Forcepoint Behavioral Analytics and SecureAuth IdP

Integration Guide

Forcepoint

Integration Guide

Ultan Casey
2 April 2020
Public

Table of Contents

Version	Date	Author	Notes
0.1	17 February 2020	Ultan Casey	First draft
0.2	26 February 2020	Mattia Maggioli	Review
0.3	30 March 2020	Neelima Rai	Added troubleshooting chapter
0.4	31 March 2020	Mattia Maggioli	Review
0.5	2 April 2020	Jonathan Knepher	Review

Summary

This guide provides step by step instructions to setup an integration between the SecureAuth Identity Platform and Forcepoint Behavioral Analytics (FBA).

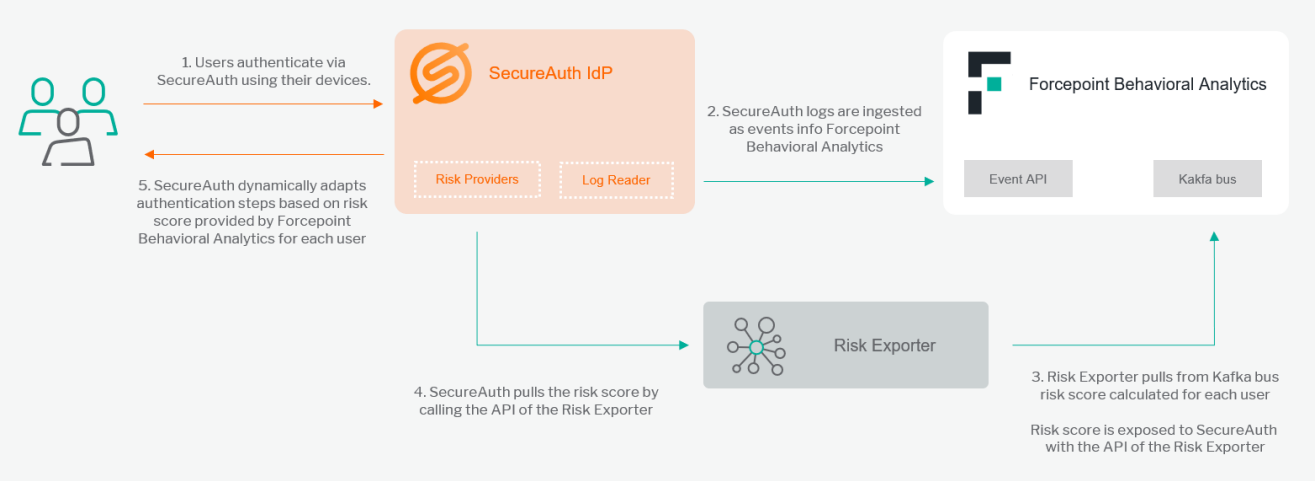
The automated real-time integration will enable the SecureAuth IdP to consume user risk levels from FBA, allowing it to react to changes in said level, whilst also sending any events and event context to FBA itself.

The code and instructions provided enable system administrators to automatically:

→Update user access permissions based on changes in user risk level.

→Ingest events into the FBA platform which provide greater context into user actions.

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 3.1.0

→SecureAuth IdP 19.07

This interoperability uses:

→**Risk Providers**, a section of the SecureAuth IdP which allows it to consume risk level information from a risk provider, in this case FBA

→**Log Exporter**, a service which reads from audit logs for each realm (application) within SecureAuth, transforms the information to relevant format and sends it to the FBA event endpoint.

→A component to export risk level information from FBA, hosted either on a Linux machine or as a docker container

The audit logs of SecureAuth are verbose in the types of events reported. Given the scope of this integration, only the events which are relevant to authentication attempts are parsed, transformed and sent to FBA.

Implementation options
Two implementation options are provided in this document

1. **Docker** – Uses a docker image where the integration component is already installed with all necessary dependencies: the user only has to edit one configuration file and run the container on an existing docker setup.
2. **Traditional** – requires manual deployment of the integration component inside a clean host machine (recommended) or an existing one, provided all requirements are satisfied.

The docker image for exporting risk level information has been tested working with

→Docker v19.03.6

while the traditional version of the risk exporter has been tested working with the following requirements

→Ubuntu 18.04

Implementation - Docker

The solution described in this chapter requires

→**fp-fba-secureauth-log-exporter-v1.zip** available at the link https://frcpnt.com/fba-secureauth-latest

→**go-risk-exporter** docker image available at the address described in the next paragraph

fp-fba-secureauth-log-exporter-v1.zip contains all files necessary to setup and run the SecureAuth **Log Exporter** which reads from all SecureAuth realms log files and transforms the information into the format required by FBA. The **Log Exporter** is designed to be deployed on the SecureAuth machine alongside SecureAuth.

go-risk-exporter is the docker image which contains the **Risk Exporter** service: this provides a REST API endpoint which can be used by SecureAuth to retrieve the risk level calculated by FBA

Setup Risk Exporter with Docker
The FBA **Risk Exporter** is provided as a docker image which can run on a docker host on premise, or on any cloud-based docker service with network access to the FBA Kafka bus.

Please note: when the risk exporter is first installed it will not contain any risk scores. This is because it consumes and stores messages containing the risk score as soon as they transit on the Kafka bus, and there will not be any risk score information until a risk score change occurs inside FBA. Therefore, we recommend running the **Risk Exporter** for some time before using the risk level inside the SecureAuth policies.

In order to setup the Risk Exporter do as follows:

1. Login to the docker registry with the following command

   docker login docker.frcpnt.com

   User: fp-integrations
   Pass: t1knmAkn19s
2. Pull the image from the registry using the following command.

   docker pull docker.frcpnt.com/go-risk-exporter

3. Create a configuration file named **exporter_config.yaml** with the following contents, editing the values as described in the Appendix A of this document.

   log_level: 'warn'
   kafka_endpoint: 'kafka-endpoint'
   kafka_port: '9093'
   ssl_key_password: 'password'

   ```
   log_level: 'warn'
   kafka_endpoint: 'kafka-endpoint'
   kafka_port: '9093'
   ssl_key_password: 'password'
   ```

   Upload the file to a location where it can be accessed over http/https (e.g. a web server, an AWS S3 bucket, Azure Blob Storage): this will make sure the configuration is not lost in case the docker container is decommissioned.
4. Retrieve the URL to the file and store it in a safe location as it will be used in the next steps of this document
5. Place the SSL certs of the Kafka bus used within FBA in a directory called **certs** inside the docker host. Make sure the files are named exactly as follows: **client.cer, client.key, client-ca.cer.**
6. Run the **Risk Exporter** with the following command, replacing **<SSL_CERTS_DIR>** with the actual path to the directory containing the SSL certs and **<YOUR_CONFIG_URL>** with the URL of the config file

   *docker run -p 8080:8080 -v **<SSL_CERTS_DIR>**:/certs --mount type=volume,src=dbdata,dst=/data -v type=volume,src=configdata,dst=/config -e CONFIG_URL=**<YOUR_CONFIG_URL>** --name go-risk-exporter docker.frcpnt.com/go-risk-exporter*

7. To confirm it is running navigate to the following URL **http://<your_domain>:8080/risk/score/<test_user>.** The response received should resemble the following one

   {"user_id":"Test.User","first_added":"0000-00-00T00:00:00.000Z","last_update":"0000-00-00T00:00:00.000Z","risk_level":-1}

8. Take note of the public IP address of the machine on which the risk exporter has been installed, along with the public port mapped using the docker run command since they will be needed in the configuration of SecureAuth.

Implementation - Traditional

The solution described in this chapter requires the following files

→**fp-fba-secureauth-log-exporter-v1.zip** available at this link https://frcpnt.com/fba-secureauth-latest

→**fp-fba-risk-exporter-v1.zip** available at this link https://frcpnt.com/fba-risk-exporter-latest

fp-fba-secureauth-log-exporter-v1.zip contains all files necessary to setup and run the SecureAuth **Log Exporter** which reads from all SecureAuth realms log files and transforms the information into the format required by FBA. The **Log Exporter** is designed to be deployed on the SecureAuth machine alongside SecureAuth.

fp-fba-risk-exporter-v1.zip contains all files necessary to setup and run FBA **Risk Exporter** service. This is designed to be deployed on an Ubuntu 18.04 machine with systemd services.

Setup Risk Exporter with Systemd

1. Create a folder named **certs** in the root of the machine. Within this directory place the kafka bus SSL certs. Make sure the files are named exactly as follows: **client.cer, client.key, client-ca.cer**
2. Download and unpack the **fp-fba-risk-exporter-v1.zip** file in the **downloads** directory
3. Navigate to the config directory in the unpacked files and open **config.yaml**.

4. Update the kafka endpoint and port to match that of the kafka bus utilised by your FBA installation, the ssl key password to the password of the SSL certificates and log level to the level of log events from the exporter you wish to have exported

```
log_level: 'warn'
kafka_endpoint: 'kafka-endpoint'
kafka_port: '9093'
ssl_key_password: 'password'
```

5. Return to the root of the folder of the extracted files and run the following command to install the **Risk Exporter** as a service

   sudo ./install.sh

6. To confirm it is running navigate to the following URL.

   *http://<**risk_exporter_hostname_or_IP**>:8080/risk/score/<test_user>*

   Make sure to replace the hostname or IP of the machine where the **Risk Exporter** is hosted**.** The response received should be similar to this one

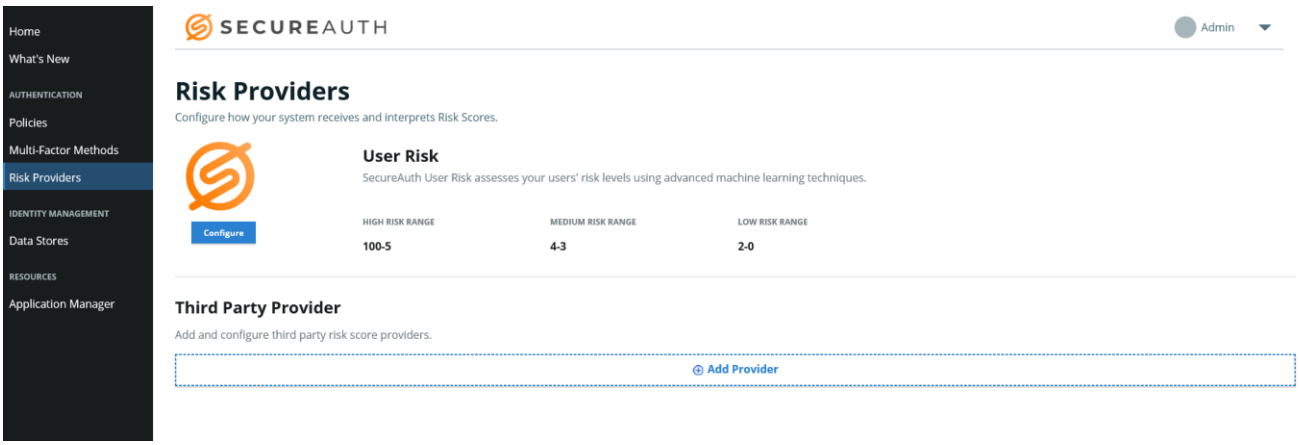
   {"user_id":"Test.User","first_added":"0000-00-00T00:00:00.000Z","last_update":"0000-00-00T00:00:00.000Z","risk_level":-1}

7. Take note of the public IP address of the machine on which the risk exporter has been installed, along with the public port mapped using the docker run command since they will be needed in the next chapter.

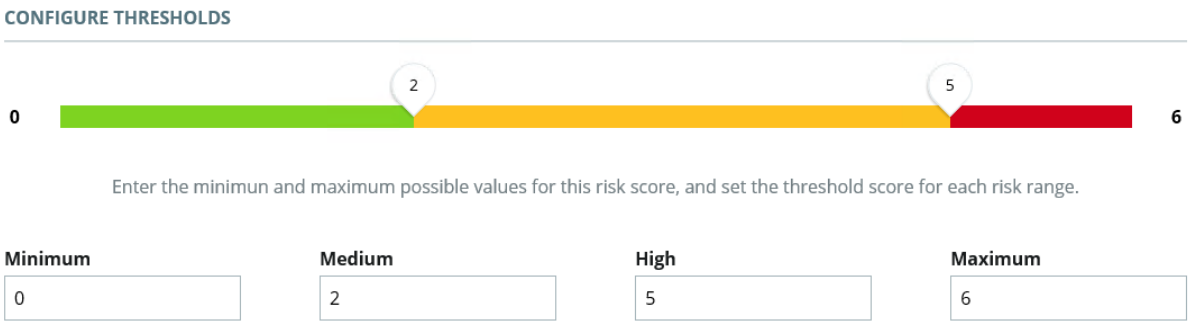
Configure SecureAuth to consume exported Risk Data

SecureAuth IdP provides a built-in system to consume risk data from external endpoints and configure policies which rely on these data. We will now configure SecureAuth to consume risk data from the **Risk Exporter**.

1. From the SecureAuth administrator dashboard, navigate to the **Risk Providers** section linked in the sidebar to the left



2. Select **Add Provider**
3. Set the minimum and maximum thresholds to **0** and **6** respectively. The medium and high levels can be configured as desired within that range



4. Set **Score Provider Name** to FBA and the **base URL** to the hostname/IP address and port of the **Risk Exporter,** for example

   http://<IP_OF_RISK_EXPORTER_MACHINE>:8080

5. In the **Get Profile** field enter **/risk/score/{username}**

   The {username} field is used by SecureAuth IdP, and will replace {username} with that of the user currently having their risk score checked.

6. The next fields are not involved in the configuration of SecureAuth and the Risk Exporter, but cannot be left empty: for **authentication method** select **Basic**, set **username** to **FBA** and **password** to **Password**
7. Set **Risk Score Attribute** to **Authenticated User ID**
8. In the **Risk Score JSON Path** field enter **{risk_level}**
9. Click **Save**
10. From the SecureAuth administrator dashboard, click the drop-down icon next to the Admin username then click **Go to Classic Experience**: you will be presented with a list of the configured Applications/Realms.
11. Select the tools button on the menu bar and from the dropdown presented click on Decrypt Web Config.
12. Enable the checkbox next to **Analyze API.**
13. Click Decrypt.
14. Navigate to **D:\Secureauth\AnalyzeAPI** and open the **web.config** file.
15. For each realm you wish to enable the Risk Score provider for, insert the following line at the end of the **<appSettings>** section replacing YOUR_REALM_ID_NUMBER with the ID of the realm.

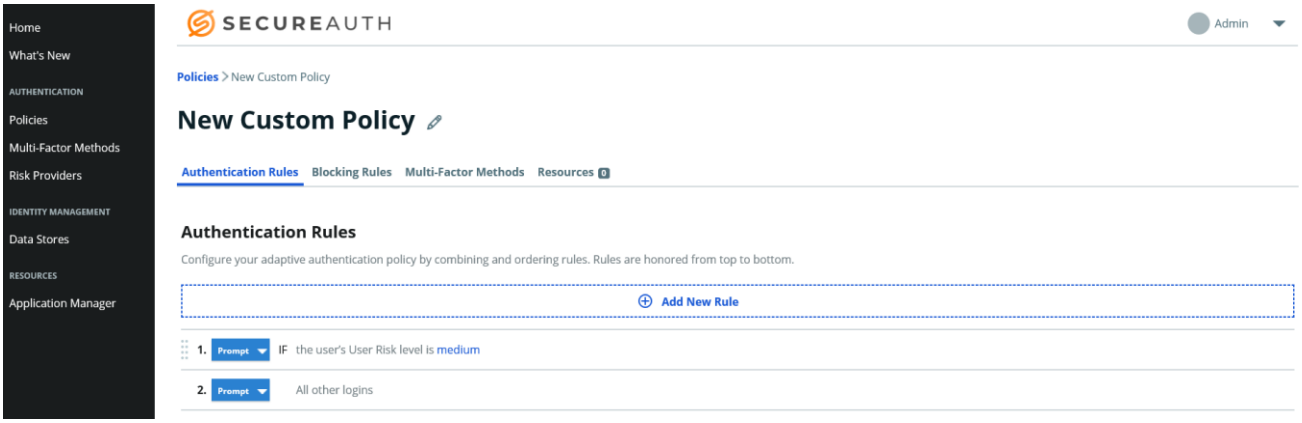
    <add key="RiskSwitch_Realm**<YOUR_REALM_ID_NUMBER>**_Provider1" value="False" />

16. Save the **web.config** file.

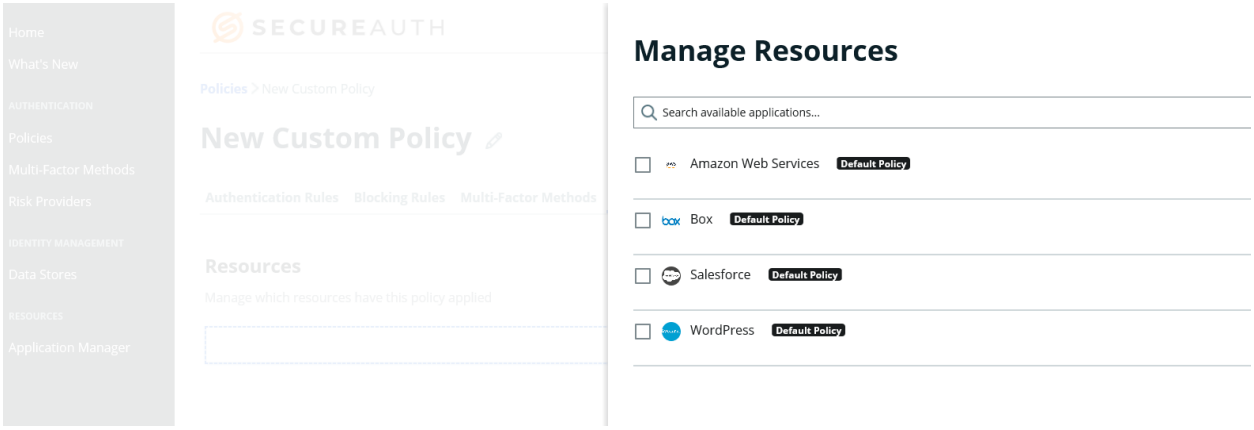
Configure risk-adaptive policies

With the SecureAuth IdP now configured to consume risk data from the **Risk Exporter**, we can now create policies which rely on this data and adjust authentication workflow based on the risk level. The following is a guide on how to configure an example setup.

1. From the SecureAuth administrator dashboard, navigate to the **Policies** section linked in the sidebar to the left.
2. Select **Add New Policy.**
3. Click **Add New Rule** and select **User Risk**



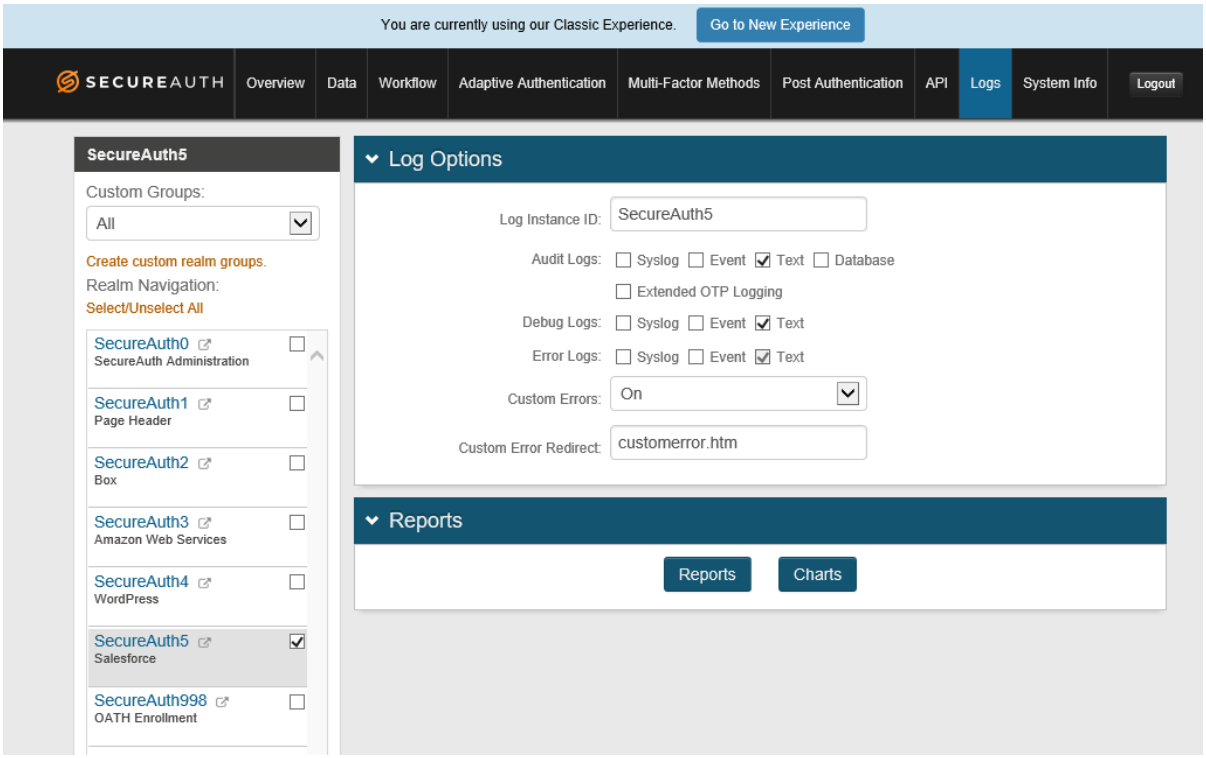
4. Configure this rule to be **Prompt** if user risk level is **Medium**: this ensures the user must login with both username and password along with a second authentication method.
5. Remove any rules added by default to **New Policies** such as **Skip if user has not moved faster than 575mph** then click **Save.**
6. Navigate to **Blocking Rules**. Click again **Add New Rule** and select **User Risk**: this should be set to **Block** if user risk level is **High**. Click **Save** once done. Make sure your changes are saved properly by refreshing the page.
7. Under **Resources** select the applications/realms (managed by SecureAuth) whose authentication will be processed according to the new policy.



Enable Log exporting on SecureAuth Realms

Forcepoint Behavioral Analytics provides context to user actions and generates risk scores based on user events. Scoring user risk requires events to be ingested into the platform. To make sure SecureAuth IdP events are sent to FBA we must first ensure that SecureAuth is generating log files.

1. From the SecureAuth administrator dashboard, click the drop-down icon next to the Admin username then click **Go to Classic Experience**: you will be presented with a list of the configured Applications/Realms
2. Click on the name of the app you wish to enable logs for. For better visibility on activities across applications we recommend enabling logs for as many applications as possible
3. Select **Logs** on the navigation bar



4. Ensure that **Text** is enabled for **Audit Logs** and click **Save**

Repeat the steps above for each realm you wish to have logging enabled.

Install LogReader to ingest logs into FBA

Now that logging for our realms is enabled, we can configure and install the **LogReader** application which parses the log files for each realm, transforms the data into the relevant format and sends it to FBA. The **LogReader** is designed to be installed on the machine running SecureAuth IdP.

1. Unpackage the **fp-fba-secureauth-log-exporter-v1.zip** archive file into a folder called **LogReader** on the root of the C: drive of the machine running SecureAuth IdP.

   Open the **config.yaml** file which is located in the config folder and edit the values for each parameter so that they match your current setup. If the LogReader is installed in the same machine of SecureAuth IdP, the **admin_url** should not be changed. Modify the log level if you desire more verbose logging from the LogReader.

```
admin_url: 'https://localhost'
fba_endpoint: 'https://fba-endpoint.address.com'
fba_port: '9000'
rose_endpoint: 'https://rose-endpoint.address.com'
rose_port: '9500'
mds_endpoint: 'https://mds1-endpoint.address.com'
mds_port: '8080'
realm_dir: './config/realms'
log_level: 'info'
ignore_ssl: true
```

2. Right click on the **install.bat** file and select **Run as administrator**
3. Enter your account password and hit enter: the install will proceed and complete within a few seconds

Once installed, to verify that the LogReader is running successfully navigate to the **/config/realms** directory: shortly after launching this will be populated with **yaml** config files for each realm/application.

Authentication logs of SecureAuth will be visible as events inside FBA within a few minutes.

Troubleshooting

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

Traditional Implementation

Validate the prerequisites

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

→ Check the versions of Forcepoint FBA and SceureAuth in use are listed as compatible

Forcepoint Behavioral Analytics 3.1.0

SecureAuth IdP 19.07

→ For installing the **Risk Exporter** via traditional implementation, it needs to be operating on a clean Ubuntu 18.04 machine

→ User needs sudo permissions for installing the **Risk Exporter**

→ User needs to login with an administrator user account on Windows machine for this integration

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

wget --content-disposition https://frcpnt.com/fba-secureauth-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 user has network connectivity to FBA: execute the following command on machine:

ping -c 2 example-fba.url

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

```
PING example-fba.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 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 Risk Exporter is configured and running as expected: Navigate to the following URL:

http://<risk_exporter_hostname_or_IP>:8080/risk/score/<test_user>

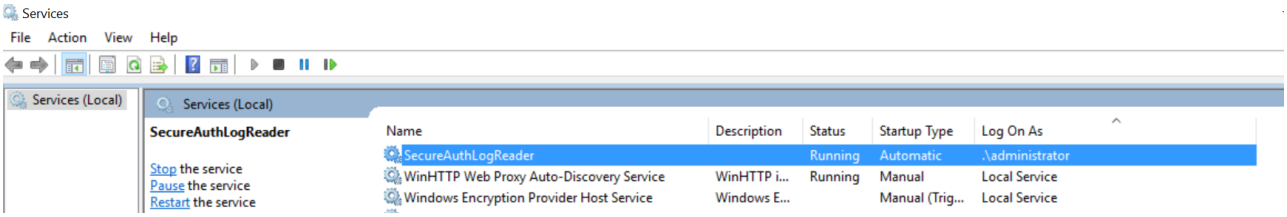
Replace <risk_exporter_hostname_or_IP> with the correct Risk Exporter hostname and <test_user> with any username managed by SecureAuth, then check the result is similar to below:

{"user_id":"Test.User","first_added":"2020-02-19T09:09:59.999Z","last_update":"2020-02-19T09:09:59.999Z","risk_level":2}

→ Check the **LogReader** is configured and running as expected:

Open ‘services’ in the windows machine hosting the SecureAuth appliance. Once in Services, check the service SecureAuthLogReader has ‘Running’ status.

Check the result is similar to below:



Docker Implementation

Validate the prerequisites

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

→ For installing the **Risk Exporter** via docker implementation, it needs to be operating on a Linux based machine with an existing docker host.

→ To install the LogReader component on the SecureAuth Windows machine the user must be logged into an account with administrator privileges. This ensures that correct permissions are supplied to it during the installation.

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

wget --content-disposition https://frcpnt.com/fba-secureauth-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 user has network connectivity to FBA: execute the following command on machine:

ping -c 2 example-fba.url

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

```
PING example-fba.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 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 **Risk Exporter** is configured and running as expected: Navigate to the following URL:

http://<risk_exporter_hostname_or_IP>:8080/risk/score/<test_user>

Replace <risk_exporter_hostname_or_IP> with the correct Risk Exporter hostname and <test_user> with any username managed by SecureAuth, then check the result is similar to below:

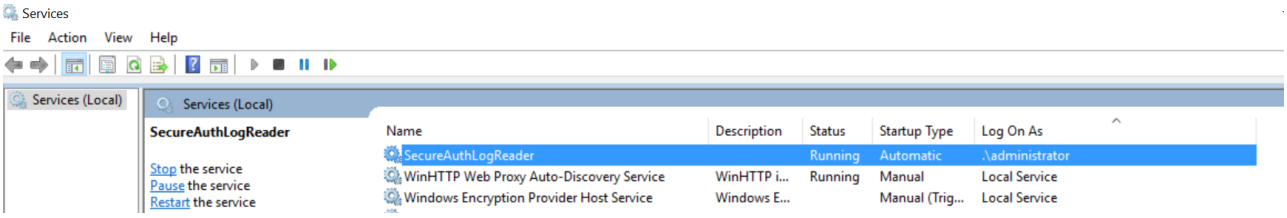
{"user_id":"Test.User","first_added":"2020-02-19T09:09:59.999Z","last_update":"2020-02-19T09:09:59.999Z","risk_level":2}

→ Check the **LogReader** is configured and running as expected:

Open ‘services’ in the windows machine hosting the SecureAuth appliance. Once in Services, check the service

SecureAuthLogReader has ‘Running’ status.

Check the result is similar to below:



Appendix A – Config File

Below is a table indicating the values required for the config file utilized by the LogReader. This is in the **/config/** directory.

Field	Examples	Notes
admin_url	https://localhost	The URL to the admin dashboard of the SecureAuth IdP. Default for this is https://localhost
fba_endpoint	https://fba-api.example.com	The FQDN or IP address of the FBA Streaming Ingest Public API
fba_port	9001	Port number for the FBA Streaming Ingest Public API
rose_endpoint	https://fba-rose.example.com	The FQDN or IP address of the FBA Rose API. Used for retrieving monitored users
rose_port	9002	Port number for the FBA Rose API
mds_endpoint	https://fba-mds.example.com	The FQDN or IP address of the FBA MDS API. This is needed to set user entities to “monitored” within so new users (not previously known) will be monitored automatically
mds_port	9003	Port number for the FBA MDS API
realm_dir	./config/realms	Leave as default. This is the directory in which realm configuration files are stored
log_level	info	The level of logging of the LogReader. Logs into the events.log file will have a different level of details
ignore_ssl	true	Sets whether the LogReader should ignore issues with SSL certs and FBA, for example when using self-signed certs.

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.