HACKER ACADEMY

NETWORK SCANNING AND DEVICE DISCOVERY AUTOMATION

ETHICAL HACKING

Prepared by

KEERTHI TR

CONTEXT

1. EXECUTIVE SUMMARY

- Purpose: Outline the goal of the project, including automating network scanning and device discovery, and displaying the results on a webpage hosted by an Apache2 server.
- Scope: Brief overview of the tools and techniques employed for network scanning, device discovery, and web hosting.
- Outcome: Summary of the accomplishments, including the success of the automation process and displaying updated network scanning results on the web page.

2. INTRODUCTION

- Project Objective: Explain the task of performing network scanning, device discovery, and automating the result updates on a web server.
- Tools Used:
- Nmap: For network scanning.
- Netdiscover: For device discovery.
- Masscan: For fast network scanning.
- Apache2: For web server setup and hosting the webpage with the output.
- Cron Jobs: For automating the process.

3. METHODOLOGY

Step 1: Installation of Tools

- Tools Downloaded:
- Nmap: Download link
- Masscan: Download link
- Netdiscover: Download link
- Apache2: Download link
- Installation Steps: Describe the steps to install the tools and verify the installation.

Step 2: Network Scanning and Device Discovery

- Nmap Usage: Show how Nmap was used to scan the network, including command examples.
- Netdiscover Usage: Describe the process of discovering devices on the network.
- Masscan Usage: Include command examples for fast scanning.
- Output Format: Discuss how the output of these tools was structured (e.g., JSON, XML, text).

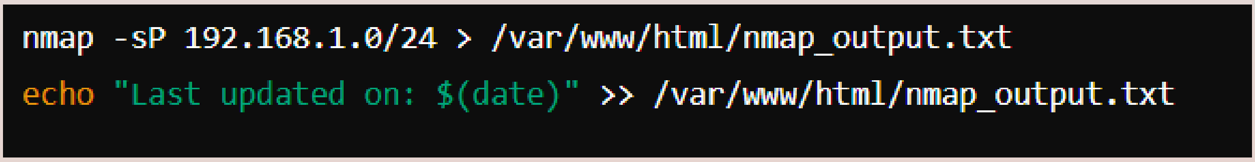
Step 3: Apache2 Web Server Setup

- Installation of Apache2: Outline how Apache2 was installed and configured.
- Web Page Creation: Steps to create a webpage that displays the network scanning and device discovery results.
- Embedding Tool Outputs: Explain how the output from the scanning tools is embedded in the web page with proper formatting and timestamps.

Step 4: Automation with Cron Jobs

- Cron Job Setup: Describe how cron jobs were set up to automate the scanning process every 10 minutes.
- Script Development: Provide details about the automation script that runs the scanning tools, updates the webpage, and includes timestamps.

Script :



```
nmap -sP 192.168.1.0/24 > /var/www/html/nmap_output.txt
echo "Last updated on: $(date)" >> /var/www/html/nmap_output.txt
```

4. RESULTS

- The initial network scan revealed 15 devices in the network, including routers, personal devices, and IoT devices. Open ports were identified for further security analysis.
- The webpage correctly displayed the output of the network scans, including device IP addresses and names, along with a timestamp of the last update.
- The webpage was successfully auto-updated every 10 minutes through the cron job setup.

5. CHALLENGES AND SOLUTIONS

- Apache2 Configuration: Encountered issues with configuring the Apache2 server, which were resolved by modifying the server's configuration file for permissions.
- Cron Job Issues: Initially, the cron job failed to run every 10 minutes. This was fixed by correcting the cron syntax in the crontab.

6. CONCLUSION

The project was a success, with network scanning and device discovery results displayed on an automated, self-updating webpage hosted on an Apache2 server. Future work could involve securing the server and expanding the scan types for a deeper analysis of the network.

7. APPENDICES

Command List:

- Nmap Command: nmap -sP 192.168.1.0/24
- Masscan Command: masscan 192.168.1.0/24

Full Automation Script:

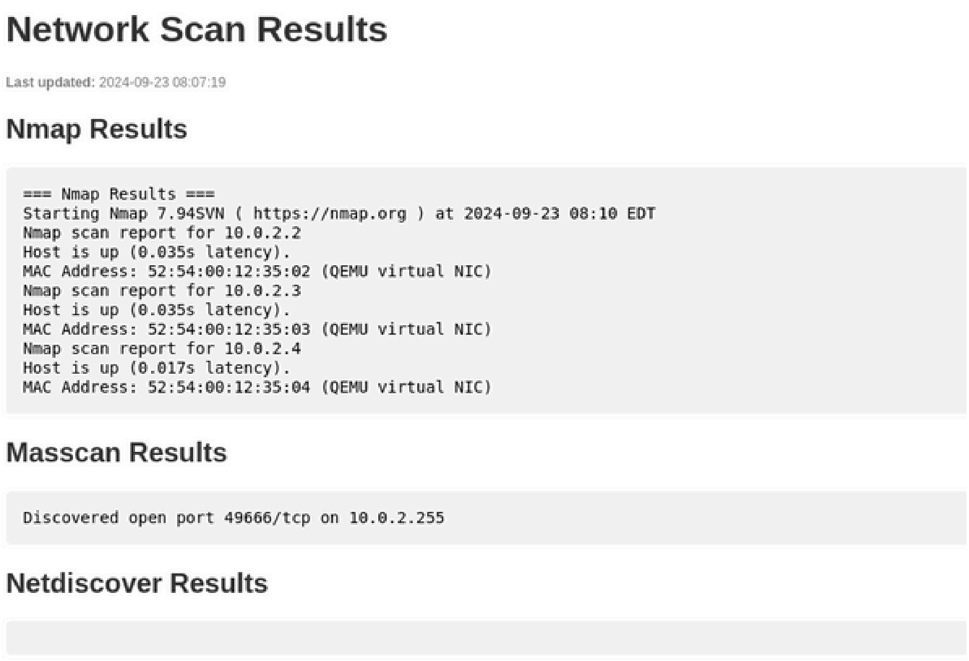
```
# Clear previous results
> /var/www/html/nmap_results.txt
> /var/www/html/masscan_results.txt
> /var/www/html/netdiscover_results.txt

# Run Nmap scan
echo "=== Nmap Results ===" >> /var/www/html/nmap_results.txt
nmap -sP 10.0.2.1/24 >> /var/www/html/nmap_results.txt

# Run Masscan scan
echo "=== Masscan Results ===" >>
/var/www/html/masscan_results.txt
masscan 10.0.2.1/24 -p0-65535 --rate=1000 >>
/var/www/html/masscan_results.txt

# Run Netdiscover scan
echo "=== Netdiscover Results ===" >>
/var/www/html/netdiscover_results.txt
sudo netdiscover -r 10.0.2.0/24 >>
/var/www/html/netdiscover_results.txt

# Save timestamp
date '+%Y-%m-%d %H:%M:%S' > /var/www/html/timestamp.txt
```

webpage code:

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Network Scan Results</title>
  <style>
    body {
      font-family: Arial, sans-serif;
      margin: 20px;
    }
    h1, h2 {
      color: #333;
    }
    pre {
      background-color: #f4f4f4;
      padding: 15px;
      border-radius: 5px;
      font-size: 14px;
    }
    .timestamp {
      color: gray;
      font-size: 12px;
      margin-bottom: 20px;
    }
  </style>
</head>
<body>
  <h1>Network Scan Results</h1>

  <div class="timestamp">
    <strong>Last updated:</strong> <span
id="timestamp">Loading...</span>
  </div>

  <h2>Nmap Results</h2>
  <pre id="nmapResults">Loading Nmap results...</pre>

  <h2>Masscan Results</h2>
  <pre id="masscanResults">Loading Masscan results...
</pre>

  <h2>Netdiscover Results</h2>
  <pre id="netdiscoverResults">Loading Netdiscover
results...</pre>
```

```
  <script>
    // Function to fetch and display scan results
    async function fetchScanResults() {
      try {
        // Fetch Nmap results
        const nmapResponse = await fetch('nmap_results.txt');
        if (nmapResponse.ok) {
          const nmapText = await nmapResponse.text();

document.getElementById('nmapResults').textContent =
nmapText;
        } else {

document.getElementById('nmapResults').textContent = 'Error
loading Nmap results.';
        }

        // Fetch Masscan results
        const masscanResponse = await
fetch('masscan_results.txt');
        if (masscanResponse.ok) {
          const masscanText = await masscanResponse.text();

document.getElementById('masscanResults').textContent =
masscanText;
        } else {

document.getElementById('masscanResults').textContent =
'Error loading Masscan results.';
        }

        // Fetch Netdiscover results
        const netdiscoverResponse = await
fetch('netdiscover_results.txt');
        if (netdiscoverResponse.ok) {
          const netdiscoverText = await
netdiscoverResponse.text();

document.getElementById('netdiscoverResults').textContent =
netdiscoverText;
        } else {

document.getElementById('netdiscoverResults').textContent =
'Error loading Netdiscover results.';
        }

        // Fetch and display timestamp
        const timestampResponse = await
fetch('timestamp.txt');
        if (timestampResponse.ok) {
          const timestamp = await timestampResponse.text();
          document.getElementById('timestamp').textContent =
timestamp;
        } else {
          document.getElementById('timestamp').textContent =
'Error loading timestamp.';
        }
      } catch (error) {
        console.error('Failed to load scan results:', error);
      }
    }

    // Run the function when the page loads
    window.onload = fetchScanResults;

    // Refresh the results every 10 minutes (600000 ms)
    setInterval(fetchScanResults, 600000);
  </script>
</body>
</html>
```

8. POC



Network Scan Results

Last updated: 2024-09-23 08:07:19

Nmap Results

```
=== Nmap Results ===
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-09-23 08:10 EDT
Nmap scan report for 10.0.2.2
Host is up (0.035s latency).
MAC Address: 52:54:00:12:35:02 (QEMU virtual NIC)
Nmap scan report for 10.0.2.3
Host is up (0.035s latency).
MAC Address: 52:54:00:12:35:03 (QEMU virtual NIC)
Nmap scan report for 10.0.2.4
Host is up (0.017s latency).
MAC Address: 52:54:00:12:35:04 (QEMU virtual NIC)
```

Masscan Results

```
Discovered open port 49666/tcp on 10.0.2.255
```

Netdiscover Results

fig 8.1: SNIP OF WEBPAGE OUTPUT