AMP4e-to-FMC-Host-Input-Script

NOTE: All Cisco software is subject to the Supplemental End User License Agreements (SEULA) located at https://www.cisco.com/c/en/us/about/legal/cloud-and-software/software-terms.html

Summary
This script imports all the operating systems (OS) information and vulnerable software detections from AMP for Endpoints (A4E) console using the A4E API, prints the output to a CSV and then imports the CSV into Firepower Management Center (FMC) using the Host Input API of FMC.
AMP for Endpoints (A4E) keeps a track of OS info and looks up for the CVEs associated with several commonly seen applications on the Endpoints, and reports them on its console as a proactive security feature called "Vulnerable Software".
Firepower Management Center (FMC) builds and maintains "Host Profiles", based on all the information it learns directly or indirectly about the hosts on the network. Based on these Host Profiles, FMC provides better tailored IPS Signature recommendations - which ones to enable/disable.

List of the files available inside the package

For AMP for Endpoints Communications

File Name	Purpose
A4E_to_FMC.py	main script that will retrieve information from A4E and push it into FMC
parameters.json	stores the hostnames, credentials and any other custom parameters
sample - parameters.json	sample file for parameters.json for reference only
amp_api.py	contains supporting functions to interact with A4E

For Host Input (FMC) Communications

Note: These files are of 'FMC Host Input API SDK' as-it-is downloaded from CCO without any modifications.

- sf_host_input_agent.pl
- SFCheckPreReq.pm
- SFHIclient.pm
- SFHIlog.pm
- SFHostInputAgent.pm
- SFPkcs12.pm
- InputPlugins\csv.pm

Prerequisites

This tool uses **python** to establish communications with AMP for Endpoints and pull the vulnerability information for the provided Group Name in AMP for Endpoints.
Then the tool uses **perl** to establish communications with FMC to add vulnerability data collected earlier to the Host profile in FMC.

In order for the tool to work properly the following requirements must be met:

1- System - This script can to be executed on any host running Windows / Any POSIX kernel based Linux OS - tested on Ubuntu. (Not tested on Mac – it may work). This script WILL NOT WORK if directly executed on FMC
2- Account on AMP for Endpoints Public cloud and an API key generated. Read-only API key is fine. This may not work with AMP Private Cloud.

3- **Firepower Management Center (FMC) 5.4+**
4- Requires both Python and Perl - **Python 2.7+ or Python 3.6+, AND Perl 5.** This script is written in Python, so Python is required; whereas this script leverages the FMC Host Input SDK, which is written in Perl, so Perl is required
5- **Python requests module**
   `$python -m pip install requests`
6- **Update the `'parameters.json'` FILE WITH THE DETAILS BEFORE EXECUTING THIS SCRIPT**
7- **TCP port 443** to AMP for Endpoints API FQDN
8- **TCP port 8307** to FMC
9- **FMC Host Input API client certificate file** (`xxxxxx.pkcs12`) generated from FMC, downloaded in this script's local directory.
   To generate the certificate, login to FMC Web GUI and navigate to System -> Integrations -> Host Input Client -> Create Client -> give the IP address of your HOST and DO NOT GIVE a password -> Save. Download the pkcs12 file in this script's local directory.
   Note that you DO NOT have specify the name of the pkcs12 certificate file anywhere in the `parameters.json` file. The file is picked up automatically by the FMC Host Input SDK. Hence ensure that there is ONLY ONE pkcs12 file in the local directory

Usage

The script is fed by one user configurable file: "parameters.json" These are the variables used to define the details of the Security Center and FMC that will be used.
Modify the file as follows:

```
# AFTER UPDATING THIS FILE, REMOVE ALL THE COMMENTS, INCLUDING THIS LINE
# YOU MAY LOOK AT 'sample - parameters.json' FILE FOR EXAMPLE
{
    "A4E_client_id": "<from amp for endpoints>",
    "A4E_api_key": "<from amp for endpoints>",
    "A4E_API_hostname": "api.amp.cisco.com",                        # <change it to US/EU/APJ AMP cloud as applicable -
                                                # "api.amp.cisco.com" / "api.eu.amp.cisco.com" / "api.apjc.amp.cisco.com"
    "A4E_group_names": ["<AMP Group Name 1>","<AMP Group Name 2>"],  # <list of AMP for Endpoints groups to be queried>.
                                                # The RECOMMENDED GROUPS are endpoints having static IP addresses like Server
                                                # segments or desktops with static addresses
    "FMC_ipaddress": "<FMC ipaddress>",                             # <DO NOT PROVIDE A HOSTNAME. JUST PROVIDE AN IP ADDRESS>
    "FMC_host_vuln_db_overwrite_OR_update": "update",               # "update" OR "overwrite", RECOMMENDED is "update"
    "push_changes_to_fmc": true                                     # true or false (without quotes)
}
```

NOTE: Keep the above variable as UPDATE only. Use OVERWRITE only if you know its implications or if guided by TAC.

Example configuration file:

```
{
    "A4E_client_id": "d93938ab27609ba060bc",
    "A4E_api_key": "7f4a04df-dfec-4067-8186-c8c2ab5223ae",
    "A4E_API_hostname": "api.amp.cisco.com",
    "A4E_group_names": ["MANGO Protect","MANGO Servers","MANGO Domain Controller"],
    "FMC_ipaddress": "10.20.60.13",
    "FMC_host_vuln_db_overwrite_OR_update": "update",
    "push_changes_to_fmc": true
}
```

Running this script

Before running the following command ensure that all prerequisites are met and run it in the same directory where you have the script loaded.
To run the tool simply execute:
`python A4E_to_FMC.py`

All the activities of the script are logged to a file called **AUDIT.log** file. The file is not over-written, only appended. This includes all INFO and ERROR messages. Only SHOWSTOPPER errors, if any, are also displayed on the screen.