Firewall Rules Inducer Tool

Standard Operating Procedure

Department: NIS
SOP ID: FRIT-01
Date: 01/Dec/2020

Overview

Firewall Rules Inducer Tool (FRIT) is a tool to create Firewall Rules on Cisco FMC using spreadsheet. Either rules can be created one by one from the spreadsheet or in a bulk to a maximum of 1000 rules in one run. Not only the rules in Cisco FMC get created, the same rules will get deployed to Cisco FTD's connected as well. This SOP is a Developer's guide to build and maintain the FRIT.

Definitions

SOP	*Standard Operating Procedure*
python	*General Purpose Interpreted Programming language*
.exe	*Windows Executable*
spreadsheet	*MS Excel file*
.yaml	*Yaml or yml file*

Assumptions & Prerequisite Skills

- This SOP assumes that you know how to program in python.
- Use command line interpreter (cmd)
- Use spreadsheets (edit, modify, maintain)
- REST API's of Cisco FMC

Supplies Needed

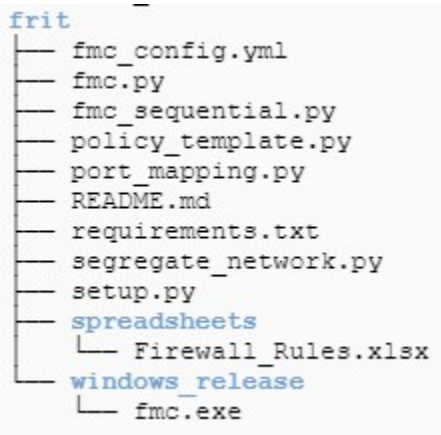
- Windows System
- Python3 and pip3 installed
- Excel spreadsheet with all the require firewall rules to be created
- Code Editor Eg: Atom, PyCharm, Notepad++

Overview of Steps

Step 1 - Code Overview
Step 2 - Project Setup
Step 3 - Create the Windows Executables
Step 4 - Commands to Run the Tool
Step 5 - Inputs and Outputs

Step 1 - Code Overview

1. **Structure of the codebase:**



2. **Codebase details of various files**

Sl/No.	File Name	Details
1	/frit	Main folder of the tool shipped as zip
2	README.md	Read Me help file explaining how to run the code
3	spreadsheets/	Folder which contains the Spreadsheets of Firewall Rules
4	Firewall_Rules.xlsx	Spreadsheet containing all the required Firewall Rules
5	fmc_sequential.py	Code to create single rule one by one from the spreadsheet
6	requirements.txt	List of python packages to be installed for preparing the windows executable
7	fmc.py	Code to create bulk rules of maximum 1000 from the spreadsheet
8	policy_template.py	File which contains policy payload class imported in both fmc.py and fmc_bulk.py
9	port_mapping.py	File which maps various ports to corresponding protocol number and port number imported in both fmc.py and fmc_bulk.py
10	segregate_network.py	File which splits the networks based on Host, Network and Range; along with segregating zones
11	setup.py	File used to generate windows executable
12	windows_release	Folder which consists of all the supporting executable libraries
13	fmc.exe	Executable created based on the required release
14	fmc_config.yml	Config file to pass the parameters

Step 2 - Project Setup

1. **To Prepare the Environment**

   Run:

   ```
   pip3 install -r requirements.txt
   ```

2. **List of open source python packages used:**

Sl/No.	Package Name	Details
1	tabulate	Package used to display the firewall rules details from the spreadsheet
2	requests	Package used to query Cisco FMC via REST API's
3	pandas and xlrd	Package used to read excel spreadsheet

Step 3 - Create the Windows Executable

1. **To Create Windows Executable**

   Run:

   ```
   pip3 install py2exe
   ```

   ```
   python3 setup.py py2exe
   ```

   `fmc.exe` file will be present inside the folder `dist`

   Rename `dist` folder to `windows_release`

   Create a zip file to ship `windows_release.zip`

2. **For Sequential windows executable repeat step #1 with** `fmc_sequential.py` **file name inside** `setup.py`

Step 4 - Commands to Execute the Tool

1. **For single rules creation (one by one)**

   Run within windows_release folder:

   ```
   fmc_sequential.exe <host_ip> <username> <password> <spreadsheet> <sheet_name>
<start range> <end range>
   ```

   Eg:

   ```
fmc_sequential.exe 10.88.88.80 apiuser P@ssw0rd Firewall_Rules.xlsx Automation 1 25
   ```

2. **For Bulk rules creation**

   Run within windows_release folder:

   ```
   fmc.exe --config_file fmc_config.yml
   ```

   Eg:

   ```
   fmc.exe --config_file fmc_config.yml
   ```

3. **Config Details:**

Sl/No.	Input Parameters	Details
1	Host IP Address	IP Address of the Cisco FMC
2	Username	User Credentials of the Cisco FMC used which can be non admin user as well, which is used by the REST API's
3	Password	Password used to by the REST API's
4	Spreadsheet Path	File and File path to the Spreadsheet with Firewall Rules
5	Sheet Name	Name of the sheet inside the Spreadsheet file
6	Start Range	Starting serial number of the rules to begin with
7	End Range	Ending serial number of the riles to end with, if same as start then that particular rule is executed Eg: 4 4

Step 5 - Inputs and Outputs

1. **Spreadsheet should contain all the Firewall Rules**
2. **Spreadsheet columns' details:**

Sl/No.	Column Names	Details
1	Sl/No.	Serial Number of the Firewall Rule acts as a pointer for the range during input
2	Name	Firewall Rule Name
3	Source Network	Source of the network, either host, range or network with subnet
4	Destination Network	Destination of the network, either host, range or network with subnet
5	Source Port	Port number of the source firewall object
6	Destination Port	Port number of the destination firewall object
7	Source Zone	Zone name of the source firewall object
8	Destination Zone	Zone name of the destination firewall object
9	Domain	Firewall Domain name
10	ACP	Name of the Access Control Policy used in the Rule
11	IPS	Name of the IPS Policy used in the Rule

3. **Sample fmc config yaml file:**

```yaml
---
ftd_deploy: yes
fmc_device:
  host_ip: 10.88.88.80
  username: apiuser
  password: P@ssw0rd123
excel:
  sheet_name: BUILD-NGFW
  file_path: ./spreadsheets/Firewall_Policies.xlsx
  rows_range:  #  max range limit is 1000
    start: 5
    end: 50
```

Example of the execution output:

```
$ ./windows_release/fmc.exe 10.120.68.157 apiuser passwrd spreadsheets/GNOC5.xlsx Automation 1 1
+-------+-----------+----------------+------------------+--------------------+
|   S/N | Name      | Source_Zone    | Destination_Zone | Destination_Port   |
|-------+-----------+----------------+------------------+--------------------|
|     1 | mig-00001 | NIS_Peering_INZ | any              | icmp               |
+-------+-----------+----------------+------------------+--------------------+
Enter ok to create the above Firewall Rules: ok
'Bulk rules are successfully created'


Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: Deploying
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: Deploying
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: Deploying
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED
Deployment status for 1cdec8f0-0952-11df-8ab4-c0c0ed24bf35: PARTIALLY_SUCCEEDED

Global/CCIS-GNOC5/G5-Internal-DMZ-NGFW successfully deployed the firewall devices

Please verify the firewall access rules and device deployment status on FMC Web GUI
```

Conclusion

Either rules can be created one by one from the spreadsheet or in a bulk of 1000 from the spreadsheet using the Firewall Rules Inducer Tool (FRIT). The same Rules which get created in Cisco FMC will also get deployed to connected Cisco FTD's by FRIT.

Caveats

1. Firewall Rules Inducer Tool (FRIT) cannot read password protected spreadsheets, so before running the tool the input spreadsheets must be unprotected.
2. Avoid unnecessary Rows or Columns in the input spreadsheet, this can harm the code during the execution
3. A separate sheet within the spreadsheet should be used for miscellaneous details.
4. Don't delete the supporting binary files generated inside windows_release/ folder, these files are used by .exe

Revision History

Date	Version	Description	Approved
01/Dec/2020	1.0.0	Initial document created	
28/Dec/2020	1.0.1	Making deployment optional and passing arguments via yaml file	