Cisco STIG Scanner v4 (RC1) Documentation

Table of Contents

Overview

The `cisco_stig_scanner_v4_rc1.py` script is designed to automate STIG (Security Technical Implementation Guide) compliance checks for Cisco devices running IOS XE and NX OS. The script generates or updates CKL (Checklist) files based on the vulnerability checks it performs.

Features

- **Multi-Device Support**: Targets multiple types of devices including IOS XE Switches, Routers, and NX OS devices.
- **Authentication Methods**: Supports both 2-Factor Authentication (2FA) and username/password.
- **Dynamic CKL Management**: Uses the CKL files to dynamically determine which vulnerability functions to call.
- **Selective Scanning**: Allows running selective vulnerability checks by specifying them in `stig_vul.csv`.
- **Output**: Findings are saved in both CKL and CSV formats in the directory where the script resides.

Requirements

- **SecureCRT Version**: 9.1.1 (Optimized for this version)
- **Python Version**: Python 3.9.7
- **Operating System**: Windows 2019 Server

Usage

Setting Up

1. **SecureCRT Configuration**: Make sure SecureCRT is installed and running version 9.1.1.
2. **Python Environment**: Ensure Python 3.9.7 is installed.

Running the Script

1. Open SecureCRT and go to `Scripts -> Run`.
2. Choose `cisco_stig_scanner_v4_rc1.py`.
3. **Authentication**:
   - For `un` in `host.csv`, you'll be prompted for username and password.
   - For 2FA, you'll be prompted for your PIN once, and it will be used for subsequent 2FA hosts.
4. **Skipping Hosts**: Any host line in `host.csv` starting with "#" will be skipped.

File Structure

Your directory should contain these baseline files:

'''
cisco_stig_scanner_v4_rc1.py
NXOS-NDM-L2S-v2_r5_r2.ckl
XE_Router-NDM-RTR-v2_r7_r8.ckl
XE_Switch-NDM-L2S-RTR-v2_r6_r4_r4.ckl
XE_Switch-NDM-L2S-v2_r6_r4.ckl
host.csv
stig_vul.csv
'''

Example host.csv

'''
skip,host,auth,ckl
,# this is a comment line,
,192.168.1.1,un,NXOS-NDM-L2S-v2_r5_r2.ckl
,192.168.1.2,2FA,XE_Router-NDM-RTR-v2_r7_r8.ckl
'''

CKL Naming Convention

- **Device Type**: Specifies the type of device the CKL file is intended for. Examples include `XE_Router` for Cisco IOS XE Routers, `XE_Switch` for Cisco IOS XE Switches, and `NXOS` for Cisco NX-OS devices.

- **STIG Types**: This part of the name indicates which STIGs are being applied or checked against. For example, `NDM-RTR` would indicate that the Network Device Management (NDM) and Router (RTR) STIGs are applicable.
- **Version and Release**: This section provides version and release information for the STIGs being applied. For example, `v2_r7_r8` would indicate:
  - NDM is at Version 2, Release 7
  - RTR is at Version 2, Release 8

Example

For a Cisco IOS XE Router involving NDM and RTR STIGs, the name could be updated as follows:

- **Old Name**: `Stigtemplate-XE-Router-NDM-RTR-v2r7_07_jun_2023.ckl`
- **New Name**: `XE_Router-NDM-RTR-v2_r7_r8.ckl`

Environment Details

- **SecureCRT Version**: 9.1.1
- **Python Version**: 3.9.7
- **Operating System**: Windows 2019 Server

History

This script is a fork of several scripts, including but not limited to:

- `autostig-XE-Switch-NDM-L2S-RTR-v3r5_2FA_TEST_1.py`
- `autostig-XE-Switch-NDM-L2S-v3r5_2FA_TEST_12_J_3_A.py`
- `autostig-XE-Router-NDM-RTR-v3r6_2FA.py`
- `autostig-NXOS-L2S-NDM-v3r4.py`

Python Packages Used

Below is an explanation of the Python packages and modules imported in the script:

Standard Library Modules

- `os`: Provides a way to interact with the operating system, such as file and directory operations.
- `datetime`: Used for manipulating dates and times.
- `array`: Provides a data structure for efficient arrays.
- `sys`: System-specific parameters and functions.
- `re`: Regular expressions library.
- `string`: Provides string constants and utility functions.
- `csv`: Reader and writer for CSV files.

- `inspect`: Provides several useful functions to help get information about live objects like modules, classes, methods, etc.
- `time`: Provides time-related functions.
- `xml.sax.saxutils`: Provides support for parsing XML documents using SAX.
- `collections.OrderedDict`: Dictionary subclass that remembers the order entries were added.
- `xml.etree.ElementTree (ET)`: Provides an API for parsing and creating XML data.

Third-Party Modules

- `SecureCRT`: Provides access to SecureCRT-specific functionality, such as session management.
- `packaging.version`: Provides support for comparing package versions.

Additional Notes

- `from datetime import date`: Imports only the `date` class from the `datetime` module for date manipulations.