Gmail marsupial seven <marsupialseven@gmail.com>

read me
1 message

marsupial seven <marsupialseven@gmail.com> Mon, Jan 13, 2025 at 11:05 PM
To: marsupial seven <marsupialseven@gmail.com>

README: Nmap Vulnerability Scanning Tool

Overview

This tool is designed to perform network scanning using Nmap and then check for known vulnerabilities based on the services discovered during the scan. It integrates Nmap scanning capabilities with a vulnerability database (in JSON format) to match detected services and display relevant CVEs (Common Vulnerabilities and Exposures). The tool provides an efficient method to identify potential risks in your network by cross-referencing Nmap service data with known vulnerabilities.

Features

- **Network Discovery**: Automatically detects local network information and scans a user-defined IP address or network range.
- **Multiple Scan Types**:
  - Simple scan (default ports).
  - Enhanced scan (all ports with service and OS detection).
  - Aggressive scan (includes detailed version detection, OS, traceroute, and more).
- **CVE Matching**: For each service detected, the tool checks for matching vulnerabilities in the provided JSON database and displays detailed CVE information.
- **Reporting**: Outputs both the Nmap scan results and the vulnerability information to a report file.

Requirements

- Python 3.x
- Nmap installed on the system
- A JSON file (`cve_vuln_data.json`) containing CVE details (this should be in the same directory as the script)

Files Used

- **`cve_vuln_data.json`**: This file should contain the CVE data. The format is expected to have entries with fields like `cve_id`, `service`, `description`, etc.

Example entry:
```json
{
"cve_id": "CVE-2020-12345",
"service": "OpenSSH",
"description": "A vulnerability in OpenSSH could allow remote code execution."
}
```

- **Python Script** (`nmap_vuln_scanner.py`): The main script that runs Nmap scans, checks for vulnerabilities, and generates the output report.

How It Works

1. **Network Discovery**:
 - The tool first tries to determine the local network by running the `ip a` command.
 - Based on the user's input, it either scans a specific IP address or an entire network range.

2. **Scan Type Selection**:
 - The user selects which type of Nmap scan to run:
 - **Simple Scan**: Scans the most common ports (default).
 - **Enhanced Scan**: Scans all ports and detects the versions of services running on them.
 - **Aggressive Scan**: Performs a detailed scan, including service detection, OS detection, and script scanning.

3. **Service Detection**:
 - Nmap is used to scan the target, and the output is parsed to extract open ports and corresponding services (along with versions, if available).

4. **CVE Matching**:
 - After extracting services from the Nmap scan output, the tool checks each service against the `cve_vuln_data.json` file to find matching vulnerabilities.
 - For each match, it displays the CVE ID, description, and other details in the console.

5. **Reporting**:

- Both the Nmap scan results and the vulnerability information are saved to a text report file (`scan_report.txt`).

How to Use

1. **Ensure Prerequisites**:
 - Make sure that Python 3.x and Nmap are installed.
 - Place the `cve_vuln_data.json` file containing CVE data in the same directory as the Python script.

2. **Run the Script**:
 - Execute the Python script by running the following command:
 ```bash
 python nmap_vuln_scanner.py
 ```

3. **Interact with the Tool**:
 - The tool will prompt you to choose whether you want to scan a specific IP address or the entire local network.
 - You will then be prompted to select the scan type: Simple, Enhanced, or Aggressive.
 - The Nmap scan will run, and the detected services will be checked for matching CVEs.
 - Finally, the results will be displayed on the console and saved to `nmap_vuln_report.txt`.

Output

The output is saved in the file `scan_report.txt` and will include:

- Nmap scan results (ports and services).
- Detected vulnerabilities (CVE IDs and descriptions) related to the detected services.

Example Output in Report:
```
Nmap Scan Results:
--------------------
Host: 192.168.1.100
Ports:
 - 22/tcp Open OpenSSH 8.9p1
 - 80/tcp Open Apache 2.4.41
```

Vulnerabilities Detected:

CVE ID: CVE-2020-12345
Service: OpenSSH
Description: A vulnerability in OpenSSH could allow remote code execution.

CVE ID: CVE-2020-54321
Service: Apache
Description: An Apache HTTP server vulnerability allows for directory traversal.
```

#### **Customizing the Script**

- **CVE File**: To use your own vulnerability database, replace `cve_vuln_data.json` with your own file in the expected format.
- **Scan Options**: Modify the scan options in the script to include other Nmap features as needed (e.g., specific scripts, IP ranges).

#### **Important Notes**

- Ensure that you have the necessary permissions to run network scans on the target IP or network.
- The accuracy of the CVE matching depends on the data available in the `cve_vuln_data.json` file.
- The tool only performs basic CVE matching based on the service name and version. For more comprehensive vulnerability assessments, additional scanning and analysis may be required.

#### **Troubleshooting**

- **No CVE Data Found**: If no CVE data is displayed, ensure that your `cve_vuln_data.json` file is correctly formatted and contains the relevant data.
- **Nmap Scan Fails**: If Nmap cannot scan the target, ensure Nmap is installed and that the target IP address is reachable from your machine.

#### **Conclusion**

This tool provides a simple and efficient way to perform network vulnerability scanning using Nmap and match detected services against a list of known vulnerabilities. It helps
```

to identify potential risks in your network and generate reports for further analysis or action.