening my foundation in web application security for future learning.

This Internship Task report was developed on [April 08, 2025]

By:

shahtareq1216@gmail.com