FORCEPOINT

# Forcepoint NGFW and Azure Sentinel

## Integration Guide

Tom Meaney
Mattia Maggioli
23 March 2020
Public

**Summary** ..... 2

**Caveats** ..... 2

**Implementation** ..... 3

**Step 1 – Set up Azure Sentinel integration** ..... 4

**Step 2 – Configure SMC to allow connections from API clients** ..... 6

**Step 3 – Creating custom log filters from SMC** ..... 6

Example of common log queries ..... 8

Adding extra filters ..... 9

Removing extra filters ..... 9

**Step 4 – Configuration and installation of the SMC2CLOUD service** ..... 10

**Appendix A – Configuration parameters** ..... 11

**Appendix B – Create a Workbook into Azure Sentinel** ..... 11

**Troubleshooting** ..... 15

| Version | Date | Author | Notes |
|---|---|---|---|
| **0.1** | 12 December 2019 | Tom Meaney | First draft |
| **0.2** | 12 December 2019 | Mattia Maggioli | Review |
| **0.3** | 18 December 2019 | Tom Meaney | Update |
| **0.4** | 18 December 2019 | Mattia Maggioli | Review |
| **0.5** | 23 March 2020 | Neelima Rai | Added troubleshooting chapter |

# Summary

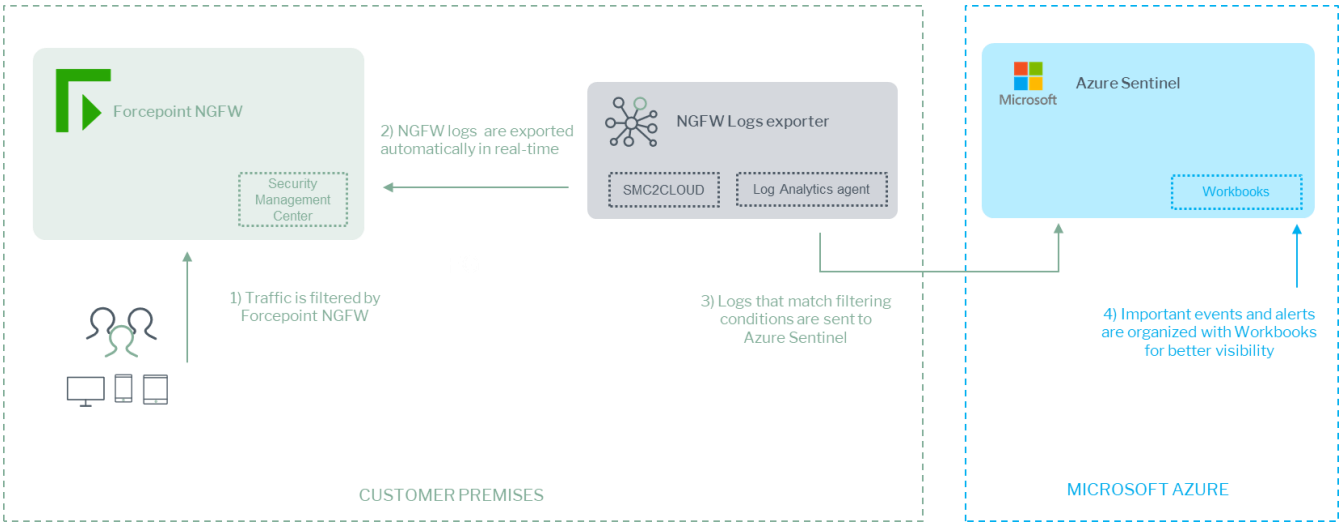
This guide provides step by step instructions to integrate Forcepoint Next Generation Firewall (Forcepoint NGFW) with Azure Sentinel to export pertinent log data from the NGFW according to user-configured filters.

The code and instructions provided enable system administrators to automatically

- Export log events from NGFW into Azure Sentinel in real-time
- Ingest logs into Azure Sentinel log analytics and visualize relevant events using Workbooks

This integration enriches visibility into user activities recorded by NGFW, enables further correlation with data from Azure workloads and other feeds, and improves monitoring capability with Workbooks inside Azure Sentinel.

A description of the workflow between the components involved in this POC is depicted in this diagram:




## Caveats

These implementation instructions are tested with the following product versions

- Forcepoint NGFW 6.5.2
- Forcepoint NGFW Security Management Center (SMC) 6.6.0

The following activities are out of the scope of this document and therefore left to the system administrator, as part of ordinary maintenance procedures to be put in place within the existing infrastructure:

- Configuration of appropriate hygiene procedures to handle logs produced during any step of the solution workflow
- monitoring of the scripts, services and applications involved in the solution

# Implementation

The solution described in this chapter requires the following files available at this link:
https://frcpnt.com/ngfw-sentinel-latest

- fp-ngfw-exporter-cloud-v1.tar.gz

The archive **fp-ngfw-exporter-cloud-v1.tar.gz** contains all files necessary to setup and run the **SMC2CLOUD** service which automatically queries, processes and uploads logs to Azure. We suggest deploying this service on an Ubuntu 18.04 machine, the instructions provided in this document are based on this operating system and the following packages

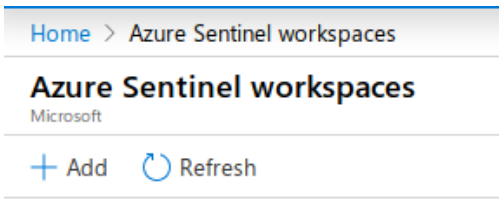
- Python3.x

The software packages and related dependencies are automatically installed by the **install.sh** script provided inside the **fp-ngfw-exporter-cloud-v1.tar.gz** file, which will execute the following commands as part of the deployment script:

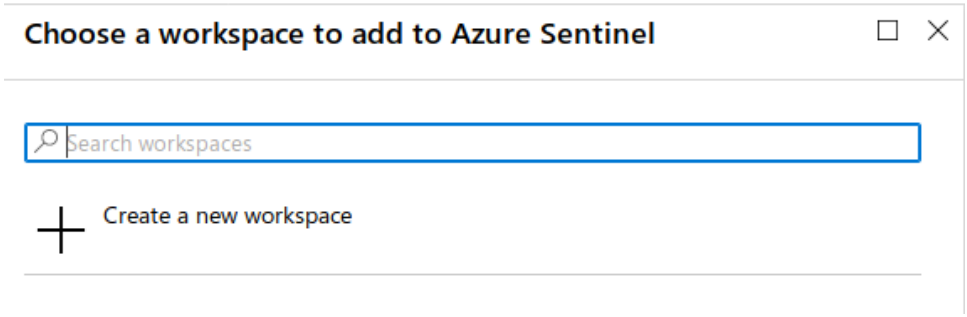
```
python3 get-pip.py
python3 -m pip install --user virtualenv
python3 -m venv venv
source venv/bin/activate
python3 -m pip install -r requirements.txt
mkdir /opt/ngfw_2_cloud
cp -r ./* /opt/ngfw_2_cloud
cp /opt/ngfw_2_cloud/SMC2CLOUD.service /lib/systemd/system/SMC2CLOUD.service
systemctl daemon-reload
systemctl enable SMC2CLOUD
systemctl start SMC2CLOUD
```

## Step 1 – Set up Azure Sentinel integration

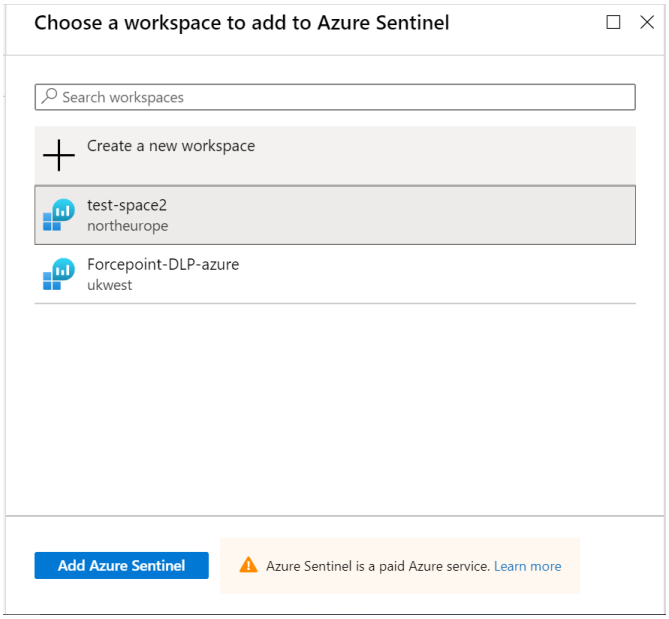
1. Sign into Azure portal
2. Click on **All services**, select **Azure Sentinel** click on it
3. Click on **Add**



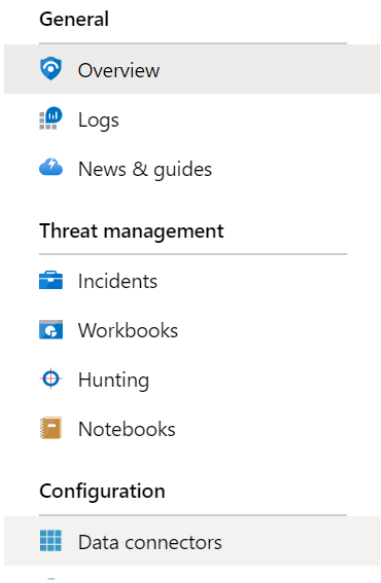

4. Click on **Create a new workspace**



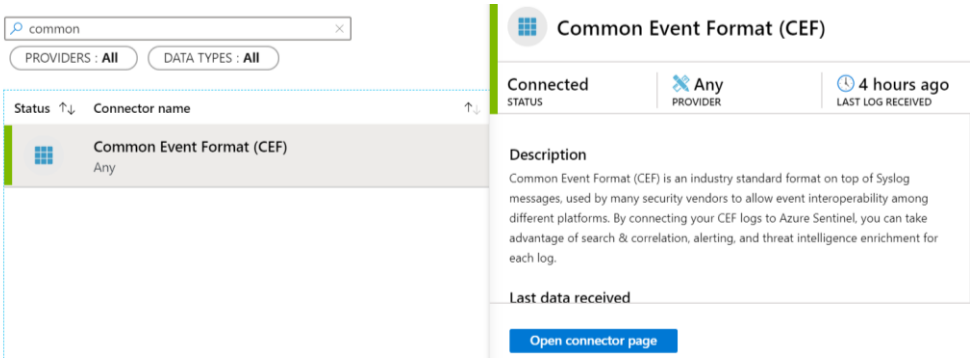

5. Give a name to this workspace, select the subscription type, the resource group (if none exists create a new one) and select the location where this workspace will be hosted. Wait a minute or so for the validation and deployment from Azure to complete.



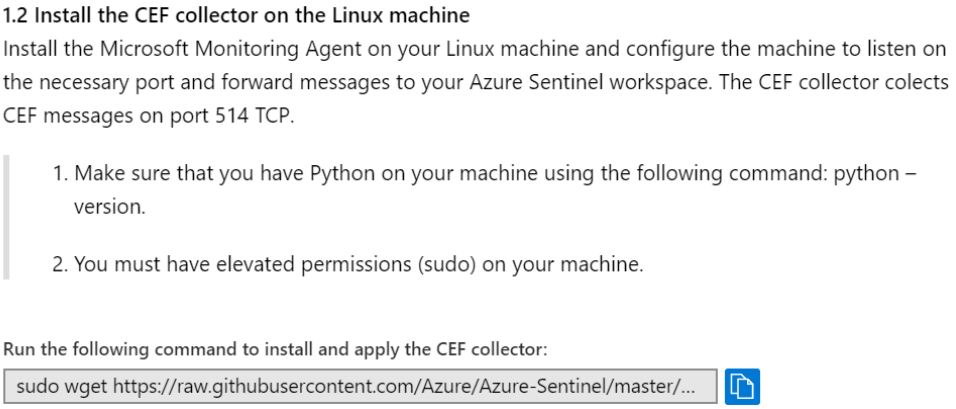
6. Select your new workspace and click **Add Azure Sentinel**, your new workspace will then be created and be added to **Sentinel**
7. In the side menu select **Data Connectors**




8. Search for **Common Event Format**, select it and then click **Open connector page**




9. Copy the command listed at chapter 1.2 and keep it in a safe location: this will be required during the installation wizard of our integration package. The ‘python’ in this command needs to be replaced by ‘python3’




## Step 2 – Configure SMC to allow connections from API clients

We need to enable API access in order to export logs from the Security Management Center. The instructions to **Enable SMC API** can be found in the official documentation at this link:

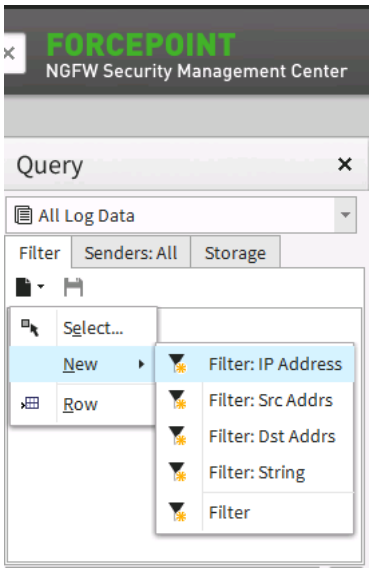
https://www.websense.com/content/support/library/ngfw/v66/rfrnce/ngfw_660_rg_smc-api_b_en-us.pdf

## Step 3 – Creating custom log filters from SMC

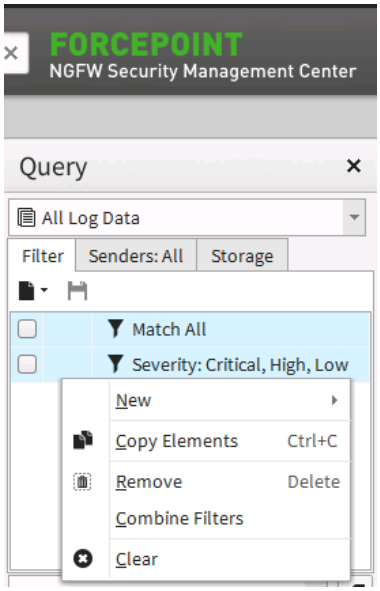
Since logs ingestion into Azure Sentinel is charged based on the number of **log events**, it is important to control what logs are forwarded from the NGFW into Azure.

This integration package enables the filtering of logs based on customizable queries. queries can be built using the SMC UI and then exported in a format that can be passed directly to the **SMC2CLOUD** service: by doing so, users will be able to find in Azure Sentinel the same logs they would see applying the filters in the SMC interface.

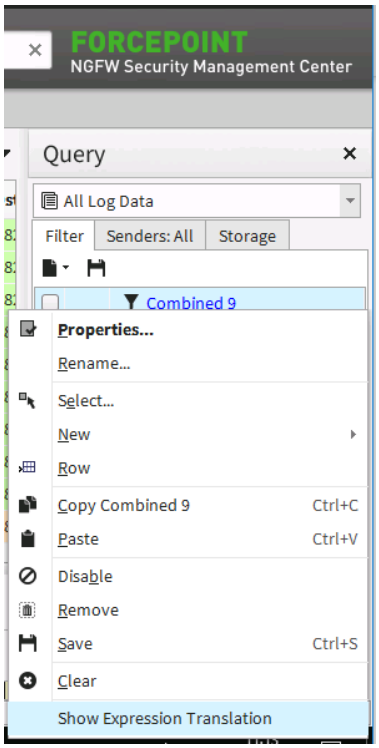
1. Open the SMC log view
2. Find the filter side bar and create a new filter, validate it returns what is required by clicking **Apply**



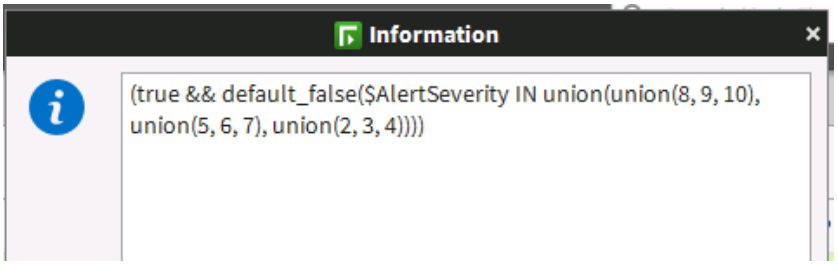

3. If you have two entries in the filter view after creating your filter, select both, right click them and select ‘Combine Filters’




4. After you combine the filters you will have one entry titled **Combined <x>** where x is some numerical value that increments after each combined filter
5. Now we need to export our filter in a format that can be used by our integration tool. Right click on the combined filter and select **Show Expression Translation**




6. A dialog will pop up with a textual representation of the filters we just created




7. Copy this line of text exactly as it is in the dialog box and store it in a safe location: this will be required during the installation steps of the **SMC2CLOUD** service.

## Example of common log queries

- All events matching a severity of Critical, High or Low
  *(true && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7), union(2, 3, 4))))*

- All events matching a specific **rule tag** (the number in red being the rule tag)
  *(true && default_false((($RuleId & 0x1fffff) | (($RuleId & 0x7fffffe00000000) >> 12)) == 2097162))*

- All events with an action matching “Terminate” or “Block”
  *(true && default_false($Action IN union(9, 13)))*

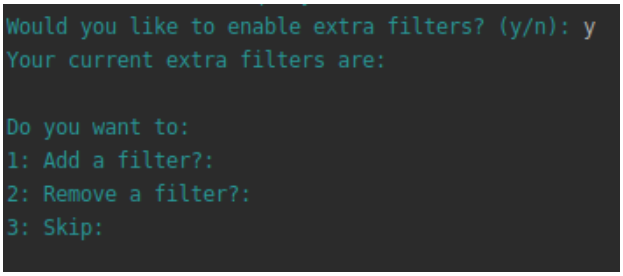
- Any System Alert events
  *(true && defined($Alert))*

- All Anomalies with severity Critical or High
  *((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))*

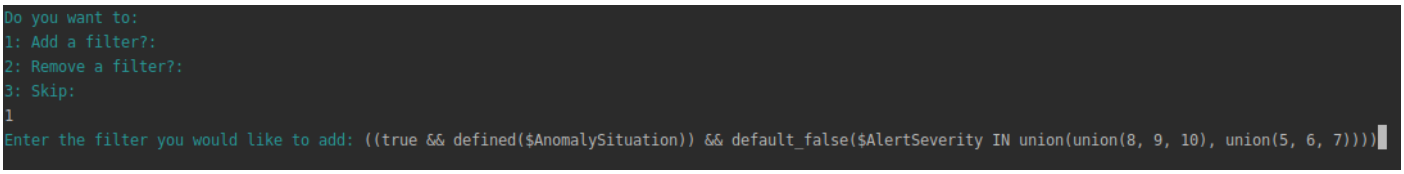
## Adding extra filters

During the configuration step the install wizard will ask for a **default filter**, since at least one filter is needed in order to match NGFW logs that will be forwarded to Azure. User can also add **extra filters** so that the filtering process can be performed in a modular way, and filters can be selectively removed at a later stage without editing the syntax of the **default filter**.

1. Choose ‘y’ and you will be presented with this screen




2. When you choose to add a filter, paste the filter syntax in the terminal




3. Once you have hit enter the configuration process will continue

## Removing extra filters

During the configuration step you will be asked if you want to add extra filters. In this case, choose **y** also if you want to remove existing filters

```
Would you like to enable extra filters? (y/n): y
Your current extra filters are:
1 : ((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))
2 : ((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))
3 : ((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))

Do you want to:
1: Add a filter?:
2: Remove a filter?:
3: Skip:
```

1. Select option 2 to remove a filter
2. Enter the number of the filter you want to remove

```
Do you want to:
1: Add a filter?:
2: Remove a filter?:
3: Skip:
2
Enter the index of the filter you would like to remove: 2
Your updated extra filters are:
1 : ((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))
2 : ((true && defined($AnomalySituation)) && default_false($AlertSeverity IN union(union(8, 9, 10), union(5, 6, 7))))
```

3. The filter at the selected will be deleted and the configuration process will continue

## Step 4 – Configuration and installation of the SMC2CLOUD service

1. cd to the directory containing **fp-ngfw-exporter-cloud-v1.tar.gz**
2. Decompress the above file with the command **tar -xvzf fp-ngfw-exporter-cloud-v1.tar.gz**
3. There will be a new folder created with the name **fp-ngfw .** cd to **fp-ngfw**
4. Make **install.sh** executable with the command **chmod u+x install.sh**
5. Run **sudo ./install.sh**
6. Fill in the requested details during the configuration step
7. Wait for the installation to complete
8. Run **sudo systemctl status SMC2CLOUD.service** to verify the service has been created and is running properly

```
● SMC2CLOUD.service - Service to query log events from the NGFW and upload to AWS Security Hub and Azure Sentinel
   Loaded: loaded (/lib/systemd/system/SMC2CLOUD.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2019-12-18 13:59:31 GMT; 5min ago
 Main PID: 13537 (python3.6)
    Tasks: 1 (limit: 4915)
   CGroup: /system.slice/SMC2CLOUD.service
           └─13537 /opt/ngfw_2_cloud/venv/bin/python3.6 /opt/ngfw_2_cloud/ServiceRunner.py
```

## Appendix A – Configuration parameters

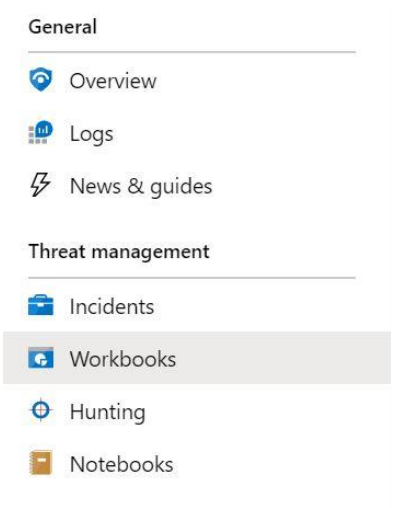
The following table provides a description of the parameters in the **cfg.json** file that are populated by the installer script upon it first execution:

| Parameter | Description | Required |
|---|---|---|
| **host-ip** | IP address of the machine hosting SMC | YES |
| **host-port** | Port opened on the SMC for the API client | YES |
| **client-api-key** | API key from obtained from NGFW for API client connection | YES |
| **fetch-size** | Number of records to retrieve from the SMC logs | YES |
| **run-interval** | How often the systemd service will run, fallback is every 900 seconds (15 mins) | NO/ FALLBACK |
| **default-filter** | Default log filter exported from the SMC | YES |
| **extra-filters-enabled** | True/False, dependent on customer config | YES |
| **extra-filters** | Array of additional filters added to the default filter | NO |
| **azure-integration** | True/False depending on customer integrations | YES |
| **azure-agent-script** | Command provided by Azure to install log agent | YES |

## Appendix B – Create a Workbook into Azure Sentinel

Workbooks combine text, Analytics queries, Azure Metrics and parameters into rich interactive reports.

1. Login to Azure Sentinel portal
2. Select **Workbooks** from the left-hand menu, under **Threat management** section. This launches a workbook gallery




3. Click on **Add workbook**, this will open a new workbook
4. Click on **Edit**, this will make workbook sections editable


5. Click **Add query**, this will launch Log Analytics workspace Logs Query
6. Insert the following query

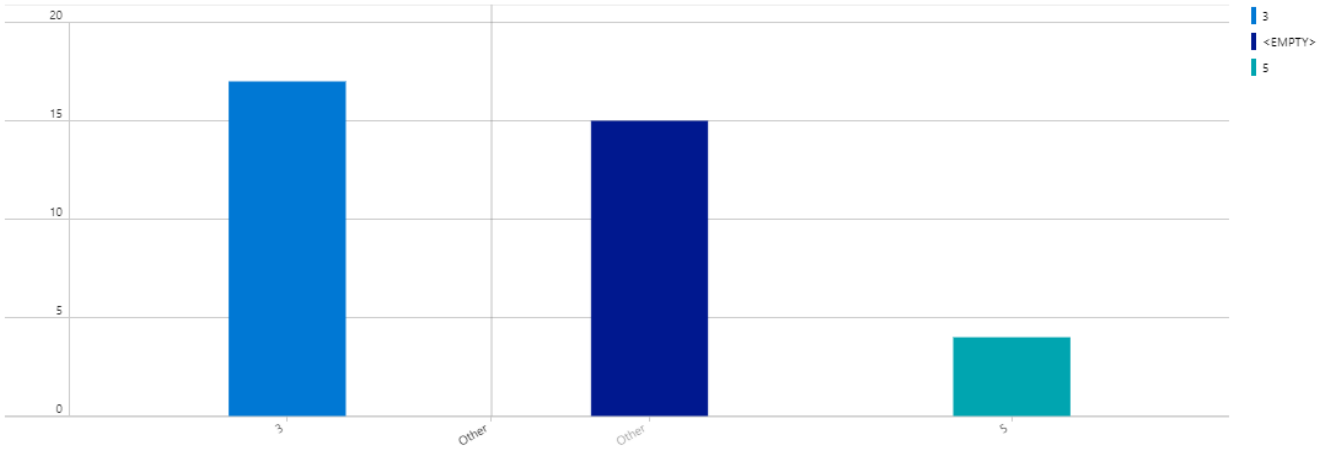
*CommonSecurityLog |summarize Count= count() by Activity | render barchart*

The above groups NGFW logs by **Activity type** and provides an output similar to this


7. Click **Done Editing**
8. Move to the next section of the workbook and click **Edit**
9. Add the following query to display a Bar Chart which provide a visual overview of the number of logs sent from NGFW into Azure Sentinel grouped by **Severity**

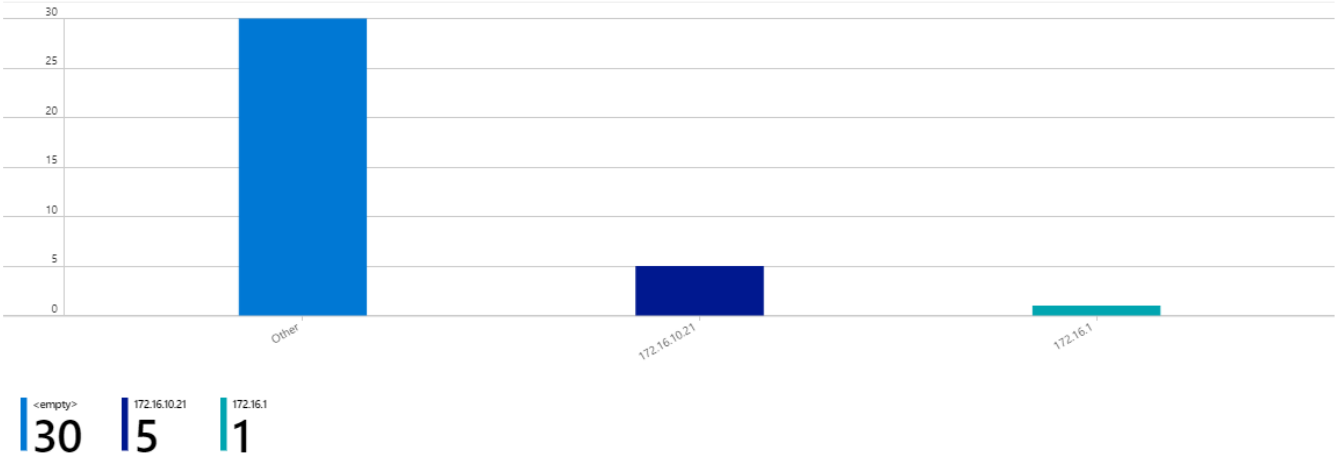
*CommonSecurityLog | summarize Count= count() by LogSeverity | render barchart*

10. Click **Done Editing**. The result displayed will be similar to this



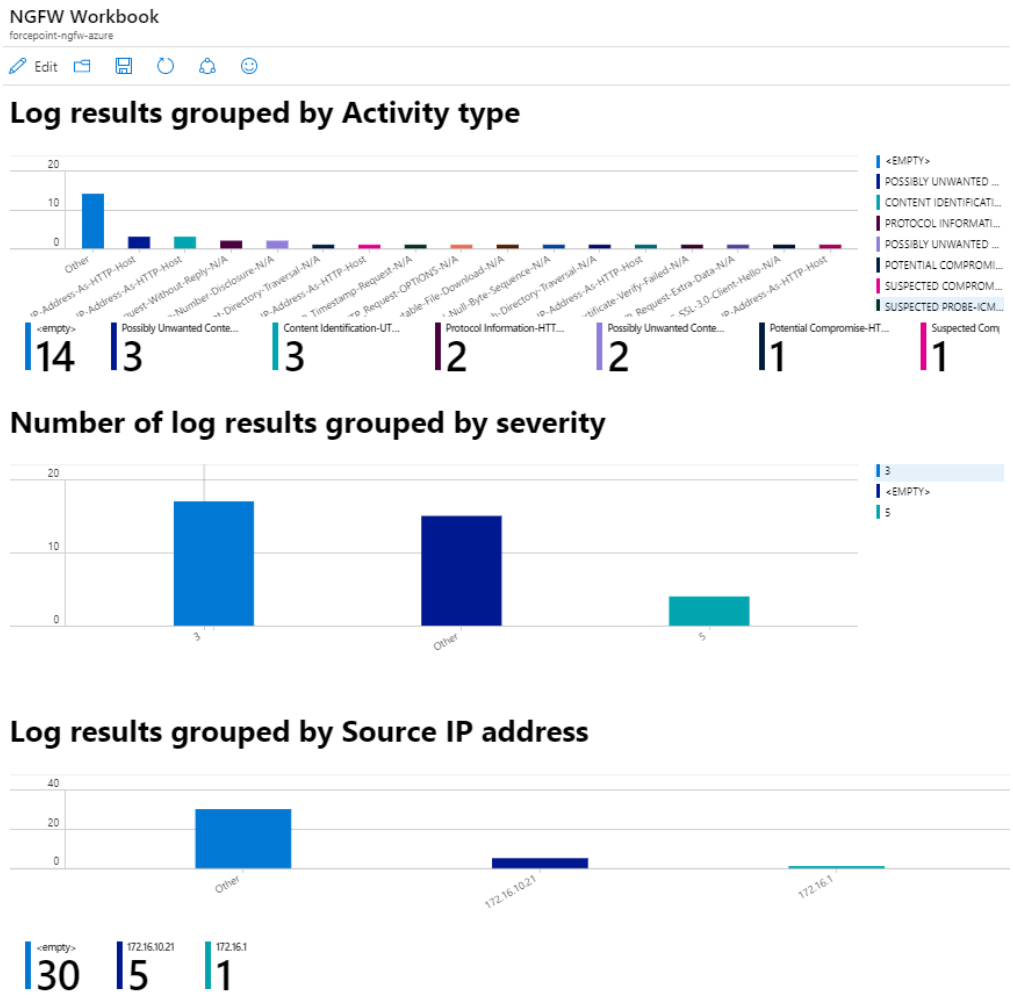

Another query to display NGFW logs grouping by **Source IP address** is

*CommonSecurityLog*
*| summarize Count= count() by SourceIP*
*| render barchart*




11. Once finished editing queries click **Done Editing** on the top left corner and on the save icon to save the workbook

Multiple queries can be used to populate a workbook with tables and chart, enabling powerful visualization of events and security related activities obtained from Forcepoint NGFW.



# Troubleshooting

Follow these steps to identify issues impacting the normal operation of the integration described in this document.

**Validate the prerequisites**

Make sure the prerequisites described in the Summary chapter are all satisfied:

- Check the versions of Forcepoint NGFW and SMC in use are listed as compatible:

  *Forcepoint NGFW 6.5.2*
  *Forcepoint NGFW Security Management Center (SMC) 6.6.0*

- Verify the integration component correctly operates on a clean Ubuntu 18.04 machine.
- Step 1 – Set up Azure Sentinel integration

  The Data Connector command mentioned in step 9 of the first step needs to be modified by the user: **python** needs to be replaced by **python3** as in th example below

  *sudo wget https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/CEF/cef_installer.py&&sudo python3 cef_installer.py1ooooo11122222nbbbbbb203jddded99993ccccccccccc22222222222221111111111144444445555388==*
- User must be root to run the installer.sh
- Check the user can download the files necessary to install **SMC2clou**d service: execute the following command

  *wget --content-disposition https://frcpnt.com/ngfw-sentinel-latest*

**Check network connectivity**

Make sure firewalls or other security appliances are not impacting the network connectivity necessary for the operation of all components involved into this integration:

- Check the host machine has network connectivity to NGFW-SMC: execute the following command

  *ping -c 5 <smc-ip-here>*

  and check that the ping is successful

- Check the host machine also has network connectivity to azure: execute the following command

  *ping -c 5 <azure-ip-here>*

  and check the ping result is successful

**Check dependencies are installed**

Make sure the software dependencies needed by the components involved into this integration are installed:

- Check the python versions installed on the host Ubuntu machine with the following commands:

  *python --version*
  *python3 --version*

  and check the result has both python 2.x and python 3 versions on the host Ubuntu machine

- Check **SMC2CLOUD.service** is installed: execute the following command on the Ubuntu machine

  *systemctl status SMC2CLOUD.service*

  and check the result is similar to below:

```
neelima@ubuntu:~/Downloads/fp-ngfw$ systemctl status SMC2CLOUD.service
● SMC2CLOUD.service - Service to query log events from the NGFW and upload to AWS Security Hub and
   Loaded: loaded (/lib/systemd/system/SMC2CLOUD.service; enabled; vendor preset: enabled)
   Active: active (running) since Mon 2020-02-24 08:58:45 PST; 16s ago
 Main PID: 6299 (python3)
    Tasks: 1 (limit: 2293)
   CGroup: /system.slice/SMC2CLOUD.service
           └─6299 /opt/ngfw_2_cloud/venv/bin/python3 /opt/ngfw_2_cloud/ServiceRunner.py

Feb 24 08:58:45 ubuntu systemd[1]: Started Service to query log events from the NGFW and upload to
```

- Check **omsagent** service is installed: execute the following command on the Ubuntu machine (user can use tab to autofill the full name for omsgent servcie)

  *systemctl status <omsagent-here>*

  and check the result is similar to below:

```
neelima@ubuntu:~/Downloads/fp-ngfw$ systemctl status omsagent-
● omsagent-f1c22682-256c-4c4a-a0c6-8d437349ec93.service - Operations Management Suite agent
   Loaded: loaded (/lib/systemd/system/omsagent-                                    ; enabl
   Active: active (running) since Mon 2020-02-24 08:57:36 PST; 1min 59s ago
 Main PID: 6105 (omsagent)
    Tasks: 32 (limit: 2293)
   CGroup: /system.slice/omsagent-
           └─6105 /opt/microsoft/omsagent/ruby/bin/ruby /opt/microsoft/omsagent/bin/omsagent -d /va
```

- If **omsagent** service is not running (or has loaded status): execute the following command:

  *journalctl -r*

  to see where the install is failing. If you see any error messages with “Connection Refused” or “dpkg”, that could have something to do with the software updates on the Ubuntu machine. Restart the host machine and see if you see any software upgrade request. If you do, please install all the updates.

Once the problem is identified and rectified, it will be necessary to stop and delete the **SMC2CLOUD.service** with the following commands:

*systemctl stop SMC2CLOUD.service*
*systemctl disable SMC2CLOUD.service*
*sudo rm /etc/systemd/system/ SMC2CLOUD.service*
*sudo rm /lib/systemd/system/omsagent*

and check that the service is stopped with the command:

*systemctl status SMC2CLOUD.service*

- The user can also make changes to the **cfg.json** file in the **fp-ngfw** folder if the input for any of the configuration parameters is wrong and then restart the service with the below command:

  *systemctl restart SMC2CLOUD.service*

  Check status of the SMC2CLOUD service with the below command:

  *systemctl status SMC2CLOUD.service*

  If the above service is running, you should start seeing logs in Azure Sentinel in 10-20 minutes.