FORCEPOINT

Forcepoint CASB and Azure Sentinel

Integration Guide

Dlo Bagari
Mattia Maggioli
14 September 2020
Public

Version	Date	Author	Notes
0.1	11 December 2019	Dlo Bagari	First draft
0.2	12 December 2019	Mattia Maggioli	Review
0.3	30 December 2019	Jonathan Knepher	Review
0.4	03 January 2020	Dlo Bagari	Update
0.5	20 January 2020	Dlo Bagari	Updated package names
0.6	12 February 2020	Dlo Bagari	Update Screenshot and fields table
0.7	23 March 2020	Neelima Rai	Added troubleshooting chapter
0.8	14 September 2020	Mattia Maggioli	Minor updates

Summary

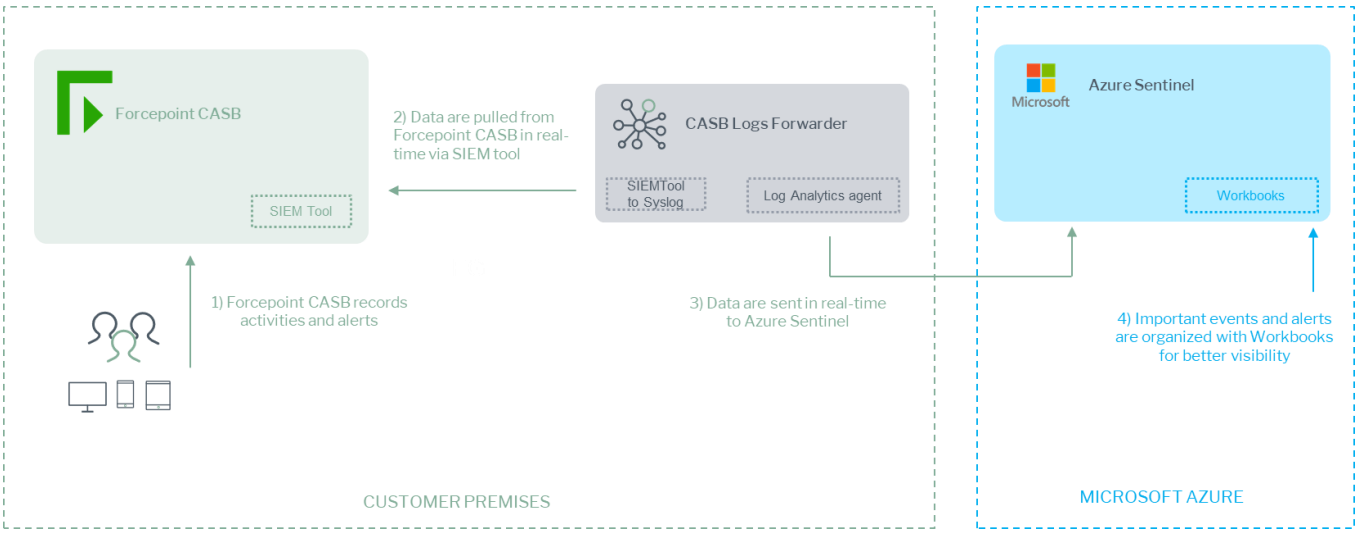
This guide provides step by step instructions to configure an event driven pipeline to pass Forcepoint CASB logs to Azure Sentinel and using the Azure Monitor Workbooks to create custom dashboards from received data.

The code and instructions provided enable system administrators to automatically:

- Export logs from Forcepoint CASB using SIEM Tool into an intermediate Syslog service
- Configure Syslog to filter logs in CEF format and forward to Azure Log Analytics Agent only the ones that match the filtering conditions set by the user
- Configure Azure Log Analytics Agent to receive data from Syslog and forward data to an Azure Workspace

This interoperability enriches visibility into user activities recorded by CASB, enables further correlation with data from Azure workloads and other feeds, and improves monitoring capability with Analytics queries 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:

- Azure Sentinel
- Forcepoint CASB SIEM Tool - version 2019-04-15

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/casb-sentinel-latest

- fp-casb-exporter-azure-v1.tar.gz

The **fp-casb-exporter-azure-v1.tar.gz** contains all files necessary to setup and run all the services used by **CASB Log Forwarder** to accomplish the interoperability between Azure Sentinel and Forcepoint CASB

- **Azure CASB Service:** runs SIEM Tool to download logs/events from Forcepoint CASB and exports them to Syslog service
- **Azure CASB Timer Service**: monitors and restarts Azure CASB Service.

We suggest deploying the **CASB Log Forwarder** on an Ubuntu 18.0.x machine with at least 2 GB RAM and 20 GB of storage, the instructions provided in this document are based on this operating system and the following packages

- Java 8
- Python 3.7
- Syslog-ng Daemon
- Firewalld
- net-tools
- unzip

The software packages and related dependencies are automatically installed by the **azure_casb_installer.sh** script provided inside the **fp-casb-exporter-azure-v1.tar.gz** file, which will execute the following commands as part of the deployment script of the **CASB Log Forwarder**:

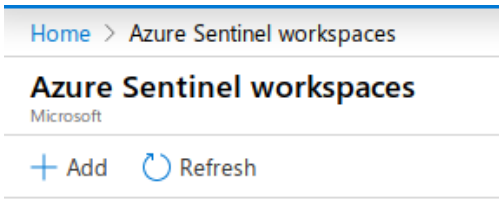
sudo apt install python3.7 -y
sudo apt install unzip -y
sudo apt install openjdk-8-jdk -y
sudo apt install syslog-ng syslog-ng-core -y
sudo apt install firewalld -y
sudo apt install net-tools -y

The machine hosting the **CASB Log Forwarder** will be referenced in the rest of this document with the name "**Syslog Proxy**".

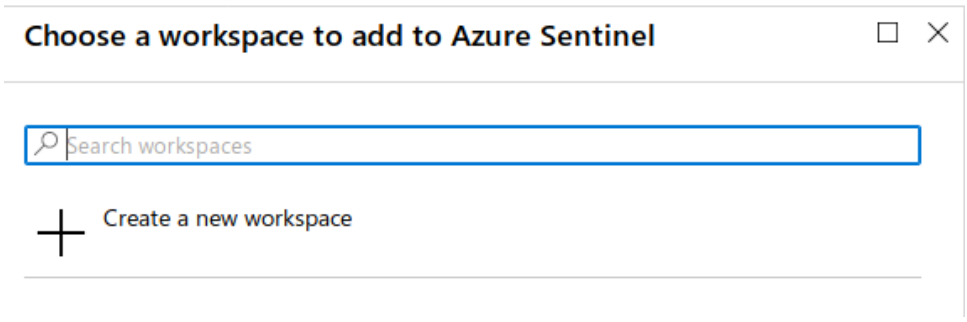
Step 1 – Create Log Analytics Workspace

In order to send logs/events to Azure Sentinel we need to create an Azure workspace where all logs/events will be stored.

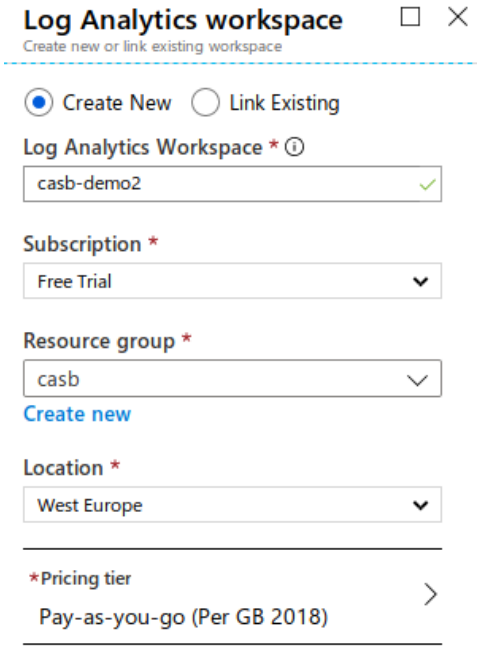
1. Sign into Azure portal
2. Click on **All services**, select **Azure Sentinel** click on it.
3. Click on **Add**



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.



6. Click **Ok** to create the workspace (this might take few minutes)
7. Click on Add Azure Sentinel

Step 2 – Configuration for CASB Log Forwarder

The parameters required by CASB Log Forwarder are stored in a single file called **settings.yml**:

```
log_azure_agent: sudo wget https://raw.githubusercontent.com/Azure/Azure-Sentinel/master/DataConnectors/CE
casb_host: my.skyfence.com
application_directory: /var/azure_casb
logs_directory: /var/azure_casb/logs
# the format for start date_time must be YYYY-MM-DD HH:MM:SS
logs_starting_date: 2019-11-01 00:00:00
logs_filter_parameters:
  include:
    device_product: Cloud Service Monitoring,SaaS Security Gateway,CASB Admin audit log,CASB Incidents
    action: act=Block,act="Block",act=Monitor,act="Monitor",act=MONITOR
  exclude:
    reason: reason=download
include_admin_audit_logs: true
SIEM_tool_outputs_location: /var/azure_casb/SIEM_TOOL_OUTPUTS
```

The following table provides a description of every parameter in the **settings.yml** file:

Parameter	Description	Requires to be changed
log_azure_agent	The command to download log analytics agents provided in the documentation of Azure Sentinel	YES
casb_host	The CASB host name, to be changed based on the instance in use (EU, US)	YES
application_directory	Directory where the CASB Log Forwarder is stored, by default /var/azure_casb	NO
logs_directory	The logs directory for application, by default /var/azure_casb/logs	NO
logs_starting_date	Allows to set a start date for the log export. Only CASB logs/events after this date will be downloaded. Must be in the format: 'YYYY-MM-DD HH:MM:SS'	YES
logs_filter_parameters	Allows to set custom filters to selectively export logs. More details on how to use this are explained in Step 2.1	YES
include_admin_audit_logs	Allows to include Admin audit logs to be sent to Log Analytics. Possible values are true or false	NO
SIEM_tool_outputs_location	The location to save SIEM tool outputs	NO

Step 2.1 – Selective logs export using filtering options

The parameter **logs_filter_parameters** in **settings.yml** is used to define the filters used to select which log will be forwarded into Azure Sentinel.

There are two groups of filters:

- **include**: this filter allows users to select CASB logs which contain at least one of the values provided as a comma-separated list. The format of each parameter is

  <parameter_name>: <possible_value1, possible value2,....., possible_valueN>

  The parameter_name is set by the user, so multiple parameters can be added in the **include** section, provided every parameter_name is unique.

  Example:
  ***device_product**: Cloud Service Monitoring,SaaS Security Gateway,CASB Admin audit log*

  This filter will match and forward to Azure Sentinel any log that contains either “Cloud Service Monitoring”, “SaaS Security Gateway” or “CASB Admin audit log”.

- **exclude**: this filter allows to exclude CASB logs which contain any of the values provided as a comma-separated list. The format of each parameter is

  <parameter_name>: <possible_value1, possible value2,....., possible_valueN>.

Example:
***reason**: reason=download*

This filter will match and exclude any log that contains the word “download”.

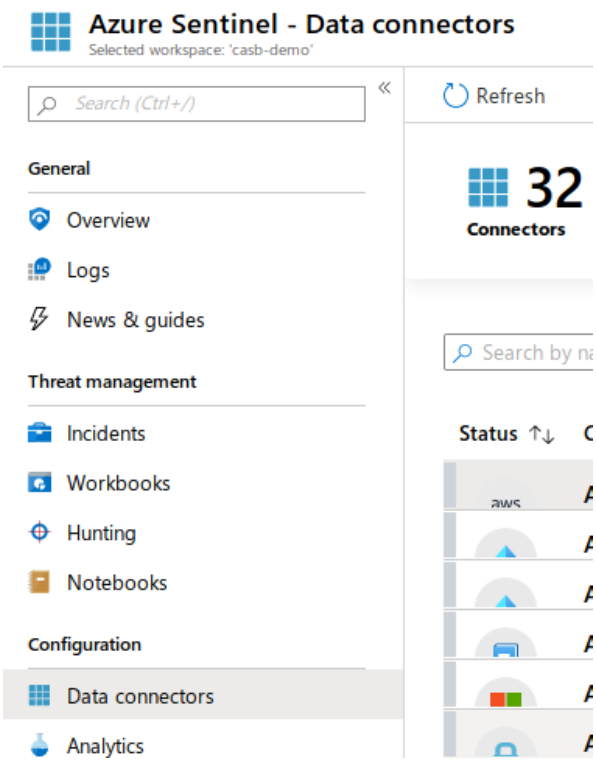
In order to fine tune the filtering logic and define appropriate filters, we advise to review a few logs offline, to identify the best parameters and keywords that will match the events that are relevant for the user.

Step 3 – Obtain Log Analytics Agent installation command

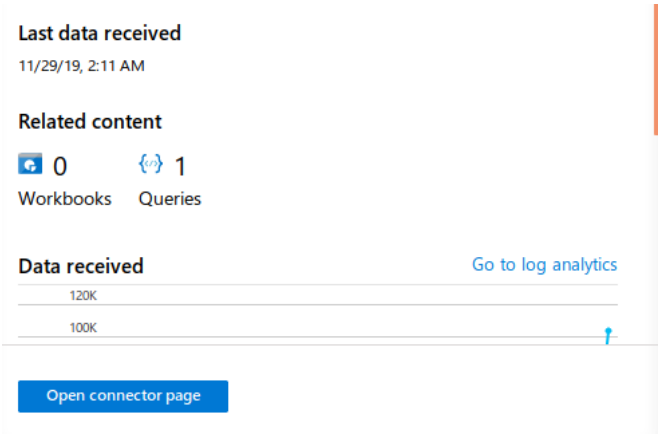
Azure Log Analytics Agent is used on the dedicated **Syslog Proxy** machine to forward filtered logs to Azure Sentinel and to handle eventual sending logic in case of network disruption that might temporarily make Azure Sentinel unreachable, so that logs are delivered and stored in an Azure Workspace.

In order to deploy the Azure Log Analytics Agent on **Syslog Proxy** machine, the Log Analytics Agent installation command must be added to the CASB Log Forwarder configuration file **settings.yml**.

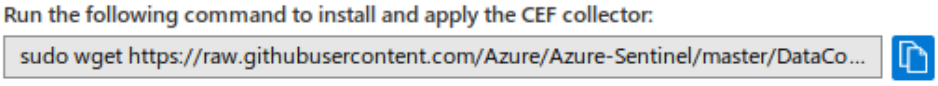
- Go to the Azure Sentinel portal click **Data connectors**



- Select Common Event Format (CEF) and then Open connector page



- Copy the command for CEF connector (Log analytics Agent)



- Insert the command as a value for **log_azure_agent** parameter in the configuration file **settings.yml**

Step 3 – Download CASB SEIM Tool and Trust Store Files

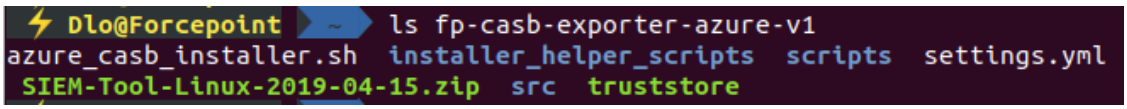
1. Login into Forcepoint CASB
2. On the top right corner, click on **Settings** to open the settings portal
3. Click on Tools and Agents, go to SIEM Tool

4. Click **Download Trust Store** and download the SIEM tool for Linux

Step 4 – Installing CASB Log Forwarder

Follow these steps to set up **CASB Log Forwarder** on the target machine

1. Login via SSH to the **Syslog Proxy** machine and copy the **fp-casb-exporter-azure-v1.tar.gz** file into **root** folder
2. Change user environment to sudo using the command **sudo su** and decompress the file using the command **tar -zxvf fp-casb-exporter-azure-v1.tar.gz**
3. Go into the **/root/ fp-casb-exporter-azure-v1** folder and edit the **settings.yml** file to update the value of the required parameters, change only the entries that require to be changed according to the table in step 2
4. Copy **truststore** and **SIEM tool** for Linux into the **fp-casb-exporter-azure-v1** directory. The context of fp-casb-exporter-azure-v1 must look as follows



5. Make sure the azure_casb_installer.sh file is executable using the command **sudo chmod a+x azure_casb_installer.sh**
6. To run **azure_casb_installer.sh** you need to have Forcepoint CASB instance username and password. Install CASB Log Forwarder using command:

sudo ./azure_casb_installer.sh --username <username> --password <password>

The installer script will install the dependencies, read the **settings.yml** file, move the code to the relevant directories, create and enable at boot all services.

7. Once the installation is finished, reboot the **Syslog Proxy** machine then log into the machine
8. Verify syslog-ng is listening to TCP port “TCP *:shell” and omsagent is listening to “TCP localhost:25226” using command **lsof -i | grep -e omsagent -e syslog-ng**

```
root@ubuntu:/home/dlo# lsof -i | grep -e omsagent -e syslog-ng
syslog-ng  645            root   14u  IPv4  28407      0t0  UDP *:syslog
syslog-ng  645            root   15u  IPv4  28408      0t0  TCP *:shell (LISTEN)
syslog-ng  645            root   16u  IPv4  39809      0t0  TCP localhost:41217->localhost:25226 (ESTABLISHED)
omsagent  1215        omsagent   11u  IPv4  35858      0t0  TCP *:25324 (LISTEN)
omsagent  1215        omsagent   21u  IPv4  35876      0t0  TCP localhost:25226 (LISTEN)
omsagent  1215        omsagent   23u  IPv4  35877      0t0  UDP localhost:25224
omsagent  1215        omsagent   25u  IPv4  39810      0t0  TCP localhost:25226->localhost:41217 (ESTABLISHED)
root@ubuntu:/home/dlo#
```

9. Verify the required systemd services are running using command

systemctl list-units | grep azure_casb

```
root@dlo:/home/dlo# systemctl list-units | grep azure_casb
azure_casb.service                                                                loaded active running
azure_casb_timer.service                                                          loaded active running
```

The logs/events will be pulled from Forcepoint CASB and forwarded to Azure Sentinel automatically. It might take few minutes for logs to appear in Azure Sentinel.

Appendix A – Updating filters configuration

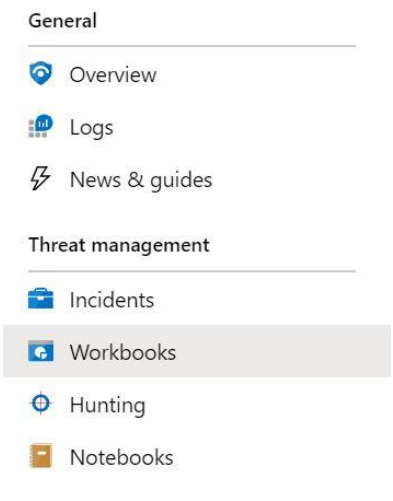
Once **CASB Log Forwarder** is installed, the only parameters of the **settings.yml** file that can be updated are **logs_filter_parameters** and **logs_starting_date**.

Any change to the parameters will be applied automatically to **CASB Log Forwarder** after a few seconds with no reboot needed.

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

```
CommonSecurityLog
| extend outcome = split(split(AdditionalExtensions, ";", 2)[0], "=", 1)[0]
| extend reason = split(split(AdditionalExtensions, ";", 3)[0], "=", 1)[0]
| where outcome =="Failure"
| summarize countFailure = count() by DestinationUserName, DestinationIP, SourceIP
| join kind= leftouter (CommonSecurityLog
    | extend outcome = split(split(AdditionalExtensions, ";", 2)[0], "=", 1)[0]
    | extend reason = split(split(AdditionalExtensions, ";", 3)[0], "=", 1)[0]
    | where outcome =="Success"
    | summarize countSuccess = count() by DestinationUserName, DestinationIP, SourceIP) on
DestinationUserName and SourceIP
| extend flagSummary = iff(isnull(countSuccess), strcat(countFailure, ":true:", iif(isnull(countSuccess),0,
countSuccess)), strcat(countFailure, ":false:", iif(isnull(countSuccess),0, countSuccess)))
|where flagSummary endswith("true:0") and isnotempty(SourceIP)
|project DestinationUserName,SourceIP,countFailure,countSuccess,flagSummary
|top 5 by countFailure
```

The above query searches for users who have logged in multiple times from multiple IP Addresses and failed. The query provides an output similar to this

DestinationUserName	SourceIP	countFailure	countSuccess	flagSummary
alan@skyromi.onmicrosoft.com	195.138.83.92	6		6:true:0
alan@skyromi.onmicrosoft.com	157.167.3.2	8		8:true:0
user1@skyromi.onmicrosoft.com	52.9.173.175	20		20:true:0
dana@redkites.onmicrosoft.com	108.47.295.23	31		31:true:0

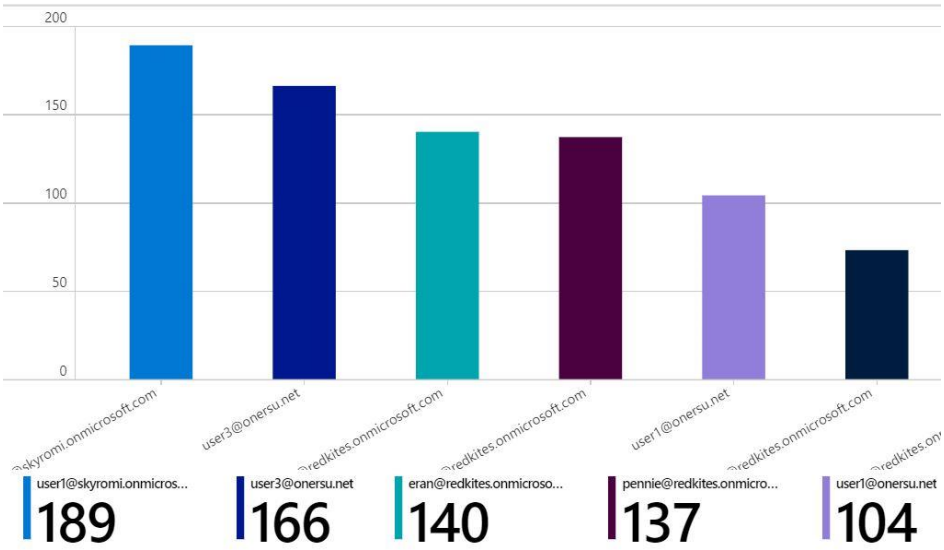
where the column **flagSummary** shows data in the following format

failed logins : flag status : successful logins

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 users with failed login attempts

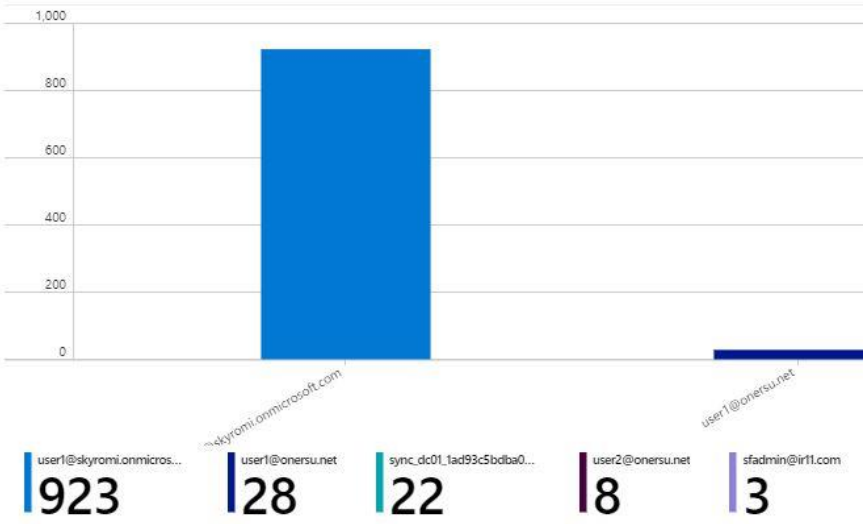
```
CommonSecurityLog
| extend outcome = split(split(AdditionalExtensions, ";", 2)[0], "=", 1)[0]
| extend reason = split(split(AdditionalExtensions, ";", 3)[0], "=", 1)[0]
| where outcome =="Failure"
| summarize Count= count() by DestinationUserName
| render barchart
```

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



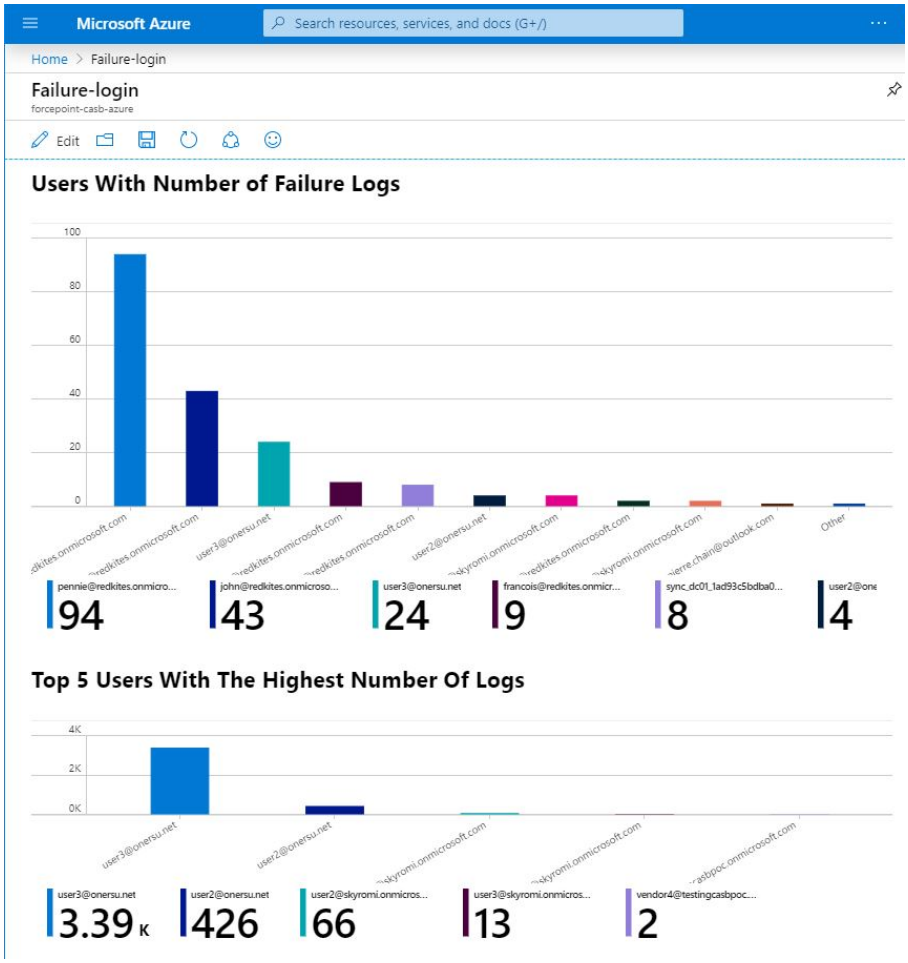
Another query to display the Top 5 Users by number of logs/events generated is

```
CommonSecurityLog
| summarize Count = count() by DestinationUserName
| top 5 by DestinationUserName
| 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 CASB.



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 CASB in use is listed as compatible

  Forcepoint CASB SIEM Tool - version 2019-04-15

- Verify the integration component correctly operates on a clean Ubuntu 18.04 with at least 2 GB RAM and 20 GB of storage
- If there are no logs in **/var/azure_casb/SCIM_TOOL_OUTPUT**, then the CASB SIEM tool truststore key is not valid. Obtain a truststore file which is current and verified working.
- User must be root to run the **azure_casb_installer.sh**
- Check the user can download the integration package with the below command:

  wget --content-disposition https://frcpnt.com/casb-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 **CASB Log Forwarder (Syslog Proxy)** has network connectivity to CASB: execute the following command on the Syslog Proxy host machine:

  ping -c 2 example-casb.url

  Replace the example URL/IP address with the one used. Once done check the result is similar to below:

  PING example-casb.url (10.10.120.12) 56(84) bytes of data.
  64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=179 ms
  64 bytes from 10.10.120.12 (10.10.120.12): icmp_seq=1 ttl=128 time=181 ms

Check dependencies are installed

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

- Check all dependencies are installed: execute the following command on the machine hosting the integration component to check for java:

java –version

and check the result is similar to below:

```
root@ubuntu:/home/neelima# java -version
openjdk version "1.8.0_242"
OpenJDK Runtime Environment (build 1.8.0_242-8u242-b08-0ubuntu3~18.04-b08)
OpenJDK 64-Bit Server VM (build 25.242-b08, mixed mode)
```

Note: The software versions may change depending on the latest upgrades.

- Check **python3** is installed, execute the following command:

  python3 --version

  Check the result is similar to below:

```
root@ubuntu:/home/neelima# python3 --version
Python 3.6.9
```

- Check **Firewalld** is operating normally, execute the following command:

  systemctl status firewalld.service

  and check the result is similar to below:

```
root@ubuntu:/home/neelima# systemctl status firewalld.service
● firewalld.service - firewalld - dynamic firewall daemon
   Loaded: loaded (/lib/systemd/system/firewalld.service; enabled; vendor preset: enabled)
   Active: active (running) since Wed 2020-03-11 10:31:42 PDT; 18h ago
     Docs: man:firewalld(1)
 Main PID: 706 (firewalld)
    Tasks: 2 (limit: 2293)
   CGroup: /system.slice/firewalld.service
           └─706 /usr/bin/python3 -Es /usr/sbin/firewalld --nofork --nopid

Mar 11 10:31:41 ubuntu systemd[1]: Starting firewalld - dynamic firewall daemon...
Mar 11 10:31:42 ubuntu systemd[1]: Started firewalld - dynamic firewall daemon.
```

- Check **unzip** is installed: execute the following command:

  unzip -h

  and check the first few lines of the result are similar to below:

```
root@ubuntu:/home/neelima# unzip -h
UnZip 6.00 of 20 April 2009, by Debian. Original by Info-ZIP.

Usage: unzip [-Z] [-opts[modifiers]] file[.zip] [list] [-x xlist] [-d exdir]
  Default action is to extract files in list, except those in xlist, to exdir;
  file[.zip] may be a wildcard.  -Z => ZipInfo mode ("unzip -Z" for usage).
```

- Verify the last few lines after installation completion are similar to below:

```
Creating systemd services
Enabling systemd services
Created symlink /etc/systemd/system/multi-user.target.wants/azure_casb.service → /etc/systemd/system/azure_casb.service.
Created symlink /etc/systemd/system/multi-user.target.wants/azure_casb_timer.service → /etc/systemd/system/azure_casb_timer.ser
vice.
```

Check all components are configured and running properly

Make sure the products and services involved into this integration are configured as expected and they are running:

- Check all components are configured and running as expected: verify **syslog-ng** daemon and **omsagent** are listening to the correct port with the following command:

  lsof -i | grep -e omsagent -e syslog-ng

  and check the result is similar to below:

```
root@ubuntu:/home/neelima# lsof -i | grep -e omsagent -e syslog-ng
syslog-ng  769            root   14u  IPv4  30779      0t0  UDP *:syslog
syslog-ng  769            root   15u  IPv4  30780      0t0  TCP *:shell (LISTEN)
syslog-ng  769            root   21u  IPv4  42101      0t0  TCP localhost:33533->localhost:25226 (ESTABLISHED)
omsagent  1208        omsagent    9u  IPv4  37815      0t0  TCP *:25324 (LISTEN)
omsagent  1208        omsagent   19u  IPv4  37819      0t0  TCP localhost:25226 (LISTEN)
omsagent  1208        omsagent   21u  IPv4  37820      0t0  UDP localhost:25224
omsagent  1208        omsagent   23u  IPv4  42102      0t0  TCP localhost:25226->localhost:33533 (ESTABLISHED)
dsc_host  2270        omsagent    6u  IPv4  50103      0t0  TCP ubuntu:48258->13.69.67.53:https (ESTABLISHED)
```

- Check the required services are running, execute the command below:

  systemctl list-units | grep azure_casb

  and check the result is similar to below:

```
root@dlo:/home/dlo# systemctl list-units | grep azure_casb
azure_casb.service                                                                    loaded active running
azure_casb_timer.service                                                              loaded active running
```