Ethical Hacking Assessment Report

Task Title: Web Application Vulnerability Assessment & Custom Port Scanner Development

Intern Name: Maxwell B. Antwi

Company: Hack Secure

Internship Duration: 10th April – 1ST May

Project Mentor: Nishant Prajapati

University: Ashesi University

Major: Computer Engineering

Domain: Red Team

GitHub Repo: Arhmfaculty/Port-Scanner:

1. Task Overview

This task involved conducting a comprehensive security assessment of the website http://testphp.vulnweb.com and demonstrating real-world hacking techniques using professional tools and self-developed scripts. The goal was to identify potential vulnerabilities, exploit weaknesses ethically, and understand the risk impact of each flaw discovered.

Tasks and Responsibilities

I was expected to:

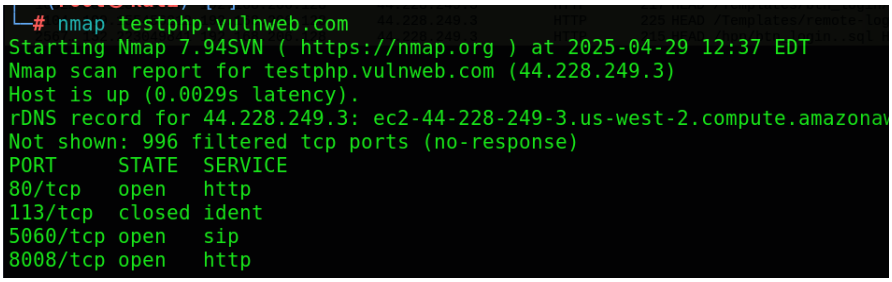
- Perform network and web application scanning to identify open ports and running services.
- Enumerate hidden files and directories for possible entry points.
- Test for known web vulnerabilities including SQL Injection and Cross-Site Scripting.
- Capture and analyze network traffic to detect insecure data transmissions.
- Develop a custom TCP port scanner using Python.
- Prepare a comprehensive vulnerability report with screenshots and mitigation recommendations.

2. Methodology & Execution

Step 1: Port Scanning using Nmap

I initiated the engagement by performing a port scan on the target website. Nmap was used to scan open, filtered, and closed TCP ports.

- **Result:**
  - 996 TCP ports were filtered.
  - Port 113 (ident) was closed.
  - Ports 80 (HTTP), 8008 (alternate HTTP), and 5060 (SIP) were found **open**.



Step 2: Service Enumeration

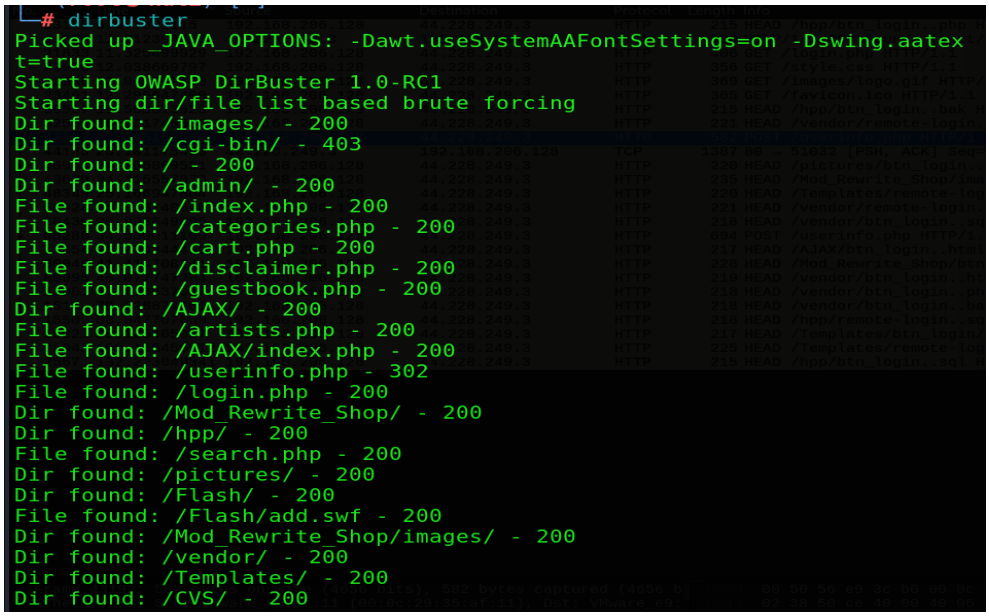
Service version detection was conducted on the open ports to identify the services and versions in use.

- **Key Finding:**
  - Port 80 was running **nginx 1.19.0** HTTP server.
  - No critical service responses on ports 8008 or 5060.

Step 3: Directory Brute-Forcing (DirBuster)

Using DirBuster, I enumerated hidden files and directories on the web server. These resources are often entry points for unauthorized access.

- **Files Found:**
- /artists.php, /categories.php, /disclaimer.php, /guestbook.php,
- /userinfo.php, /login.php, /search.php, /AJAX/index.php, /Flash/add.swf
- **Directories Found:**
- /, /images, /cgi-bin, /admin, /pictures, /AJAX/,
- /Mod_Rewrite_Shop, /http, /vendor/, /Templates/



Step 4: Default Credential Login

Navigating to /login.php, I discovered that the application accepted **default credentials**:

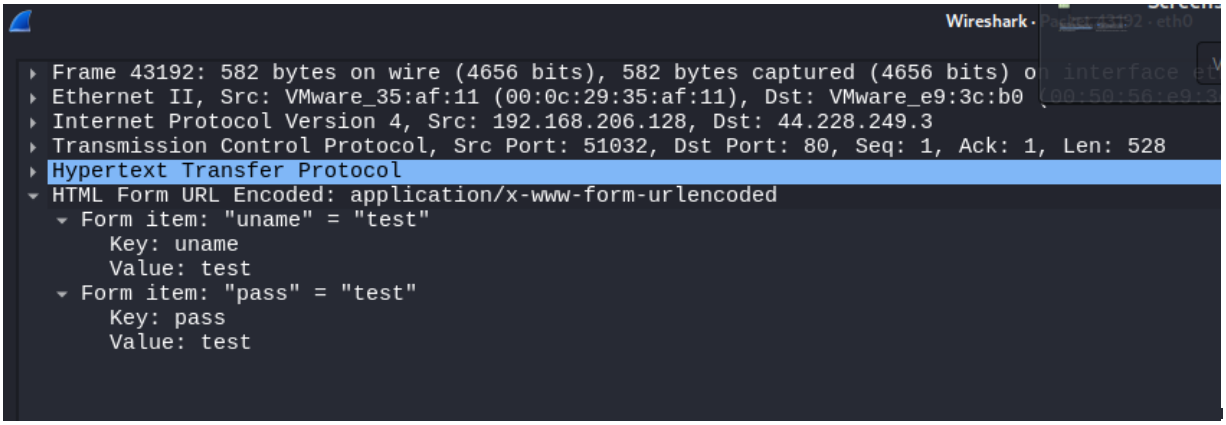
- **Username:** test
- **Password:** test

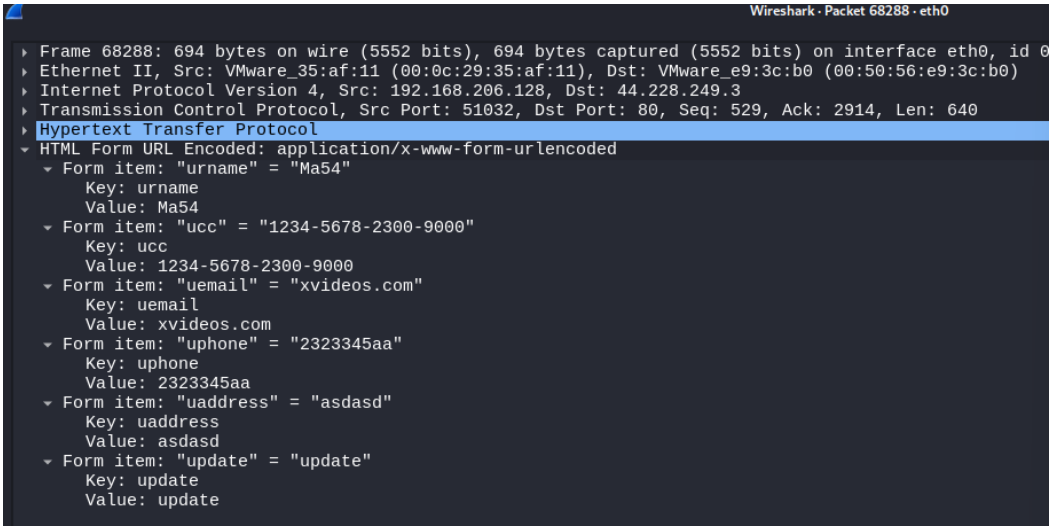
This allowed access to the userinfo.php page, where sensitive data such as username, credit card number, email, and phone number were visible and **editable**.

Step 5: Traffic Interception with Wireshark

I monitored network traffic during login using Wireshark to analyze the transmission of sensitive data.

- **Key Discovery:**
  The application transmits credentials and user data **in clear text** over HTTP, making it vulnerable to **credential theft** through **network sniffing**. This sniffing reveled user login details as well as user information.

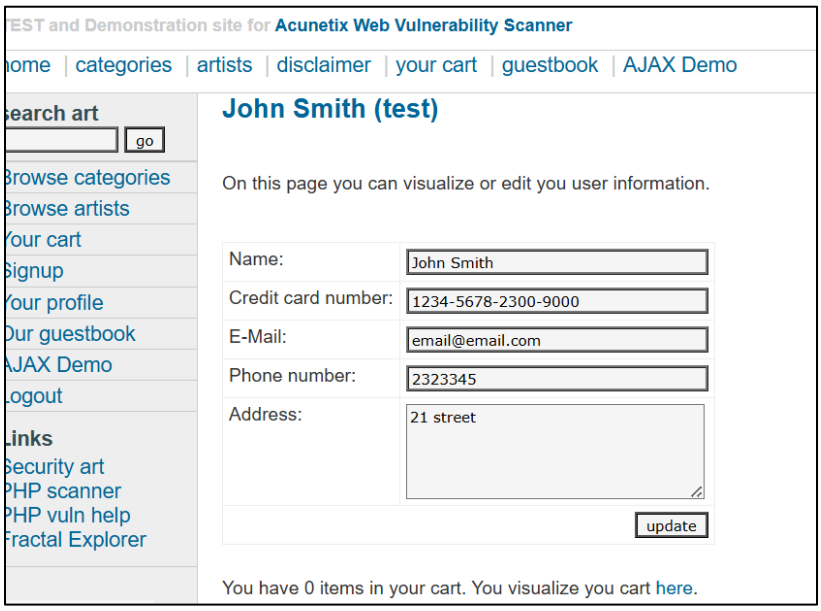




Step 6: SQL Injection (SQLi)

I tested the login and searched page inputs for SQL Injection vulnerabilities.

- **Login Page:**
  - **Payload Used:**
  - Username: admin' --
  - Password: '
  - **Result:** Successfully bypassed authentication and logged in as another user (John Smith) without needing their password.



Step 7: Reflected Cross-Site Scripting (XSS)

I tested input fields for XSS vulnerabilities using this payload:

```
<script>alert('XSS')</script>
```

- **Affected Fields:** Search bar and comment section.
- **Result:** Successful reflected XSS popups were triggered, confirming a reflected XSS vulnerability.

3. Additional Project: Custom Port Scanner

As part of the ethical hacking project, I developed a **custom TCP port scanner** in Python. This tool replicates basic Nmap functionality, allowing users to:

- Scan a given IP/domain for open TCP ports within a specified range.
- Resolve domain names to IPs.
- Track scan duration.
- Display open ports found during execution.

Tool Description:

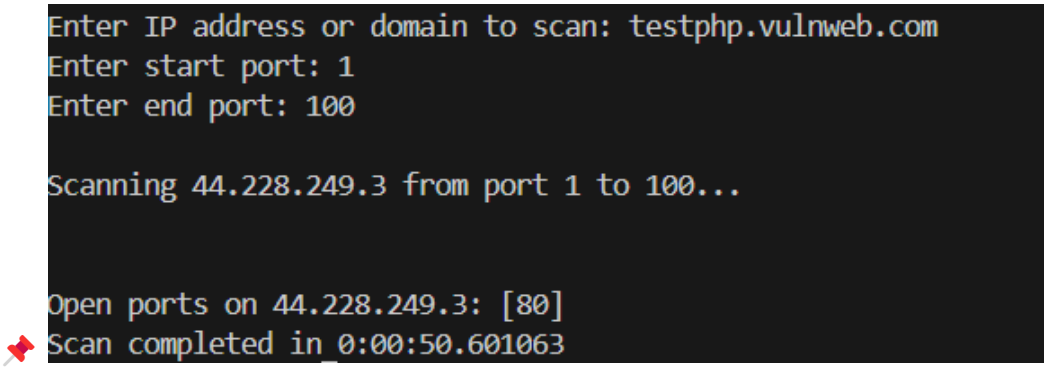
- The tool uses Python's socket module to attempt TCP connections to each port.
- Results are printed and include all **open ports** found in the scanned range.
- Error handling for invalid inputs and unreachable hosts is built in.

Code Snippet (Main Scanner Function):

```
def portScanner(ip, start_port, end_port):
    ...
```

How It Works:

1. User enters the IP address or domain.
2. Specifies the port range.
3. The scanner attempts connections to each port in the range.
4. Reports all open ports along with the scan time.



4. Conclusion

Through this intermediate ethical hacking project, I practiced and demonstrated essential techniques including port scanning, service enumeration, directory discovery, vulnerability exploitation (SQLi, XSS), traffic interception, and authentication bypass. Developing my own scanner helped reinforce my understanding of how TCP connections and port states are handled programmatically.

Key Skills Gained:

- Vulnerability analysis and exploitation.
- Traffic interception with Wireshark.
- Developing custom tools in Python.
- Manual testing of real-world web application flaws.
- Documenting findings professionally.

5. Challenges Encountered

- **Identifying the Right Scripts for SQL Injection and XSS:** Initially, I was unsure of which payload formats or scripts to use for performing successful SQL injections and cross-site scripting attacks. I overcame this by researching online resources, reviewing different injection techniques, and iteratively testing various formats until the inputs produced the expected behavior.
- **Analyzing the Correct Packets in Wireshark:** With Wireshark capturing a large volume of network traffic, pinpointing the exact packets containing login credentials was a challenge. I worked alongside OpenAI guidance to refine my filtering techniques—focusing on HTTP POST requests—and learned how to isolate and interpret sensitive data in captured packets.
- **Building the Custom Port Scanner:** Developing the port scanner from scratch required me to understand how TCP connections are established and how domain names are resolved to IP addresses. I had to familiarize myself with Python's socket module and implement logic for scanning port ranges, managing connection timeouts, and handling DNS resolution efficiently.

6. Acknowledgment

I sincerely thank **Hack Secure** for this enriching internship opportunity. This project sharpened my practical cybersecurity skills and exposed me to real-world penetration testing scenarios. The mentorship and support from the Hack Secure team were instrumental in helping me think critically like an ethical hacker and execute each task responsibly and effectively.