Forcepoint Behavioral Analytics and Azure Active Directory secure hybrid access

Integration Guide

Forcepoint

Integration Guide

Dlo Bagari
04 September 2020
Public

 Public

Table of Contents

Version	Date	Author	Notes
0.1	26 May 2020	Dlo Bagari	First draft
0.2	08 June 2020	Neelima Rai	Added Troubleshooting chapter
0.3	09 June 2020	Mattia Maggioli	Review
0.4	24 June 2020	Jonathan Knepher	Review
0.5	04 September 2020	Mattia Maggioli	Updated layout and styles

Summary

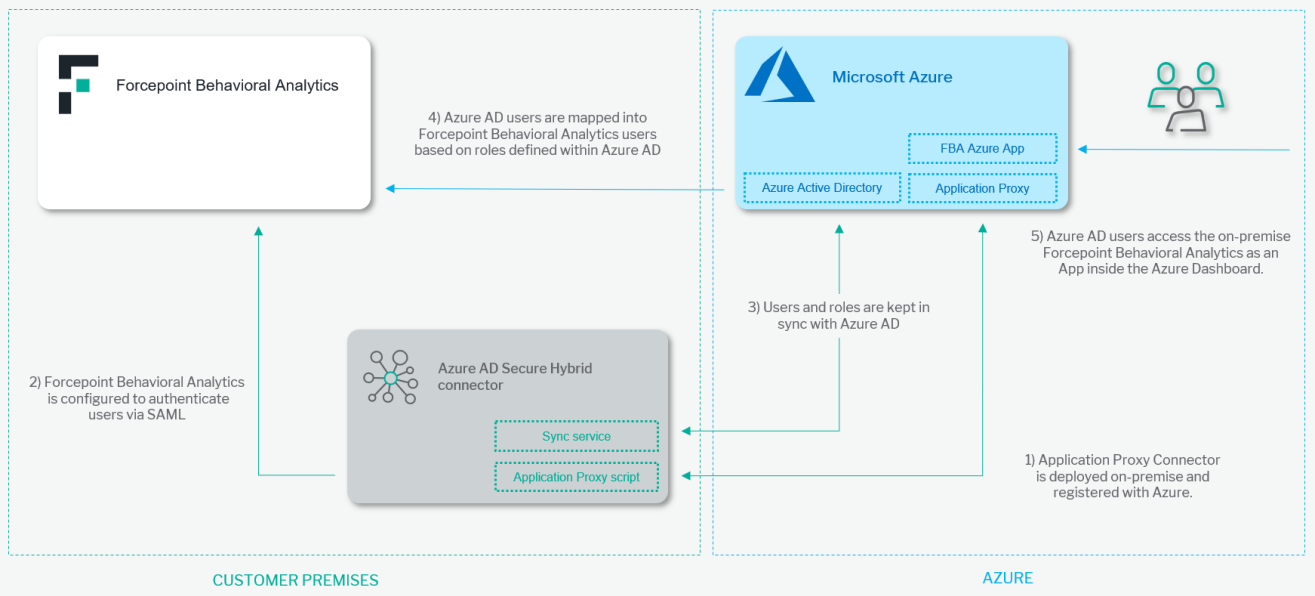
This guide provides step by step instructions to set up an integration between **Azure Active Directory (Azure AD) secure hybrid access** and **Forcepoint Behavioral Analytics.**

The automated integration enables Forcepoint Behavioral Analytics access and authentication through Azure AD users/policies and exposes Forcepoint Behavioral Analytics as an Azure app for remote management: selected Azure AD users can be assigned with different levels of access into Forcepoint Behavioral Analytics.

The code and instructions provided enable system administrators to **automatically:**

- → Silently download and install Azure Application Proxy Service Connector on a Windows machine.
- → Register Azure Application Proxy Service Connector with your Azure tenant.
- → Create an Azure Application and link it with Azure Application Proxy Connector.
- → Configure an Azure Application for Single Sign-On with SAML.
- → Create Azure groups for Forcepoint Behavioral Analytics Roles
- → Sync users assigned to the Azure application with Forcepoint Behavioral Analytics user accounts
- → Control and manage Forcepoint Behavioral Analytics user roles via Azure Portal
- → Configure Forcepoint Behavioral Analytics UI for Single Sign-On with SAML.

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



Caveats

The integration described in this document was developed and tested with the following product versions:

- → **Forcepoint Behavioral Analytics** version 3.1.0
- → **Windows Server 2016** as hosting machine of the **Application Proxy Connector** component

This interoperability uses:

- → **Application Proxy script:** a PowerShell script to download and install Azure **Application Proxy Connector**, and create an Azure

Application configured with Application Proxy and single sign-on using SAML.

→ **Sync Service**: a service that syncs users assigned with the Azure Application and Forcepoint Behavioral Analytics users. Also, this service applies roles to Forcepoint Behavioral Analytics users according to group membership assigned to selected Azure AD users.

Implementation options
Two implementation options are provided in this document

1. Docker – leverages docker images where the integration component is already installed with all necessary dependencies.
2. Traditional – requires the manual deployment of the integration component inside a clean Centos 7 host-machine.

The docker images for this integration have been tested working with:

→ Docker 19.03.6

while the traditional version of this integration has been tested working with the following requirements

→ Centos 7.x with at least 2 GB RAM and 20 GB disk

Azure Application Proxy Connector & Azure App

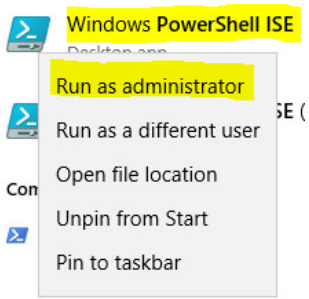
The solution described in this chapter requires a Windows machine (recommended Windows Server 2016) **within the same network of Forcepoint Behavioral Analytics UI machine**.

Application_proxy.ps1 is a PowerShell script implemented to automate the following task:

1. Install required packages and PowerShell modules on the Windows machine
2. Download Azure Application Proxy Service connector on the Windows machine
3. Install Azure Application Proxy Service connector on the Windows machine
4. Register your Azure AD tenant with the Application Proxy connector installed on the Windows machine
5. Create an Azure Application with the name '**Forcepoint Behavioral Analytics'**
6. Configure '**Forcepoint Behavioral Analytics**' application to use Application proxy and Single sign-on with SAML

To download and execute the script on your Windows machine do the following steps:

1. Run PowerShell ISE as administrator



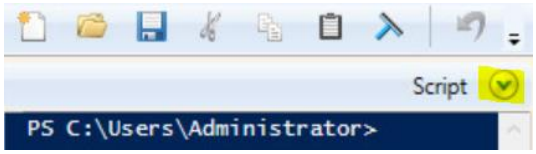
2. Execute the following command to allow your PowerShell to download files

```
[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
```

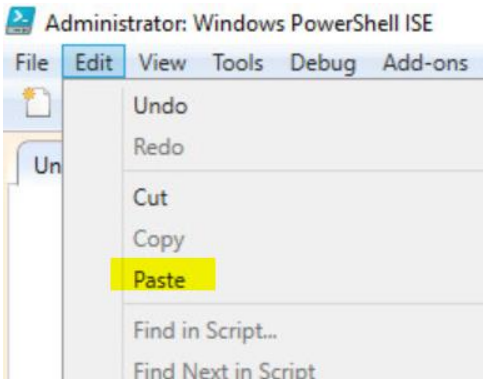
3. Execute the following command to download the application_proxy.zip file into your home directory

```
Invoke-WebRequest –Uri https://frcpnt.com/application_proxy-latest  -OutFile ~\application_proxy.zip
```

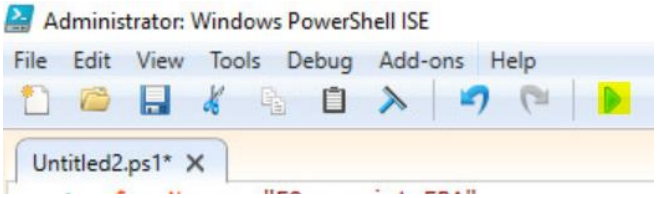
4. Unzip the **application_proxy.zip** file. This will generate a folder called **application_proxy** which contains **application_proxy.ps1** script.
5. Open **application_proxy.ps1** with Notepad and copy its contents.
6. On the **Powershell ISE** click on **Script** to open the script pane.



7. Paste the contents of **application_proxy.ps1** into script pane.

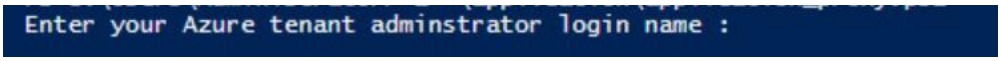


8. Run the script by clicking on  icon.

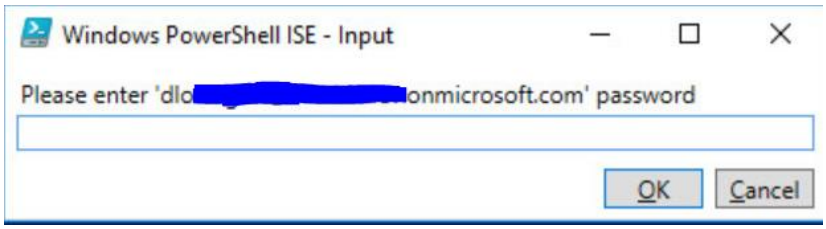


When the script runs for the first time it will check the PowerShell version and .NET framework version: if an update is required the script will download these updates and install them (this might take a few minutes). Your Windows machine might reboot after installing the required updates. If your machine has rebooted, repeat the steps above to copy the script into PowerShell script pane and click the  icon to execute it.

9. Enter your Azure administrator login-name:

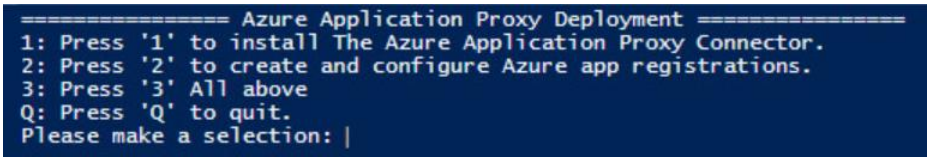


10. Next, you will be asked to enter the password for your Azure administrator account. Enter your password and click **OK**.

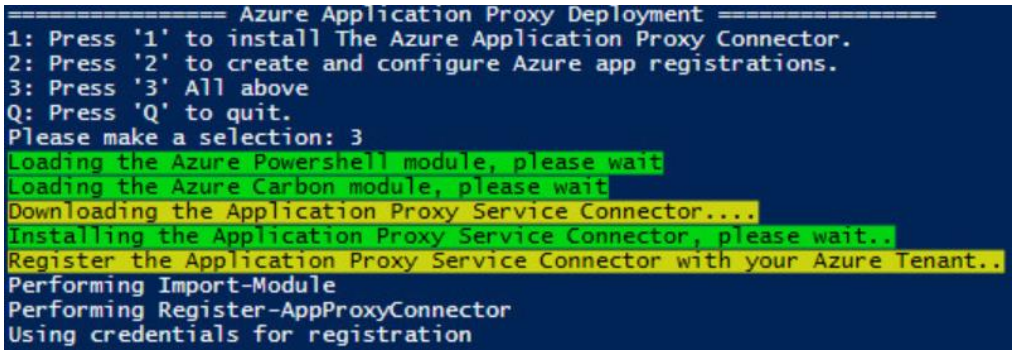


11. Next, you will be asked to enter the private IP address for the Forcepoint backend server: this is the private IP address of the machine hosting the Forcepoint Behavioral Analytics UI component (also called **ro-ui** service).

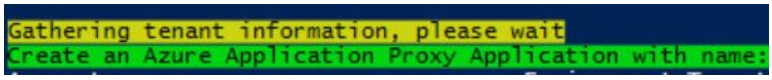
12. The following menu will be displayed, press **3** then **Enter**.



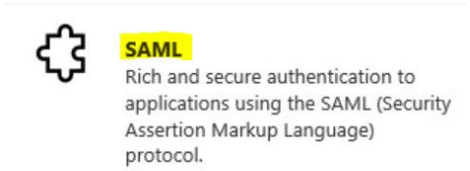
The first part of the script will download, install and register Azure **Application Proxy Service Connector**



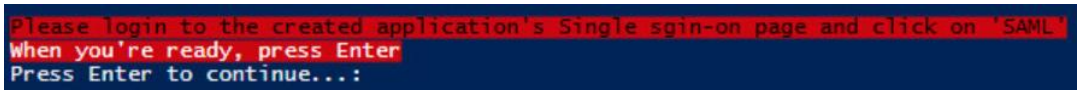
The second part of the script creates an Azure application with the name **Forcepoint Behavioral Analytics** and configure it.



13. Once the Azure application **Forcepoint Behavioral Analytics** has been created and configured, the script will open your web browser and ask you to login to the application's single sign-on page. Enter your Azure credentials to load the single sign-on page and click on **SAML**.



14. Go back to your PowerShell console and press **Enter**.



15. Press **q** then **Enter** to quit.

Implementation - Docker

The solution described in this chapter requires:

→ A Linux machine (Centos 7.3 recommended with at least 2 GB RAM and 20 GB disk) within the same network where Forcepoint Behavioral Analytics is installed. This machine will be referenced in the rest of this document as the **Docker-host Machine.**

The following component must be existing in the **Docker-host Machine**:

→ Docker Engine installed on the Docker-host: if Docker Engine is not installed visit docker-installation-docs to install Docker Engine on Docker-host

Step 1: Configure PostgresSQL SSL communication with Docker-host Machine
The **Sync Service** interacts with the Forcepoint Behavioral Analytics PostgresSQL database to add/remove users, user's roles, and sessions using SSL connections to encrypt the communications between the Sync Service component and the Forcepoint Behavioral Analytics PostgresSQL database.
The Forcepoint Behavioral Analytics Postgres machine needs to be configured to accept SSL connections from the Sync Service.

Login to **Forcepoint Behavioral Analytics postgres machine** and do the following:

1. Open **/data/ro-postgres/pg_hba.conf**

```
vi /data/ro-postgres/pg_hba.conf
```

2. Add the following line into **/data/ro-postgres/pg_hba.conf** under # **"local"**

```
hostssl    the_ui      postgres     <DOCKER_PRIVATE_HOST_IP_ADDRESS>   255.255.255.0    trust
```

If your PostgresSQL databse name is not **the_ui**, replace **the_ui** with your database name.

If your PostgresSQL username is not **postgres**, replace **postgres** with your username

3. Save **/data/ro-postgres/pg_hba.conf**
4. Reboot the Postgres service using the following command

```
systemctl restart postgresql-9.6.service
```

Next, Login to your **Docker-Host Machine** and do the following steps.

Step 2: Login to Docker Registry

```
root@linux:~# docker login docker.frcpnt.com
Username: fp-integrations
Password: t1knmAkn19s
Login Succeeded
```

Step 3: Create a configs directory

```
mkdir /root/configs
```

Step 4: Create config.yml file

```
vi /root/configs/config.yml
```

Step 5: Copy the following into /root/configs/config.yml file and update the required parameters

```
POSTGRES_HOST: INSERT_FBA_POSTGRES_SERVER_PRIVATE_IP_ADDRESS_HERE
POSTGRES_PORT: 5432
POSTGRES_USER_NAME: postgres
POSTGRES_DATABASE_PASSWORD: “”
POSTGRES_DATABASE_NAME: the_ui
AZURE_APPLICATION_NAME: Forcepoint Behavioral Analytics
USERS_SYNC_TIME_IN_MINUTES: 3
DEFAULT_FBA_PASSWORD: INSERT_A_DEFAULT_PASSWORD_FOR_NEW_FBA_USERS
SSO_CONFIG_SCRIPT_PATH: /app/configs/sso_config_script.sh
```

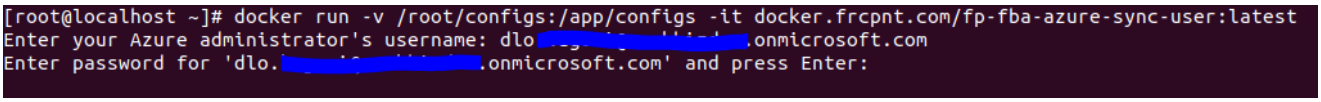
The parameters in **/root/configs/config.yml** are:

- → **POSTGRES_HOST**: is the Forcepoint Behavioral Analytics Postgres server’s private IP address. Change the value of this parameter to match your Forcepoint Behavioral Analytics Postgres server private IP address.
- → **POSTGRES_PORT**: is the Postgres port number, the default value is 5432. DO NOT CHANGE this value unless your Postgres service configured with a different port
- → **POSTGRES_USER_NAME**: the username for SQL Postgres, the default value is **postgres**. DO NOT CHANGE this value unless your SQL Postgres service configured with a different username
- → **POSTGRES_DATABASE_PASSWORD**: The Postgres user’s password. if you DO NOT have password remove this parameter from **config.yml**
- → **POSTGRES_DATABASE_NAME**: is the name of the database that contains Forcepoint Behavioral Analytics user’s tables, the default name is **the_ui**. DO NOT CHANGE this value unless your Postgres service has a different database name for user’s tables
- → **AZURE_APPLICATION_NAME**: is the Azure Application name, the default value for this parameter is **Forcepoint Behavioral Analytics**. DO NOT CHANGE the value of this parameter.
- → **USERS_SYNC_TIME_IN_MINUTES**: default value for this parameter is 3, the **Sync Service** will sync between Azure users and Forcepoint Behavioral Analytics users with the frequency defined with this value.
- → **DEFAULT_FBA_PASSWORD:** a default password for the newly created users in Forcepoint Behavioral Analytics. When a new user is assigned to your Azure application and that user does not exists in Forcepoint Behavioral Analytics users database, the **Sync Service** will create that user in the Forcepoint Behavioral Analytics database and use this value as its password. Once the user logs in to Forcepoint Behavioral Analytics they can change their password. Change the value of this parameter.
- → **SSO_CONFIG_SCRIPT_PATH**: is the path for the **sso_config_script.sh** script which will be generated by **Sync Service**. DO NOT CHANGE the value of this parameter. This script is necessary to switch Forcepoint Behavioral Analytics UI from local credentials into SAML authentication.

Step 6: Start Sync Service Docker Container
Execute the following command to start the **Sync Service** docker container:

```
docker run -v /root/configs:/app/configs -it docker.frcpnt.com/fp-fba-azure-sync-user:latest
```

Then enter your Azure credentials (username and password):

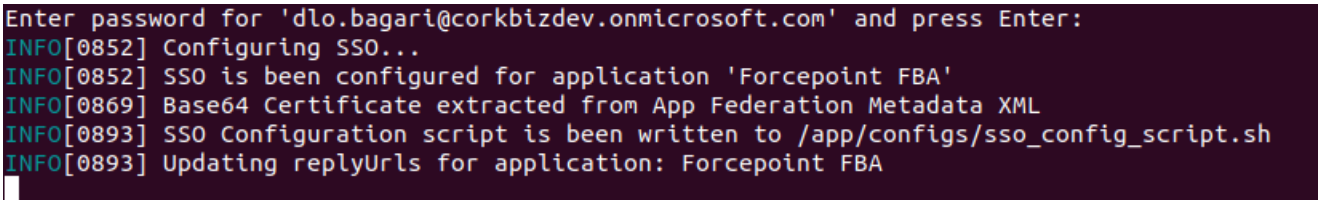


Note: you can add the following parameters to the **/root/configs/config.yml** file to avoid entering credentials manually.

```
AZURE_ADMIN_LOGIN_NAME: INSERT_YOUR_AZURE_USER_NAME_HERE

AZURE_ADMIN_LOGIN_PASSWORD: INSERT_YOUR_AZURE_PASSWORD_HERE
```

When **Sync Service** runs for the first time, it will configure your Azure application for single sign-on with SAML and generate a shell script for configuring Forcepoint Behavioral Analytics for Single sign-on with SAML. This script will have to run inside the machine hosting the Forcepoint Behavioral Analytics UI service in order to switch authentication method to SAML.



Step 7: Configure Forcepoint Behavioral Analytics for Single sign-on with SAML
Once **Sync Service** runs for the first time it will generate a bash script named **sso_config_script.sh** under **/root/configs** directory in the **Docker-host machine**

To configure Forcepoint Behavioral Analytics for single sign-on with SAML do the following steps:

1. Copy **/root/configs/sso_config_script.sh** from the **Docker-host machine** to Forcepoint Behavioral Analytics UI server
2. Make **sso_config_script.sh** executable inside Forcepoint Behavioral Analytics UI server

```
chmod +x sso_config_script.sh
```

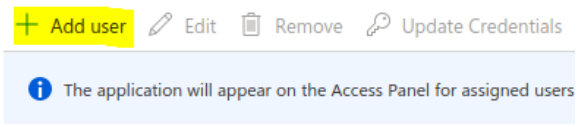
3. Execute **sso_config_script.sh** inside Forcepoint Behavioral Analytics UI server

```
sudo ./sso_config_script.sh
```

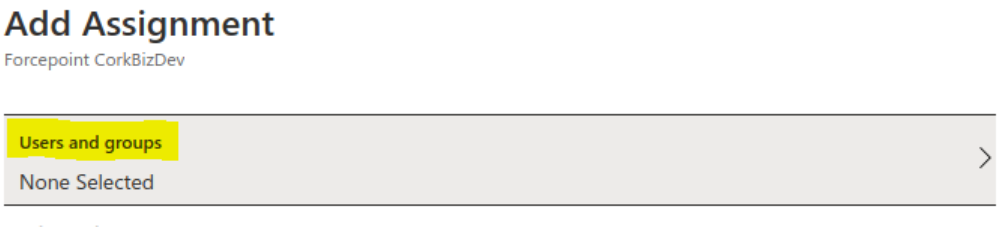
Step 8: Assign users to Azure Application
The users assigned to the Azure Application **Forcepoint Behavioral Analytics** just created will have the right access to sign-on to Forcepoint Behavioral Analytics with single sign-on (SAML).

1. Log-in to Azure portal
2. Navigate to **Azure Active Directory** > **Enterprise applications** > **All applications**
3. Click on **Forcepoint Behavioral Analytics** application
4. Select **Users and groups**

5. Click on **Add user**



6. Click on **Users and groups**



7. Select the users you want to assign to the application. You can only select users since there is no provision to map Azure AD groups into Forcepoint Behavioral Analytics.
8. Click on **Select** then **Assign**

Step 9: Assign Forcepoint Behavioral Analytics Roles to users of Forcepoint Behavioral Analytics Azure Application
This integration allows system administrators to control Forcepoint Behavioral Analytics user roles and permissions via Azure Portal.

Each Forcepoint Behavioral Analytics role is mapped to a group in Azure AD, doing this we can assign Forcepoint Behavioral Analytics permissions to Azure AD users by using group memberships. The Azure AD groups are:

- → **FP-FBA Role: admin**: User management only. Manages users, permissions, and user activity logs
- → **FP-FBA Role: analyst**: access the Review Dashboard page as well as the Job Status and Profile pages under the Settings menu.
- → **FP-FBA Role: behaviors analyst**: access the Behaviors page, Analytic Dashboard, and the Job Status and Profile pages under the Settings menu.
- → **FP-FBA Role: developer**: In-progress use, access pages that are experimental or under development
- → **FP-FBA Role: exporter**: File exporting, access functionality for exporting events
- → **FP-FBA Role: modeler**: Behavioral Modeling, create, update, and delete Models and Features (need Behaviors Analyst Role to read Behaviors page)
- → **FP-FBA Role: RAP user**: Risk-Adaptive-Protection User. Has access to dashboard, entities page, entity profile, jobs, exports, and

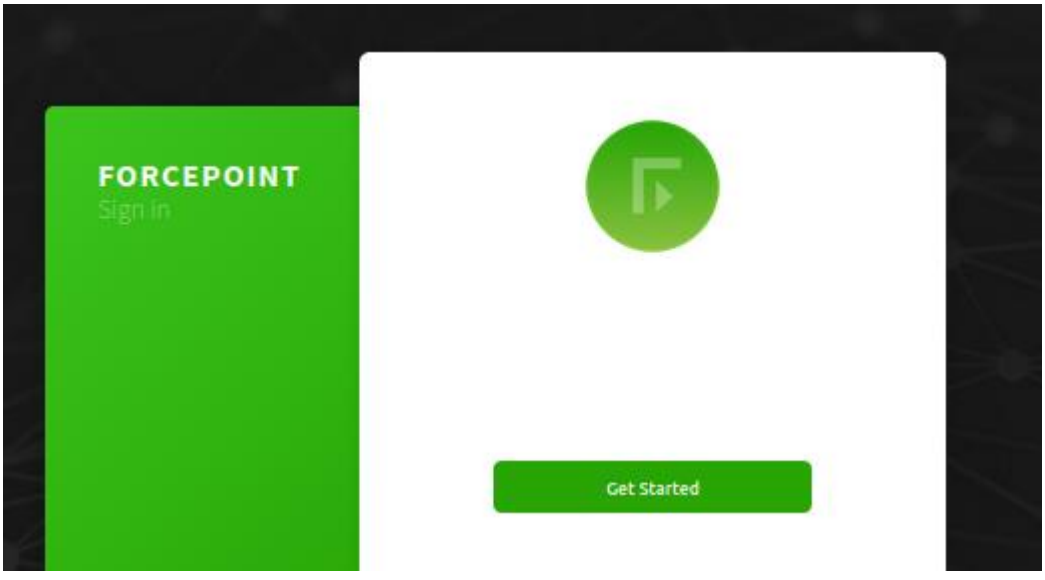
explore page.

- → **FP-FBA Role: RAP user admin**: Risk-Adaptive-Protection Admin. Has access to user management pages.
- → **FP-FBA Role: recycler**: Impending removal: access pages that are under development for removal.
- → **FP-FBA Role: restricted reviewer**: Can access and use the Review Dashboard, with the restrictions on available Actions in the Event Viewer and limited to only seeing Features that are in the users Saved Searches.
- → **FP-FBA Role: restricted user**: Restricted User. Can access the Explore page as well as the Configuration, Guide, and Profile pages under the Settings menu.
- → **FP-FBA Role: reviewer**: Can access and use the Review Dashboard
- → **FP-FBA Role: shielded user**: Can access Analytic Dashboard, Review Dashboard, Entity Timeline, Explore page as well as the Configuration and Guide, but is shielded from certain raw fields
- → **FP-FBA Role: user**: access the Explore and Entities pages as well as the Configuration, Guide, and Profile pages under the Settings menu.
- → **FP-FBA Status: Active**: Active users have all granted privileges to them. All actions are visible.
- → **FP-FBA Status: Inactive**: Inactive users cannot login, their history is visible within Forcepoint Forcepoint Behavioral Analytics and indicated with inactive label.

Note: the role **restricted user** and **shielded user** cannot be used in conjunction with other roles since these roles limit the permissions and would conflict with the other roles.

Step 10: Access on-premise Forcepoint Behavioral Analytics via Azure application
Users assigned to the Azure application **Forcepoint Behavioral Analytics** can access the UI of the on-premise Forcepoint Behavioral Analytics instance from remote, following these steps:

1. Login to https://myapplications.microsoft.com/
2. Click on **Forcepoint Behavioral Analytics**
3. Click on **Get Started** to login to Forcepoint Behavioral Analytics.



Implementation - Traditional

The solution described in this chapter requires

→ A CentOS 7.x machine able to reach the Forcepoint Behavioral Analytics services over the network. This machine will be referenced in the rest of this document with the name **host-machine**.

Step 1: Configure PostgresSQL SSL communication with host-machine

The **Sync service** interacts with the Forcepoint Behavioral Analytics PostgresSQL database to add/remove users, user roles and sessions using SSL connections to encrypt the communications between the **Sync service** components and the Forcepoint Behavioral Analytics PostgresSQL database.
The Forcepoint Behavioral Analytics Postgres machine needs to be configured to accept the SSL contamination from the **Sync Service**.

Login to **Forcepoint Behavioral Analytics postgres machine** and do the following:

1. Open **/data/ro-postgres/pg_hba.conf**

```
vi /data/ro-postgres/pg_hba.conf
```

2. Add the following line into **/data/ro-postgres/pg_hba.conf** under # **"local"**

```
hostssl    the_ui      postgres     <HOST-MACHINE_IP_ADDRESS>   255.255.255.0    trust
```

If your PostgresSQL databse name is not **the_ui**, replace **the_ui** with your database name.
If your PostgresSQL username is not **postgres**, replace **postgres** with your username.

3. Save **/data/ro-postgres/pg_hba.conf**
4. Reboot the Postgres service using the following command

```
systemctl restart postgresql-9.6.service
```

Next, login to your **host-machine** and proceed described in the following steps.

Step 2: Download Source Code

1. Tthe **fp-fba-sso-connector-azure.tar.gz** file contains the source code for the Traditional Implementation which can be downloaded from this link: https://frcpnt.com/fp-fba-sso-connector-azure-latest
2. Login to the **host-machine** as root
3. Download the **fp-fba-sso-connector-azure.tar.gz** into **/root** directory and decompress it with this command:

```
tar -zxvf  fp-fba-sso-connector-azure.tar.gz
```

Step 3: Update the config file

4. Change your directory to **/root/fp-fba-sso-connector-azure**

```
cd /root/fp-fba-sso-connector-azure
```

5. Edit **config.yml** and insert the values for the parameters listed in the file.

```
vi config.yml
```

Parameters listed in the **config.yml** file are:

- → **POSTGRES_HOST**: is the Forcepoint Behavioral Analytics Postgres server's private IP address. Change the value of this parameter to match your Forcepoint Behavioral Analytics Postgres server private IP address.
- → **POSTGRES_PORT**: is the Postgres port number, the default value is 5432. DO NOT CHANGE this value unless your Postgres service configured with a different port
- → **POSTGRES_USER_NAME**: the username for SQL Postgres, the default value is 'postgres'. DO NOT CHANGE this value unless your SQL Postgres service configured with a different username
- → **POSTGRES_DATABASE_PASSWORD**: The Postgres user's password. If you DO NOT have password remove this parameter from config.yml
- → **POSTGRES_DATABASE_NAME**: is the name of the database that contains Forcepoint Behavioral Analytics user's tables, the default name is 'the_ui. DO NOT CHANGE this value unless your Postgres service has a different database name for user's tables
- → **AZURE_APPLICATION_NAME**: is the Azure Application name, the default value for this parameter is Forcepoint Behavioral Analytics. DO NOT CHANGE the value of this parameter unless you used a different application name.
- → **USERS_SYNC_TIME_IN_MINUTES**: default value for this parameter is 3, the Sync Service will sync between Azure users and Forcepoint Behavioral Analytics users every 3 minutes.
- → **DEFAULT_FBA_PASSWORD**: a default password for the newly created users in Forcepoint Behavioral Analytics. When a new user is assigned to your Azure application and that user does not exists in Forcepoint Behavioral Analytics users database, the **Sync Service** will create that user in the Forcepoint Behavioral Analytics database and use this value as its password. Once the user logs in to Forcepoint Behavioral Analytics they can change their password. Change the value of this parameter.
- → **SSO_CONFIG_SCRIPT_PATH**: is the path for the **/sso_config_script.sh** script which will be generated by **Sync Service**. DO NOT CHANGE the value of this parameter.
- → **AZURE_ADMIN_LOGIN_NAME**: change this to your Azure administrator login-name
- → **AZURE_ADMIN_LOGIN_PASSWORD**: change this to your Azure administrator password

Step 4: Install the required packages

fp-fba-azure-installer.sh creates a systemd service called **fba_azure_sync.service** and installs the following packages

- → Golang v1.14
- → Azure CLI latest version
- → Python3

Make **fp-fba-azure-installer.sh** executable using the following command

```
chmod +x fp-fba-azure-installer.sh
```

then execute the **fp-fba-azure-installer.sh** script

```
sudo ./fp-fba-azure-installer.sh
```

Once the installation is completed move to the next step.

```
Complete!
Created symlink from /etc/systemd/system/multi-user.target.wants/fba_azure_sync.service to /etc/systemd/system/fba_azure_sync.service.
```

Step 5: Reboot host-machine

Reboot the **host-machine** and execute the following command to ensure the **Sync service** is running.

```
[root@localhost ~]# systemctl list-units | grep fba_azure_sync
```

fba_azure_sync.service	loaded active **running** Forcepoint FBA and Azure users sync

Step 6 Configure Forcepoint Behavioral Analytics for Single sign-on with SAML
Once **fba_azure_sync.service** runs for the first time, it will generate a bash script named **sso_config_script.sh** under **/root/configs** directory. This needs to be run inside the Forcepoint Behavioral Analytics machine hosting the UI service.

To configure Forcepoint Behavioral Analytics for single sign-on with SAML do the following steps:

1. Copy **/root/configs/sso_config_script.sh** to Forcepoint Behavioral Analytics UI server
2. Make **sso_config_script.sh** executable inside Forcepoint Behavioral Analytics UI server

```
chmod +x sso_config_script.sh
```

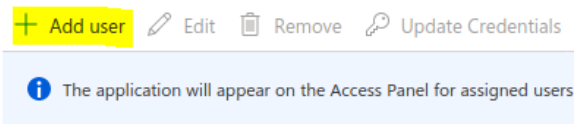
3. Execute sso_config_script.sh inside Forcepoint Behavioral Analytics UI server

```
sudo ./sso_config_script.sh
```

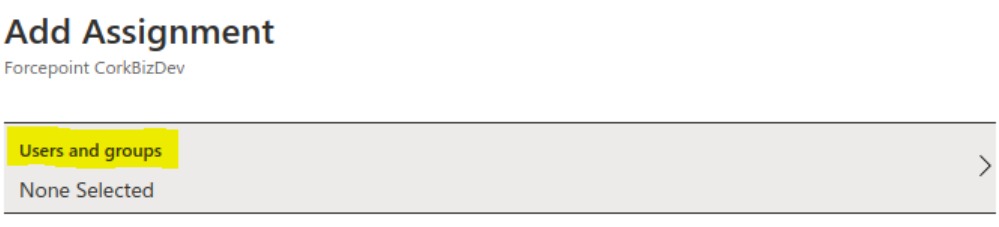
Step 7: Assign users to Azure Application
The users assigned to the Azure Application **Forcepoint Behavioral Analytics** will have access right into Forcepoint Behavioral Analytics with single sign-on (SAML).

1. Log-in to Azure portal
2. Navigate to **Azure Active Directory** > **Enterprise applications** > **All applications**
3. Click on the **Forcepoint Behavioral Analytics** application
4. Select **Users and groups**

5. Click on **Add user**



6. Click on **Users and groups**



9. Select the users you want to assign them to the application. You can only select users since there is no provision to map Azure AD groups into Forcepoint Behavioral Analytics.
7. Click on **Select** then **Assign**

Step 8: Assign Forcepoint Behavioral Analytics Roles to users of Forcepoint Behavioral Analytics Azure Application
This integration allows system administrators to control Forcepoint Behavioral Analytics user roles and permissions via Azure Portal.

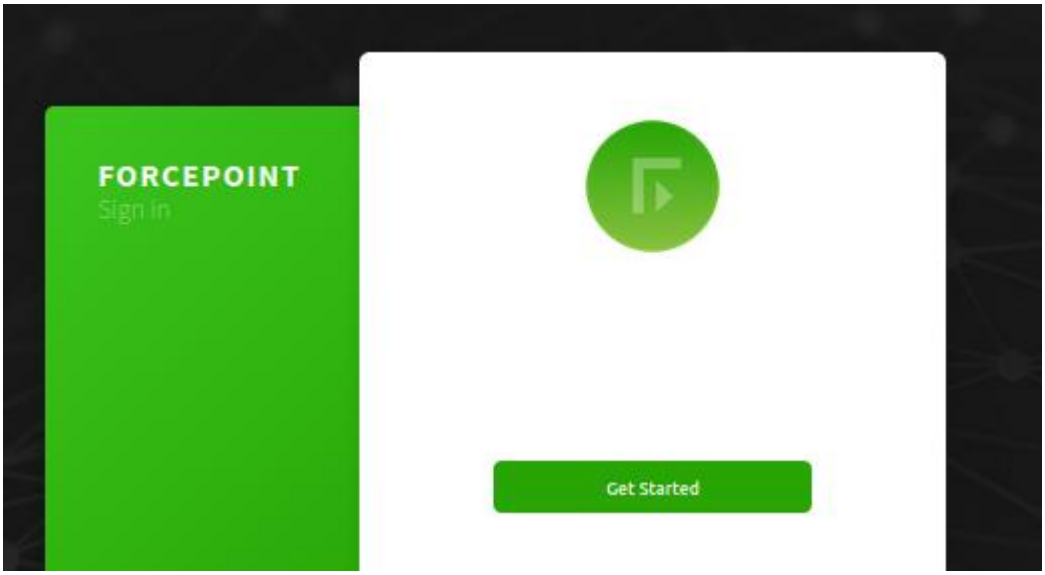
Each Forcepoint Behavioral Analytics role is mapped to a group in Azure AD, doing this we can assign Forcepoint Behavioral Analytics permissions to Azure AD users by using group memberships. The Azure AD groups are:

- → **FP-FBA Role: admin**: User management only. Manages users, permissions, and user activity logs
- → **FP-FBA Role: analyst**: access the Review Dashboard page as well as the Job Status and Profile pages under the Settings menu.
- → **FP-FBA Role: behaviors analyst**: access the Behaviors page, Analytic Dashboard, and the Job Status and Profile pages under the Settings menu.
- → **FP-FBA Role: developer**: In-progress use, access pages that are experimental or under development
- → **FP-FBA Role: exporter**: File exporting, access functionality for exporting events
- → **FP-FBA Role: modeler**: Behavioral Modeling, create, update, and delete Models and Features (need Behaviors Analyst Role to read Behaviors page)
- → **FP-FBA Role: RAP user**: Risk-Adaptive-Protection User. Has access to dashboard, entities page, entity profile, jobs, exports, and explore page.
- → **FP-FBA Role: RAP user admin**: Risk-Adaptive-Protection Admin. Has access to user management pages.
- → **FP-FBA Role: recycler**: Impending removal: access pages that are under development for removal.
- → **FP-FBA Role: restricted reviewer**: Can access and use the Review Dashboard, with the restrictions on available Actions in the Event Viewer and limited to only seeing Features that are in the users Saved Searches.
- → **FP-FBA Role: restricted user**: Restricted User. Can access the Explore page as well as the Configuration, Guide, and Profile pages under the Settings menu.
- → **FP-FBA Role: reviewer**: Can access and use the Review Dashboard
- → **FP-FBA Role: shielded user**: Can access Analytic Dashboard, Review Dashboard, Entity Timeline, Explore page as well as the Configuration and Guide, but is shielded from certain raw fields
- → **FP-FBA Role: user**: access the Explore and Entities pages as well as the Configuration, Guide, and Profile pages under the Settings menu.
- → **FP-FBA Status: Active**: Active users have all granted privileges to them. All actions are visible.
- → **FP-FBA Status: Inactive**: Inactive users cannot login, their history is visible within Forcepoint Forcepoint Behavioral Analytics and indicated with inactive label.

Note: the role **restricted user** and **shielded user** cannot be used in conjunction with other roles since these roles limit the permissions and would conflict with the other roles.

Step 9: Access on-premise Forcepoint Behavioral Analytics via Azure application
Users assigned to the Azure application **Forcepoint Behavioral Analytics** can access the UI of the on-premise Forcepoint Behavioral Analytics instance from remote, following these steps:

1. Login to https://myapplications.microsoft.com/
2. Click on **Forcepoint Behavioral Analytics**
3. Click on **Get Started** to login to Forcepoint Behavioral Analytics.


FORCEPOINT
Sign in
Get Started

Troubleshooting

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

Docker Implementation

Validate the prerequisites

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

→ Check the version of Forcepoint Forcepoint Behavioral Analytics in use is listed as compatible

Forcepoint Behavioral Analytics version 3.1.0

→ Docker images for this integration have been tested with

Docker 19.03.6

→ The docker implementation has been tested on a CentOS 7.3 machine (with at least 2 GB RAM and 20 GB disk) with docker engine installed

→ User needs sudo permissions in the docker host machine

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 docker host machine has connectivity to Forcepoint Behavioral Analytics UI machine: execute the following command on docker host machine:

ping -c 2 FBA-UI-IP

Once done check the result is similar to below:

```
PING FBA-UI-IP (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 the docker host machine has connectivity to Forcepoint Behavioral Analytics PostgreSQL machine: execute the following command on docker host machine:

ping -c 2 FBA-PostgreSQL-machine-IP

Once done check the result is similar to below:

```
PING FBA-PostgreSQL-machine-IP (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 the host machine has docker installed: Execute the following command on the host machine:
docker info

Check the first few lines of the output are similar to below:

Client:
Debug Mode: false

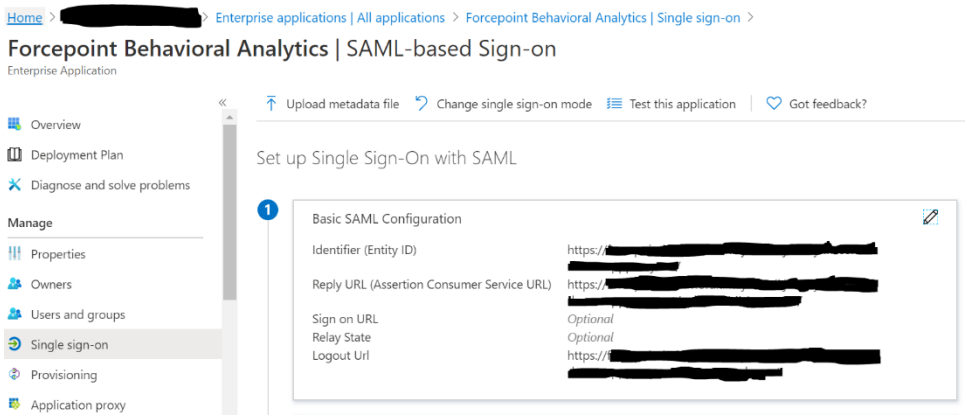
Server:
Containers: 3
Running: 2
Paused: 0
Stopped: 1
Images: 3
Server Version: 19.03.8

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:

→ Verify the integration completed with no errors: In Azure portal, go to Azure Active Directory > Enterprise Applications > Forcepoint Behavioral Analytics > Single Sign-on

There should only be SAML option in the Single sign-on as shown in the picture below:



Traditional Implementation

Validate the prerequisites

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

→ Check the version of Forcepoint Forcepoint Behavioral Analytics in use is listed as compatible

Forcepoint Behavioral Analytics version 3.1.0

→ Verify the integration is correctly operating on a CentOS 7.x machine with at least 2 GB RAM and 20 GB disk

→ User needs to be root to install dependencies

→ Check the user can download the file with the below command:

wget --content-disposition https://frcpnt.com/fp-fba-sso-connector-azure-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 connectivity to Forcepoint Behavioral Analytics UI machine: execute the following command on host machine:

ping -c 2 FBA-UI-IP

Once done check the result is similar to below:

```
PING FBA-UI-IP (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 the host machine has connectivity to Forcepoint Behavioral Analytics PostgreSQL machine: execute the following command on host machine:

ping -c 2 FBA-PostgreSQL-machine-IP

Once done check the result is similar to below:

```
PING FBA-PostgreSQL-machine-IP (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 host machine to check go is installed:

go version

Check the output is similar to below:

go version go1.14.1 linux/amd64

→ Check Azure CLI is installed: Execute following command on host machine:

az version

Check the output is similar to below:

```
{
  "Azure-cli": "2.3.1",
  "Azure-cli-command-modules-nspkg": "2.0.3",
  "Azure-cli-core": "2.3.1",
  "Azure-cli-nspkg": "3.0.4",
  "Azure-cli-telemetry": "1.0.4",
  "extensions": {}
```

→ Check python3.6 is installed: Execute following command on host machine:

python3 --version

Check the output is similar to below:

Python 3.6.x

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 the sync service is running properly by executing this command:

systemctl list-units | grep fba_azure_sync

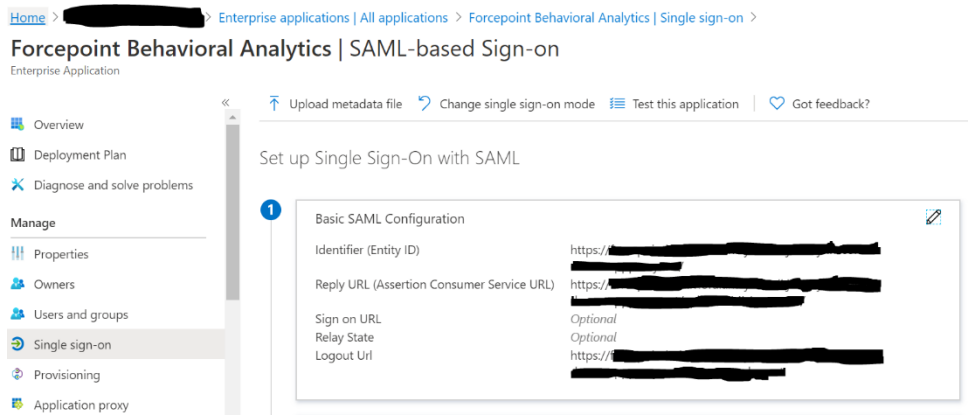
Verify the output is similar to below:

```
[root@localhost ~]# systemctl list-units | grep fba_azure_sync

fba_azure_sync.service                    loaded active running   Forcepoint FBA and Azure users sync
```

→ Verify the integration completed with no errors: In Azure portal, go to Azure Active Directory -> Enterprise Applications -> Forcepoint Behavioral Analytics -> Single Sign-on

There should only be SAML option in the Single sign-on as shown in the picture below:



Forcepoint

forcepoint.com/contact

About Forcepoint

Forcepoint is the global human-centric cybersecurity company transforming the digital enterprise by continuously adapting security response to the dynamic risk posed by individual users and machines. The Forcepoint human point system delivers risk-adaptive protection to continuously ensure trusted use of data and systems. Based in Austin, Texas, Forcepoint protects the human point for thousands of enterprise and government customers in more than 150 countries.