Created by Yuvin Perera

Nessus Script Documentation

Purpose of the Scripts

Nessus is a widely used vulnerability assessment tool for identifying security weaknesses in systems. The provided scripts automate tasks such as starting a new scan, listing existing scans, and downloading scan results. These scripts simplify the interaction with the Nessus API, enabling security teams to conduct assessments efficiently and retrieve actionable results.

The first script, Start a New Scan, initiates a vulnerability scan for a specified target. The second script, List All Scans, retrieves and displays all existing scans in Nessus. The third script, Download Scan Results, automates the process of exporting scan results to a file, such as a PDF report.

Usage Instructions

General Setup

1. Ensure Nessus is installed and running. You can download it from the official Nessus website.
2. Obtain an API key or access token for authenticating with the Nessus API.
3. Install Python and the requests library if not already installed:
4. pip install requests
5. Save each script as a .py file and run them from the command line or any Python IDE.

Script 1: Start a New Scan

Command:

```
python start_nessus_scan.py
```

Required Parameters:

- ✓ scan_name: The name for the new scan (e.g., Weekly Scan).
- ✓ target_ip: The IP address or range of the target (e.g., 192.168.1.1).
- ✓ api_key: Your Nessus API key.

Script 2: List All Scans

Command:

```
python list_nessus_scans.py
```

Required Parameters:

- ✓ api_key: Your Nessus API key.

Script 3: Download Scan Results

Command:

```
python download_scan_results.py
```

Required Parameters:

- scan_id: The ID of the scan to download.
- api_key: Your Nessus API key.

Expected Outputs or Results

Start a New Scan

- A new Nessus scan is created and started.
- The scan ID is displayed in the terminal.

List All Scans

- Displays a list of all existing scans, including their names, IDs, and statuses.

Download Scan Results

- Downloads the scan report for the specified scan ID and saves it as a PDF file.
- Outputs a confirmation message with the file name.

Dependencies Needed

- Nessus Server: Required for running and managing vulnerability scans.
- Python: Required to run the scripts.
- Requests Library: For making HTTP API calls to Nessus. Install it using:
- pip install requests

Line-by-Line Explanation

Start a New Scan

```
import requests
```

- requests: Used for interacting with the Nessus API.

```
def start_nessus_scan(scan_name, target_ip, api_key):
```

- Defines a function to start a new Nessus scan with the provided parameters.

```
headers = {"X-ApiKeys": f"accessKey={api_key}"}
```

- Sets up the headers for API authentication using the provided API key.

```
data = {
    "uuid": "basic",  # UUID for a basic scan template
    "settings": {
        "name": scan_name,
        "text_targets": target_ip,
        "launch_now": True
    }
}
```

- Constructs the payload for creating a new scan. The uuid specifies the scan template.

```
response = requests.post("https://nessus-server:8834/scans", headers=headers, json=data)
print(f"Started Nessus scan: {response.json().get('scan')}")
```

- Sends a POST request to the Nessus API to create the scan and displays the scan ID.

List All Scans

```
import requests
```

- requests: Used for interacting with the Nessus API.

```
def list_nessus_scans(api_key):
```

- Defines a function to retrieve and list all scans.

```
headers = {"X-ApiKeys": f"accessKey={api_key}"}
response = requests.get("https://nessus-server:8834/scans", headers=headers)
scans = response.json().get('scans', [])
```

- Sets up headers for authentication and sends a GET request to retrieve all scans.

```
for scan in scans:
    print(f"ID: {scan['id']}, Name: {scan['name']}, Status: {scan['status']}")
```

- Iterates through the list of scans and prints their details.

Download Scan Results

```
import requests
```

- requests: Used for interacting with the Nessus API.

```
def download_scan_results(scan_id, api_key):
```

- Defines a function to download scan results for a specified scan ID.

```
headers = {"X-ApiKeys": f"accessKey={api_key}"}
export_url = f"https://nessus-server:8834/scans/{scan_id}/export"
```

- Sets up headers for authentication and constructs the API endpoint for exporting scan results.

```
export_response = requests.post(export_url, headers=headers, json={"format": "pdf"})
file_id = export_response.json().get("file")
```

- Sends a POST request to initiate the export and retrieves the file ID.

```
download_url = f"{export_url}/{file_id}/download"
download_response = requests.get(download_url, headers=headers)
```

- Constructs the download URL and retrieves the scan report file.

```
with open(f"nessus_scan_{scan_id}.pdf", "wb") as file:
    file.write(download_response.content)
```

- Saves the downloaded report to a file named with the scan ID.

```
print(f"Report downloaded: nessus_scan_{scan_id}.pdf")
```

- Confirms that the report has been successfully downloaded.

Why These Scripts Are Needed ?

Automating Nessus tasks reduces the time and effort required to manage vulnerability scans, allowing security teams to focus on analysis and remediation. The Start a New Scan script simplifies the process of launching scans, ensuring consistent configurations. The List All Scans script provides an overview of ongoing and completed scans, aiding in progress tracking and reporting. The Download Scan Results script automates the retrieval of detailed reports, making it easier to share findings with stakeholders and integrate results into security workflows.