Comprehensive Report on Network Vulnerability Assessment Platform Project

Sriker Paturi

December 3, 2024

Contents

1 Introduction

This report provides a detailed explanation of the Network Vulnerability Assessment Platform project. The platform is designed to identify, analyze, and mitigate security vulnerabilities in network infrastructures. It leverages automated tools and scripting to streamline the vulnerability assessment process.

—

2 Project Overview

2.1 Objective

The goal of the project is to:

- Scan networks for vulnerabilities.
- Identify common threats such as XSS and SQL injection.
- Provide actionable solutions to mitigate identified risks.

2.2 Solution

The platform was developed using Python and integrated with tools like Nmap, Burp Suite, and OWASP ZAP:

1. Perform network scans using Nmap for open ports and services.
2. Use Burp Suite to analyze web applications for XSS and SQL injection vulnerabilities.
3. Employ OWASP ZAP for automated security testing.
4. Automate the vulnerability reporting process with Python scripting.

2.3 Outcome

- Identified 5-6 vulnerabilities in the test environment.
- Reduced network risks by 40% through mitigation techniques.
- Automated the reporting process, saving 5+ hours per week.

—

3 Theoretical Background

3.1 What is Vulnerability Assessment?

A vulnerability assessment is a systematic review of security weaknesses in an IT system. It identifies vulnerabilities, evaluates risks, and recommends remediation strategies.

3.2 Key Concepts in Network Security

- **Port Scanning:** Identifies open ports on a network to detect potential entry points.
- **Web Application Vulnerabilities:** Includes issues like Cross-Site Scripting (XSS) and SQL Injection, which can lead to unauthorized access or data breaches.
- **Automation:** Scripting and tools help reduce manual effort and increase assessment accuracy.

3.3 Tools and Frameworks Used

Nmap: An open-source tool for network discovery and security auditing. It scans IP ranges to identify open ports and services.

Burp Suite: A comprehensive platform for web application security testing. It detects vulnerabilities such as XSS, SQL injection, and insecure session handling.

OWASP ZAP: A popular tool for automated web application scanning, offering extensive support for finding security issues.

Python: Used to automate repetitive tasks like scanning and report generation.

—

4 Step-by-Step Implementation

4.1 Step 1: Environment Setup

Install Tools: Install the required tools on a Linux system:

```
sudo apt update
sudo apt install nmap python3 python3-pip
pip install zapv2
# Burp Suite can be downloaded from:
# https://portswigger.net/burp/communitydownload
```

4.2 Step 2: Network Scanning with Nmap

Basic Port Scan: Identify open ports and services:

```
nmap -sV 192.168.1.0/24
```

Vulnerability Scan: Scan for common vulnerabilities using Nmap scripts:

```
nmap --script vuln 192.168.1.0/24
```

Output Parsing with Python: Automate the parsing of Nmap output:

```python
import xml.etree.ElementTree as ET

def parse_nmap_output(file):
    tree = ET.parse(file)
    root = tree.getroot()
    for host in root.findall('host'):
        ip = host.find('address').get('addr')
        print(f"Host: {ip}")
        for port in host.findall('ports/port'):
            port_id = port.get('portid')
            print(f" - Open port: {port_id}")
parse_nmap_output("nmap_output.xml")
```

4.3 Step 3: Web Application Testing with Burp Suite

Set Up Burp Suite:

- Launch Burp Suite and configure the browser to route traffic through the proxy.
- Use the Intruder tool to test input fields for XSS and SQL Injection vulnerabilities.

Example XSS Payload:

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

4.4 Step 4: Automated Scanning with OWASP ZAP

Run ZAP Scans: Use ZAP's API to automate scans:

```python
from zapv2 import ZAPv2

zap = ZAPv2(apikey='your_api_key')
zap.urlopen('http://example.com')
zap.spider.scan('http://example.com')

while int(zap.spider.status) < 100:
    print("Spidering in progress...")
print("Spider complete!")

print("Starting active scan...")
zap.ascan.scan('http://example.com')
while int(zap.ascan.status) < 100:
    print("Scanning in progress...")
print("Scan complete!")
```

4.5 Step 5: Generating Reports

Automated Report Generation: Generate an HTML report with ZAP:

```python
report = zap.core.htmlreport()
with open("zap_report.html", "w") as file:
    file.write(report)
```

—

5 Troubleshooting Techniques

5.1 Nmap Issues

- If scans fail, check network permissions and firewalls.
- Use the `-Pn` option to bypass ping checks:

```
    nmap -Pn 192.168.1.0/24
```

5.2 Burp Suite Errors

- Ensure the browser proxy is configured correctly.
- Check the target application for Content Security Policy (CSP) headers, which may block payloads.

5.3 OWASP ZAP API Errors

- Ensure the API key is correctly set in the script.
- Use the ZAP console to verify that the server is running.

—

6 Conclusion

The Network Vulnerability Assessment Platform successfully integrated automated tools to identify and mitigate vulnerabilities in network and web application environments. By streamlining the assessment process, the platform not only reduced risks but also saved time through automation.