REVA UNIVERSITY
Bengaluru, India
REVA Academy for Corporate Excellence (RACE)

NETWORK SCAN TOOL

The Enhanced Network Scan Tool is a Python-based solution for administrators and network engineers who require automated scanning of devices within a network. This tool leverages the Nmap library to perform comprehensive scans, detect open ports, identify operating systems, and generate detailed reports. The results are organized into an HTML report that can be automatically emailed to a designated recipient.

This tool is ideal for auditing networks, identifying potential vulnerabilities, and monitoring devices across single hosts or IP ranges.

Features

The Enhanced Network Scan Tool offers the following key features, making it a powerful utility for network scanning and reporting:

Dual Scanning Modes

The tool provides two scanning modes to suit different use cases:

- **Single IP Scan**: Use this mode to scan a single device. This is particularly useful for troubleshooting or analyzing a specific machine on the network.
- **CIDR Range Scan**: This mode allows scanning of an entire subnet or range of devices, specified in CIDR notation (e.g., `192.168.1.0/24`).

Comprehensive Port and OS Detection

- The script performs a **TCP SYN scan** (`-sS`) to detect open ports in a stealthy manner.
- It also uses **OS fingerprinting** (`-O`) to identify the operating systems running on devices. This information is crucial for ensuring devices have up-to-date security patches.
- Service version detection (`-sV`) attempts to identify software versions running on open ports, which helps with vulnerability assessment.

Report Generation

- Once the scan is completed, the results are processed into a structured **HTML table**.
- The table includes columns for IP addresses, OS details, device status, open ports, and detected services.
- The HTML format is designed for readability, making it ideal for both technical and non-technical audiences.

Automated Email Reporting

- The tool supports automated email delivery of the scan report.
- Using SMTP, the script sends the generated HTML report to the recipient specified during execution.
- This feature ensures reports are shared promptly without requiring manual intervention.

Flexible Configuration

- The tool's SMTP settings, timeout thresholds, and scan arguments can be customized to meet specific requirements.
- This makes it adaptable to different network environments, email services, and security policies.

Error Handling

- The script includes mechanisms to handle common errors, such as missing SMTP credentials, invalid recipient addresses, or inaccessible networks.

System Requirements

To use the Enhanced Network Scan Tool, your system must meet the following requirements:

Operating System

The tool is compatible with:

- **Linux**: Preferred platform for running the script due to Nmap's seamless integration.
- **Windows**: Requires additional steps to install Nmap and configure Python paths.
- **macOS**: Supported but may require installing dependencies via Homebrew.

Python Environment

- **Python Version**: The script requires **Python 3.6** or later. Python 2.x is not supported.
- Ensure the Python installation includes `pip`, which is used to install additional libraries.

Nmap Tool

- The script uses the Nmap library for scanning. This requires the Nmap command-line utility to be installed on the system.
- Installation steps for Nmap are provided in the next section.

Required Python Libraries

The script depends on the following Python packages:

- **python-nmap**: Provides a Pythonic interface to Nmap. Install it using:

  ```
  pip install python-nmap
  ```

- **smtplib**: Used for sending emails via SMTP (built into Python).
- **email.mime**: Used for formatting email messages (built into Python).

Hardware Requirements

A basic system with at least 2 GB of RAM and a dual-core processor is sufficient for scanning small networks. Larger networks may require additional resources, especially for CIDR range scans.

Network Access

- Ensure the system running the script has access to the target network or devices.
- For best results, the system should be directly connected to the same subnet as the devices being scanned.

Installation and Setup

Follow these steps to install and set up the Enhanced Network Scan Tool:

Step 1: Install Python

Check if Python is already installed:

```
python3 --version
```

If not, download and install Python from https://www.python.org/.

Step 2: Install Nmap

- On **Linux**:

```
sudo apt update
sudo apt install nmap
```

- On **macOS**: Install Homebrew, then run:

```
brew install nmap
```

- On **Windows**: Download and install Nmap from https://nmap.org/download.html.

Step 3: Install Python Libraries

Install required libraries using `pip`:

```
pip install python-nmap
```

Step 4: Configure SMTP Settings

- Update the SMTP settings in the script. Replace the placeholder values with your actual email credentials:

```
SMTP_SERVER = "smtp.gmail.com"
SMTP_PORT = 587
SMTP_USERNAME = "your-email@gmail.com"
SMTP_PASSWORD = "your-email-password"
EMAIL_FROM = "your-email@gmail.com"
```

- If using Gmail, generate an **App Password** for enhanced security.

Configuration

Proper configuration is essential for ensuring the script runs smoothly. Below are the configurable parameters within the script:

SMTP Settings

The SMTP settings control how the email is sent. These parameters are located at the top of the script:

```
SMTP_SERVER = "smtp.gmail.com"  # SMTP server for Gmail
SMTP_PORT = 587                 # Port for TLS
SMTP_USERNAME = "your-email@gmail.com"
SMTP_PASSWORD = "your-app-password"
EMAIL_FROM = "your-email@gmail.com"
```

- **SMTP_SERVER**: The email service's SMTP server address. For Gmail, this is `smtp.gmail.com`.
- **SMTP_PORT**: The port used for encrypted email communication. Use port 587 for TLS.
- **SMTP_USERNAME**: Your email address used to send the report.
- **SMTP_PASSWORD**: The password for your email account. If using Gmail, this must be an **App Password**.

Timeout Settings

The `MAX_EXECUTION_TIME` parameter limits how long the script runs before timing out. This is useful for preventing the script from hanging indefinitely during large scans.

```
MAX_EXECUTION_TIME = 600  # 600 seconds (10 minutes)
```

Adjust this value based on the size of your network and the speed of your scan.

Script Workflow

The Enhanced Network Scan Tool operates in five distinct stages:

1. **User Input**: The user specifies the target network or device and provides an email address for receiving the report.
2. **Network Scan**: The script uses Nmap to perform a detailed scan of the specified range.
3. **Result Parsing**: The raw scan data is processed into structured information, including OS details, ports, and statuses.
4. **Report Generation**: A well-formatted HTML report is created using the parsed data.
5. **Email Delivery**: The report is sent via SMTP to the specified recipient.

Each of these stages is executed in sequence, with error handling mechanisms in place to address common issues (e.g., network errors, invalid inputs).

Functions in Detail

Below is a detailed explanation of each function used in the script.

`perform_scan(ip_range)`
This function performs the network scan using Nmap and returns the scan results. It takes one parameter:

> `ip_range` (string): The target IP address or CIDR range.

`parse_results(nm)`
This function processes the raw scan data returned by `perform_scan()`. It organizes the data into dictionaries containing:

- IP Address
- OS Details
- Open Ports

`generate_html_report(scan_data)`
This function converts the parsed data into an HTML table. The output is an HTML string that can be used in the email body.

`send_email(subject, html_content, recipient_email)`
This function sends the generated HTML report to the specified recipient. It uses the SMTP settings defined in the configuration section.

Sample Execution

The Enhanced Network Scan Tool is designed to be intuitive and interactive. Below is a step-by-step walkthrough of how to execute the script in real-world scenarios. This section demonstrates both the input prompts and expected outputs for each use case.

Running the Script

1. **Launch the Script**: Execute the script from the terminal or command prompt:
   ```
   python network_scan_tool.py
   ```
2. **Choose the Scan Mode**: The script will display the following options:
   ```
   Enhanced Network Scan
   1. Single IP Scan
   2. CIDR Range Scan
   Enter your choice (1 or 2):
   ```
   - **Option 1**: Enter 1 to scan a single IP address.
   - **Option 2**: Enter 2 to scan a range of devices using CIDR notation.
3. **Provide Scan Details**: Based on your choice:
   - If you selected **Single IP Scan**:
     ```
     Enter the IP address to scan: 192.168.1.1
     ```
   - If you selected **CIDR Range Scan**:
     ```
     Enter the CIDR range (e.g., 192.168.1.0/24): 192.168.1.0/24
     ```
4. **Provide Recipient Email**: Enter the email address where the HTML report should be sent:
   ```
   Enter the recipient email address: admin@example.com
   ```
5. **Execution and Output**:
   - The script performs the scan. Depending on the size of the range, this may take some time.
   - Once complete, you will see:
     ```
     Scanning IP range: 192.168.1.0/24
     Email sent successfully to admin@example.com
     ```

Expected Results

The report will be emailed as an HTML attachment to the provided address. Below is an example of the content in the email:

Subject: Enhanced Network Scan Report

Body:

- Includes a summary of the scan (e.g., total devices scanned, errors encountered).
- The HTML table is embedded or attached for detailed analysis.

Example Console Output:

```
Scanning IP range: 192.168.1.0/24
Processing results...
Generating HTML report...
Sending report to admin@example.com...
Email sent successfully.
```

HTML Report Example

The HTML report generated by the script is a well-structured, tabular document designed for readability. Below is an example of how the report is formatted.

Report Overview

The report includes the following details for each scanned device:

- **IP Address**: The device's IP address.
- **OS Details**: Detected operating system or "Unknown" if not identified.
- **Device Status**: The device state (e.g., "up" or "down").
- **Firewall Info**: Currently "None" (reserved for future updates).
- **Ports**: List of detected open TCP ports and their states.

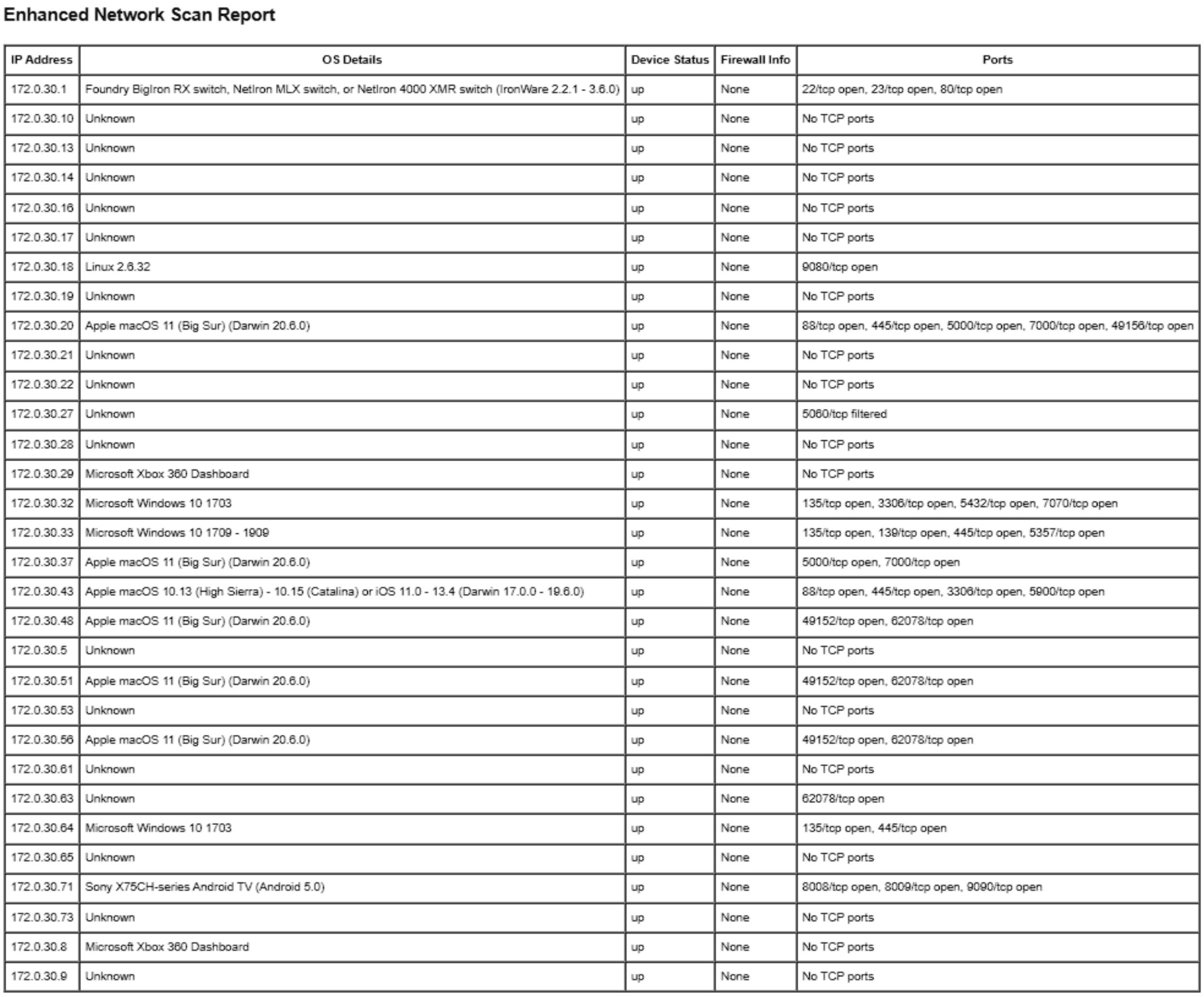
Sample HTML Code

```
<html>
    <body>
        <h2>Enhanced Network Scan Report</h2>
        <table border="1" cellpadding="5" cellspacing="0">
            <thead>
                <tr>
                    <th>IP Address</th>
                    <th>OS Details</th>
                    <th>Device Status</th>
                    <th>Firewall Info</th>
                    <th>Ports</th>
                </tr>
            </thead>
            <tbody>
                <tr>
                    <td>192.168.1.1</td>
                    <td>Linux 3.10 - 3.16</td>
                    <td>up</td>
                    <td>None</td>
                    <td>22/tcp open, 80/tcp open</td>
                </tr>
                <tr>
                    <td>192.168.1.2</td>
                    <td>Windows 10</td>
                    <td>up</td>
                    <td>None</td>
                    <td>3389/tcp open, 135/tcp open</td>
                </tr>
            </tbody>
        </table>
    </body>
</html>
```

Rendered Output

IP Address	OS Details	Device Status	Firewall Info	Ports
192.168.1.1	Linux 3.10 - 3.16	up	None	22/tcp open, 80/tcp open
192.168.1.2	Windows 10	up	None	3389/tcp open, 135/tcp open

This HTML table format allows administrators to quickly review key information about their network.



Enhanced Network Scan Report

IP Address	OS Details	Device Status	Firewall Info	Ports
172.0.30.1	Foundry BigIron RX switch, NetIron MLX switch, or NetIron 4000 XMR switch (IronWare 2.2.1 - 3.6.0)	up	None	22/tcp open, 23/tcp open, 80/tcp open
172.0.30.10	Unknown	up	None	No TCP ports
172.0.30.13	Unknown	up	None	No TCP ports
172.0.30.14	Unknown	up	None	No TCP ports
172.0.30.16	Unknown	up	None	No TCP ports
172.0.30.17	Unknown	up	None	No TCP ports
172.0.30.18	Linux 2.6.32	up	None	9080/tcp open
172.0.30.19	Unknown	up	None	No TCP ports
172.0.30.20	Apple macOS 11 (Big Sur) (Darwin 20.6.0)	up	None	88/tcp open, 445/tcp open, 5000/tcp open, 7000/tcp open, 49156/tcp open
172.0.30.21	Unknown	up	None	No TCP ports
172.0.30.22	Unknown	up	None	No TCP ports
172.0.30.27	Unknown	up	None	5060/tcp filtered
172.0.30.28	Unknown	up	None	No TCP ports
172.0.30.29	Microsoft Xbox 360 Dashboard	up	None	No TCP ports
172.0.30.32	Microsoft Windows 10 1703	up	None	135/tcp open, 3306/tcp open, 5432/tcp open, 7070/tcp open
172.0.30.33	Microsoft Windows 10 1709 - 1909	up	None	135/tcp open, 139/tcp open, 445/tcp open, 5357/tcp open
172.0.30.37	Apple macOS 11 (Big Sur) (Darwin 20.6.0)	up	None	5000/tcp open, 7000/tcp open
172.0.30.43	Apple macOS 10.13 (High Sierra) - 10.15 (Catalina) or iOS 11.0 - 13.4 (Darwin 17.0.0 - 19.6.0)	up	None	88/tcp open, 445/tcp open, 3306/tcp open, 5900/tcp open
172.0.30.48	Apple macOS 11 (Big Sur) (Darwin 20.6.0)	up	None	49152/tcp open, 62078/tcp open
172.0.30.5	Unknown	up	None	No TCP ports
172.0.30.51	Apple macOS 11 (Big Sur) (Darwin 20.6.0)	up	None	49152/tcp open, 62078/tcp open
172.0.30.53	Unknown	up	None	No TCP ports
172.0.30.56	Apple macOS 11 (Big Sur) (Darwin 20.6.0)	up	None	49152/tcp open, 62078/tcp open
172.0.30.61	Unknown	up	None	No TCP ports
172.0.30.63	Unknown	up	None	62078/tcp open
172.0.30.64	Microsoft Windows 10 1703	up	None	135/tcp open, 445/tcp open
172.0.30.65	Unknown	up	None	No TCP ports
172.0.30.71	Sony X75CH-series Android TV (Android 5.0)	up	None	8008/tcp open, 8009/tcp open, 9090/tcp open
172.0.30.73	Unknown	up	None	No TCP ports
172.0.30.8	Microsoft Xbox 360 Dashboard	up	None	No TCP ports
172.0.30.9	Unknown	up	None	No TCP ports

Figure: Scan network tool result

Troubleshooting and Error Handling

The Enhanced Network Scan Tool is equipped to handle various common errors. Below is a comprehensive list of possible issues, their causes, and recommended solutions.

Error: `nmap.PortScannerError`

- **Cause**: Nmap is not installed or cannot be found in the system's PATH.
- **Solution**:
  Verify Nmap installation:
  ```
  nmap --version
  ```
  Ensure Nmap is properly added to the PATH (Windows users may need to add the installation directory to PATH).

Error: `KeyError: 'tcp'`

- **Cause**: The scanned device has no open TCP ports.
- **Solution**:
  - Ensure the target device is accessible and that the firewall allows scanning.
  - Check if the device has any active services running.

Error: `SMTPAuthenticationError`

- **Cause**: Incorrect email credentials or App Password configuration.
- **Solution**:
  - Double-check `SMTP_USERNAME` and `SMTP_PASSWORD`.
  - For Gmail users, ensure that **App Passwords** are enabled and used instead of the account password.

Error: `Scan Taking Too Long`

- **Cause**: Large IP range or slow network response.
- **Solution**:
  - Use a smaller range for testing purposes.
  - Adjust scan timing parameters (e.g., `-T4` can be changed to `-T3` or `-T5`).

Email Not Delivered

- **Cause**: Incorrect recipient email address or SMTP server issues.
- **Solution**:
  - Verify the recipient email address.
  - Test the SMTP configuration using a standalone tool or script.

Security Considerations

Using the Enhanced Network Scan Tool requires attention to the following security practices:

Permission

Scanning unauthorized networks is illegal and unethical. Ensure you have explicit permission to scan the target network or devices.

SMTP Security

- Always use secure email credentials. For Gmail, enable **App Passwords** and avoid sharing them publicly.
- Consider encrypting email communications using **TLS**.

Data Handling

The generated reports may contain sensitive information about the network. Store them securely and delete outdated reports to prevent misuse.

Scanning Best Practices

- Use stealth scan modes (`-sS`) to minimize the impact on production networks.
- Avoid scanning during peak hours to prevent network congestion.