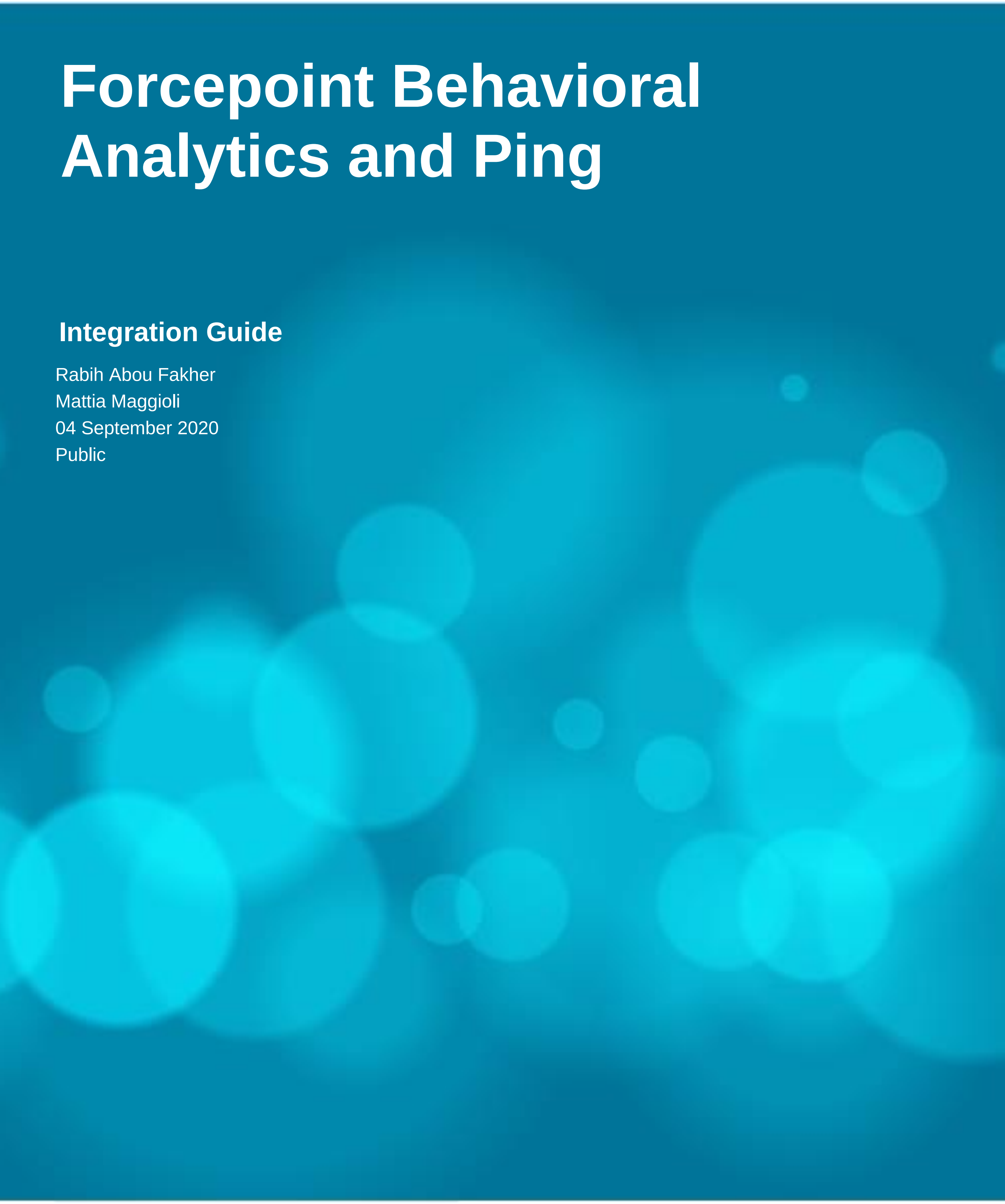


Forcepoint Behavioral Analytics and Ping

Integration Guide

Rabih Abou Fakher
Mattia Maggioli

04 September 2020

Public

Version	Date	Author	Notes
0.1	22 October 2019	Mattia Maggioli	First draft
0.2	23 October 2019	Mattia Maggioli	Updated with risk level mapping to IdP policies
0.3	23 October 2019	Jonathan Knepher	Review
0.4	24 October 2019	Audra Simons	Review
0.5	24 October 2019	Jonathan Knepher	Review
0.6	25 October 2019	Mattia Maggioli	Updated Risk Export chapter
0.7	21 November 2019	Mattia Maggioli	Added MFA policy with PingID, instructions to ingest failed login attempts and examples
0.8	22 January 2020	Rabih Abou Fakher	Updated package names
0.9	28 February 2020	Rabih Abou Fakher	Updated risk exporter chapter and Ingesting failed logins chapter
1.0	09 April 2020	Neelima Rai	Updated with contents for Ping Federate 10
1.1	04 September 2020	Mattia Maggioli	Minor updates

Summary

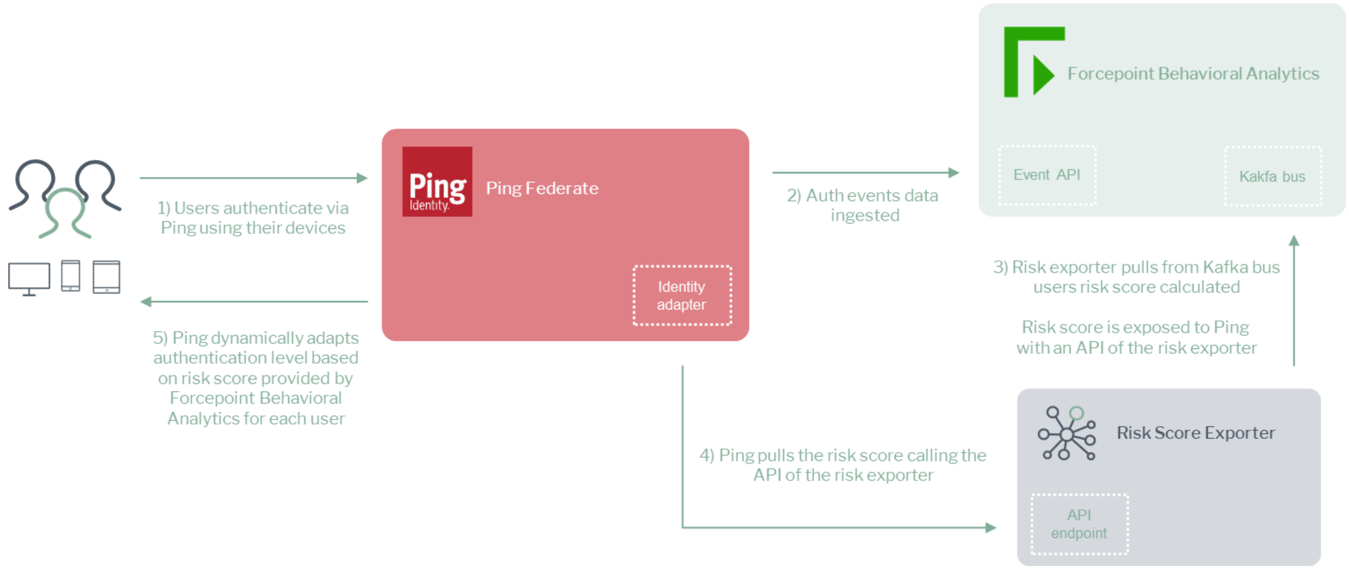
This guide provides step by step instructions to configure Forcepoint Behavioral Analytics and Ping Federate to pass risk level and login / event information.

The code and instructions provided enable administrators to automatically:

- Export events from Ping Federate into Forcepoint Behavioral Analytics
- Provide the risk level calculated by Forcepoint Behavioral Analytics for each user to Ping Federate
- Adjust authentication policies applied by Ping Federate to users based on their risk level

This interoperability enriches visibility into user activities, enhances risk scoring, and enables risk-adaptive authentication policy for Ping Federate users based on the intelligence provided by Forcepoint Behavioral Analytics.

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


Forcepoint Behavioral Analytics
Event API
Kakfa bus
Ping Identity
Ping Federate
Identity adapter
1) Users authenticate via Ping using their devices
2) Auth events data ingested
3) Risk exporter pulls from Kafka bus users risk score calculated
Risk score is exposed to Ping with an API of the risk exporter
4) Ping pulls the risk score calling the API of the risk exporter
5) Ping dynamically adapts authentication level based on risk score provided by Forcepoint Behavioral Analytics for each user
Risk Score Exporter
API endpoint

Caveats

These implementation instructions are tested with the following product versions:

- Forcepoint Behavioral Analytics 3.1.0
- Ping Federate 9.2 and 10.0

This interoperability is based on

- an **Identity Adapter** provided by Ping Identity, which exports successful events from Ping

Federate to Forcepoint Behavioral Analytics via the Forcepoint Streaming Ingest Public API

- a script which parses audit logs of Ping Federate to extract information of failed login attempts and sends them to Forcepoint Behavioral Analytics

Audit logs of Ping Federate only provide information related to known users (i.e. users who belong to the directory being used in the customer environment), so verbosity is limited to relevant information for ingestion into Forcepoint Behavioral Analytics.

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:

- monitoring of the scripts, services and applications involved in the solution

Implementation options

Two implementation options are provided in this document

- **Docker** – leverages 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
- **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 the following requirements

- Docker 19.03.5
- The docker host machine should meet the minimum hardware requirements of 2GB RAM, 20GB free storage, 64bit version only for the operating system

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

- CentOS 7 or Ubuntu 18.04 (64bit versions only) with at least 2GB RAM, 20GB free storage

The files needed to setup the integration are available at the following links:

- **fp-riskexporter-api-v1.tar.gz** available at https://frcpnt.com/fp-riskexporter-api-latest

- **fp-fba-failed-logins-importer-ping-v1.tar.gz** available at https://frcpnt.com/fp-fba-failed-logins-importer-ping-latest
- **ping-connector.tar.gz** available at https://frcpnt.com/ping-connector-latest

Step 1 – Setup Risk Exporter

Risk Exporter provides a REST API endpoint used by the Ping connector to retrieve the risk level calculated by Forcepoint Behavioral Analytics. It is provided as a tar file with one associated configuration file.

Risk Exporter is deployed to a Linux machine with a working network connectivity to Forcepoint Behavioral Analytics and from Ping Federate, typically within the same infrastructure hosting both components.

Implementation – Traditional

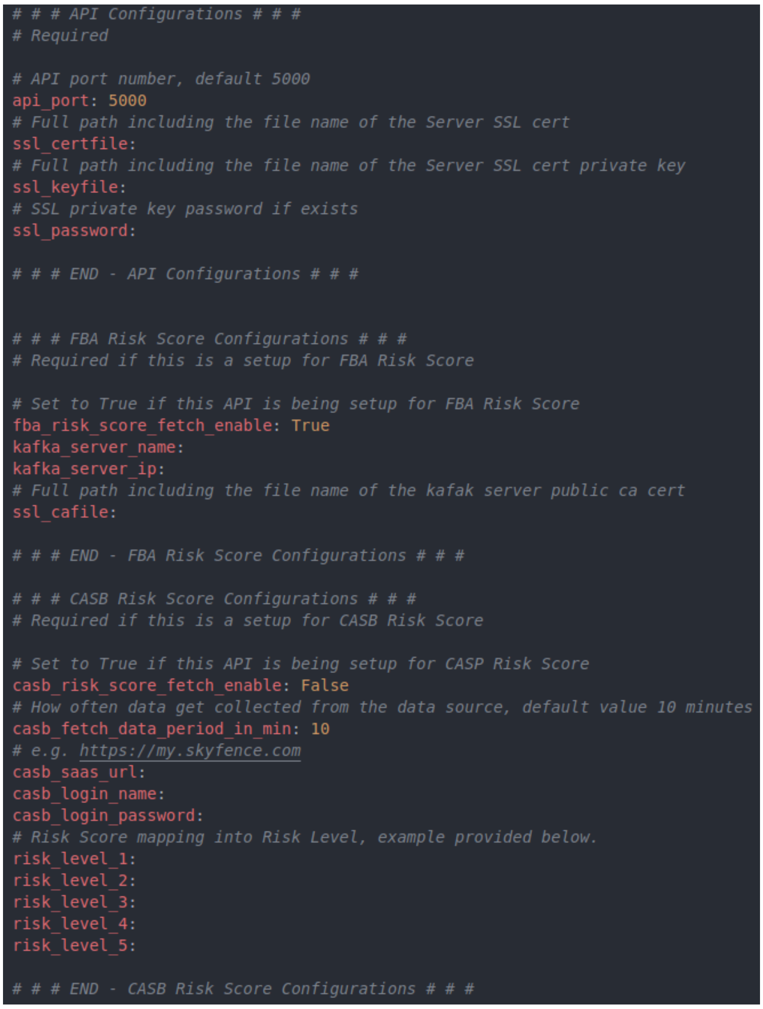
1. Unpack the **fp-riskexporter-api-v1.tar.gz** file. The examples use the location **/opt/fp-riskexporter-api-v1/**, however, the administrator can change to another location if desired.

   ```
   wget --content-disposition https://frcpnt.com/fp-riskexporter-api-latest
   tar -zxvf fp-riskexporter-api-v1.tar.gz -C /opt/
   ```

2. Install script below will install the system prerequisites, run with a user with administrative privileges.

   ```
   /opt/fp-riskexporter-api/deploy/install.sh
   ```

3. Edit the **cfg.yml** file located in /opt/fp-riskexporter-api-v1 and change the values to match the hostnames/IP addresses, ports, paths, filenames, and credentials in your current environment relevant to your setup.



```
# # # API Configurations # # #
# Required

# API port number, default 5000
api_port: 5000
# Full path including the file name of the Server SSL cert
ssl_certfile:
# Full path including the file name of the Server SSL cert private key
ssl_keyfile:
# SSL private key password if exists
ssl_password:

# # # END - API Configurations # # #


# # # FBA Risk Score Configurations # # #
# Required if this is a setup for FBA Risk Score

# Set to True if this API is being setup for FBA Risk Score
fba_risk_score_fetch_enable: True
kafka_server_name:
kafka_server_ip:
# Full path including the file name of the kafak server public ca cert
ssl_cafile:

# # # END - FBA Risk Score Configurations # # #

# # # CASB Risk Score Configurations # # #
# Required if this is a setup for CASB Risk Score

# Set to True if this API is being setup for CASP Risk Score
casb_risk_score_fetch_enable: False
# How often data get collected from the data source, default value 10 minutes
casb_fetch_data_period_in_min: 10
# e.g. https://my.skyfence.com
casb_saas_url:
casb_login_name:
casb_login_password:
# Risk Score mapping into Risk Level, example provided below.
risk_level_1:
risk_level_2:
risk_level_3:
risk_level_4:
risk_level_5:

# # # END - CASB Risk Score Configurations # # #
```

4. Setup script below will install the program prerequisites and run the program, run with a user with administrative privileges.

```
/opt/fp-riskexporter-api/deploy/setup.sh
```

Implementation – Docker

1. Login into docker repository, you'll be asked to enter your username and password (provided below):

```
docker login docker.frcpnt.com
User: fp-integrations
Pass: t1knmAkn19s
```

2. Run the below to download the image

```
docker pull docker.frcpnt.com/fp-riskexporter-api
```

3. Create a new file named **cfg.yml** and insert the following contents

```
# API port number, default 5000
api_port: 5000
# Full path including the file name of the Server SSL cert in the docker container (leave as it is)
ssl_certfile: /app/fp-riskexporter-api/certs/server.crt
# Full path including the file name of the Server SSL cert private key in the docker container (leave as it is)
ssl_keyfile: /app/fp-riskexporter-api/certs/server.key
# SSL private key password if exists
ssl_password:
# Set to True if this API is being setup for FBA Risk Score
fba_risk_score_fetch_enable: True
kafka_server_name:
kafka_server_ip:
# Full path including the file name of the kafak server public ca cert in the docker container (leave as it is)
ssl_cafile: /app/fp-riskexporter-api/certs/kafka-ca.crt
```

4. Run the container with either one of the following commands, depending on your scenario

- if **cfg.yml** file is located locally, then run the command below, replacing the red part with the full path of the cfg.yml file and the SSL certifcates:

```
docker run --detach \
--name fp-riskexporter-api \
--publish 5000:5000 \
--volume <cfg.yml-full-path>:/app/fp-riskexporter-api/cfg.yml \
--volume <server.crt-full-path>:/app/fp-riskexporter-api/certs/server.crt \
--volume <server.key-full-path>:/app/fp-riskexporter-api/certs/server.key \
--volume <kafka-ca.crt-full-path>:/app/fp-riskexporter-api/certs/kafka-ca.crt \
--volume RiskScoreDBVolume:/app/fp-riskexporter-api/db \
docker.frcpnt.com/fp-riskexporter-api
```

- if **cfg.yml** file is hosted in a remote location,then run the command below, replacing the red part with the URL of the cfg.yml file to download and the full path of the SSL certificates:

```
docker run --detach \
--name fp-riskexporter-api \
--publish 5000:5000 \
--env CONFIG_FILE_URL_LOCATION=<config-file-url> \
--volume <server.crt-full-path>:/app/fp-riskexporter-api/certs/server.crt \
--volume <server.key-full-path>:/app/fp-riskexporter-api/certs/server.key \
--volume <kafka-ca.crt-full-path>:/app/fp-riskexporter-api/certs/kafka-ca.crt \
--volume RiskScoreDBVolume:/app/fp-riskexporter-api/db \
docker.frcpnt.com/fp-riskexporter-api
```

Step 2 – Setup Ping Federate

Ping Federate normally uses **Identity Provider** (IdP) as system entities that authenticate users and provide identity attributes to Ping. In our case, we will leverage an IdP to communicate with Forcepoint Behavioral Analytics in both ways (sending event data to Forcepoint Behavioral Analytics, passing risk level back to Ping Federate).

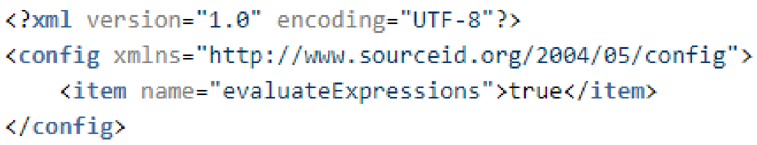
1. Unpack **ping-connector.tar.gz** and place the files in the location specified in the following table (change the red parts to match your setup)

   wget --content-disposition https://frcpnt.com/ping-connector-latest

   tar -zxvf ping-connector.tar.gz

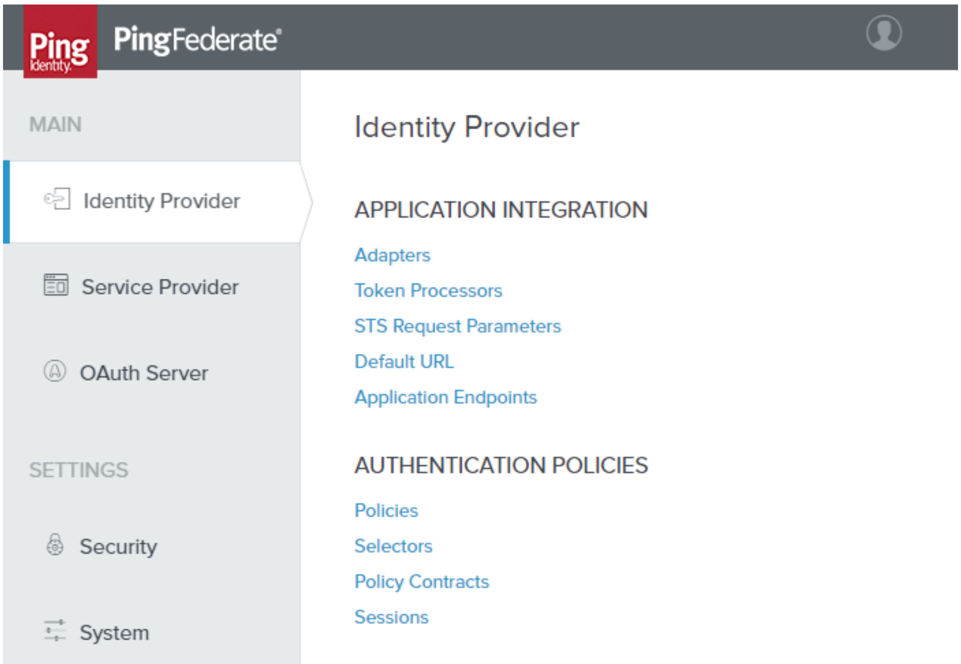
File	Target location
pf.plugins.generic-device-risk-adapter.jar	/<pingfed-home>/pingfederate/server/default/deploy
devicerisk.html.form.login.template.html	/<pingfed-home>/pingfederate/server/default/conf/template
devicerisk.min.capture.template.html	/<pingfed-home>/pingfederate/server/default/conf/template
client.min.js	/<pingfed-home>/pingfederate/server/default/conf/template/assets/scripts

2. Edit the file /<pingfed-home>/pingfederate/server/default/data/config-store/org.sourceid.common.ExpressionManager.xml and change “evaluateExpressions” to true



```
<?xml version="1.0" encoding="UTF-8"?>
<config xmlns="http://www.sourceid.org/2004/05/config">
    <item name="evaluateExpressions">true</item>
</config>
```

3. Restart the Ping service to load the new files, see this page on Ping website for options available: https://support.pingidentity.com/s/document-item?bundleId=pingfederate-92&topicId=gettingStartedGuide%2Fpf_t_startAndStopPingfederate.html
4. Login to the Ping Federate console
5. Select Identity Provider > Integration > Adapters



Identity Provider

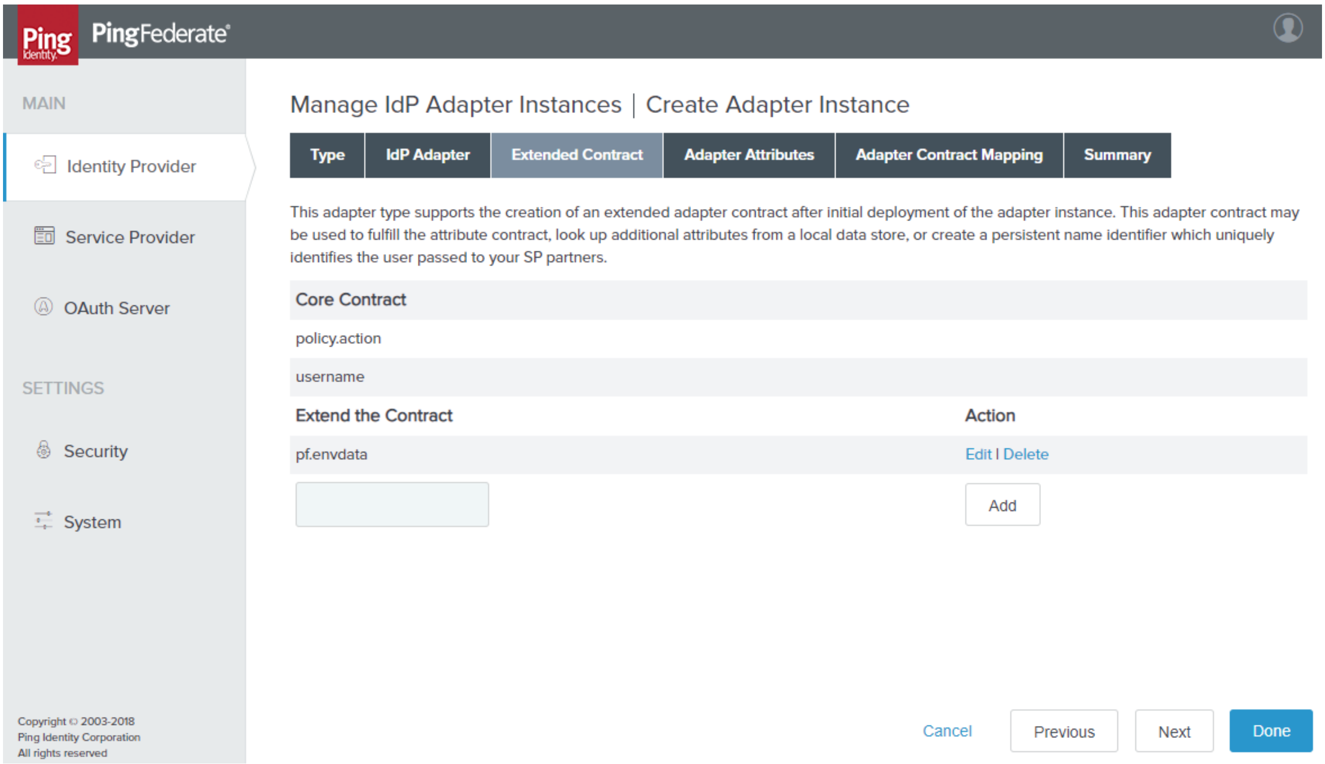
APPLICATION INTEGRATION

Adapters
Token Processors
STS Request Parameters
Default URL
Application Endpoints

AUTHENTICATION POLICIES

Policies
Selectors
Policy Contracts
Sessions

6. Select the existing **HTMLFormSimplePCV** adapter
7. Select **Extended Contract** and add **pf.envdata** in **Extend the Contract** field, then select **Done** to save the configuration of this IdP. Click **Save** on the next page to save these changes.



Manage IdP Adapter Instances | Create Adapter Instance

This adapter type supports the creation of an extended adapter contract after initial deployment of the adapter instance. This adapter contract may be used to fulfill the attribute contract, look up additional attributes from a local data store, or create a persistent name identifier which uniquely identifies the user passed to your SP partners.

Core Contract	
policy.action	
username	
Extend the Contract	**Action**
pf.envdata	Edit I Delete

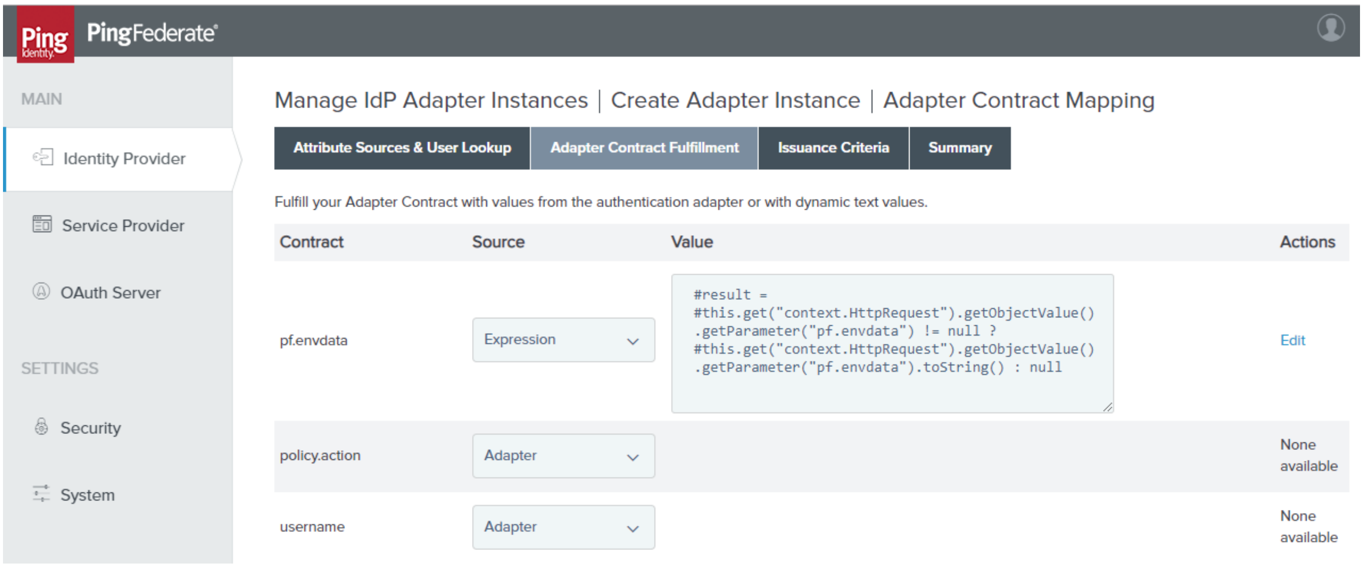
Copyright © 2003-2018
Ping Identity Corporation
All rights reserved

8. Go to Identity Provider > Integration > Adapters > HTMLFormSimplePCV > Adapter Contract

Mapping, then click Configure Adapter Contract

9. Click on **Adapter Contract Fulfillment,** change the Source for **pf.envdata** to **Expression** (as shown in the screenshot below) and add the following expression into the **Value** area as one line.

   #result = #this.get("context.HttpRequest").getObjectValue().getParameter("pf.envdata") != null ? #this.get("context.HttpRequest").getObjectValue().getParameter("pf.envdata").toString() : null



Manage IdP Adapter Instances | Create Adapter Instance | Adapter Contract Mapping

Fulfill your Adapter Contract with values from the authentication adapter or with dynamic text values.

Contract	Source	Value	Actions
pf.envdata	Expression	`#result = #this.get("context.HttpRequest").getObjectValue().getParameter("pf.envdata") != null ? #this.get("context.HttpRequest").getObjectValue().getParameter("pf.envdata").toString() : null`	Edit
policy.action	Adapter		None available
username	Adapter		None available

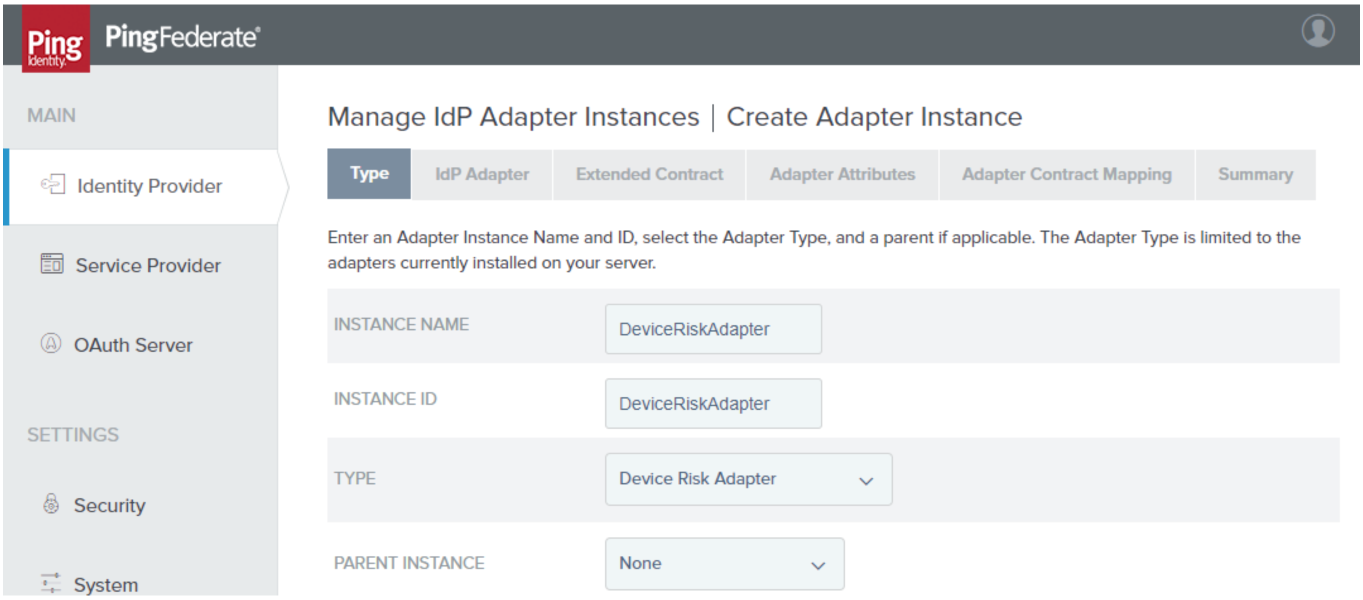
10. Click **Done** to save the new value. On the next page that appears, click **Done** again. Click **Save** on the next screen to finally save these changes.

Next, we create a new IdP that will be used by Ping Federate to action authentication policies

11. Click **Identity Provider > Adapters > Create New Instance** and fill the relevant fields using the following values so they match the instructions in the rest of this document:

    Instance Name: DeviceRiskAdapter
    Instance ID: DeviceRiskAdapter
    Type: Device Risk Adapter

    Once done click **Next**.



Manage IdP Adapter Instances | Create Adapter Instance

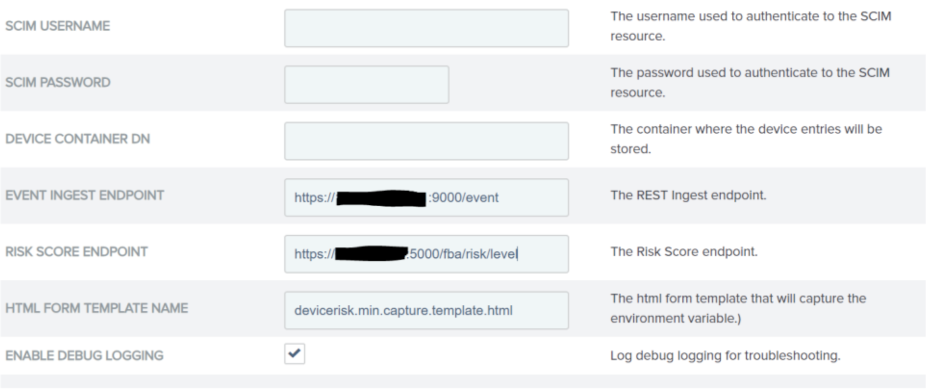
Type | IdP Adapter | Extended Contract | Adapter Attributes | Adapter Contract Mapping | Summary

Enter an Adapter Instance Name and ID, select the Adapter Type, and a parent if applicable. The Adapter Type is limited to the adapters currently installed on your server.

INSTANCE NAME	DeviceRiskAdapter
INSTANCE ID	DeviceRiskAdapter
TYPE	Device Risk Adapter
PARENT INSTANCE	None

12. In the page that follows fill the relevant fields with the following values, making sure to change the path and ports to match the ones used in your current setup:

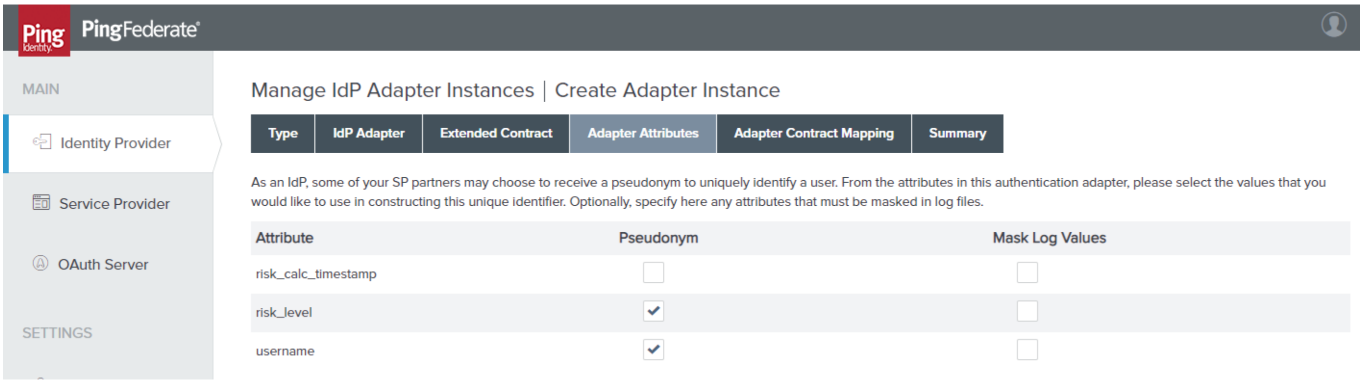
    Event ingest endpoint: https://<Forcepoint Behavioral Analytics Streaming Ingest Public API hostname or IP>:9000/event
    Risk score endpoint: https://<Risk Exporter hostname or IP>:5000/fba/risk/level
    Note: The certs for the two URLs above need to be trusted by ping host machine.



SCIM USERNAME		The username used to authenticate to the SCIM resource.
SCIM PASSWORD		The password used to authenticate to the SCIM resource.
DEVICE CONTAINER DN		The container where the device entries will be stored.
EVENT INGEST ENDPOINT	https://[redacted]:9000/event	The REST ingest endpoint.
RISK SCORE ENDPOINT	https://[redacted]5000/fba/risk/level	The Risk Score endpoint.
HTML FORM TEMPLATE NAME	devicerisk.min.capture.template.html	The html form template that will capture the environment variable.)
ENABLE DEBUG LOGGING	☑	Log debug logging for troubleshooting.

Once done click **Next**.

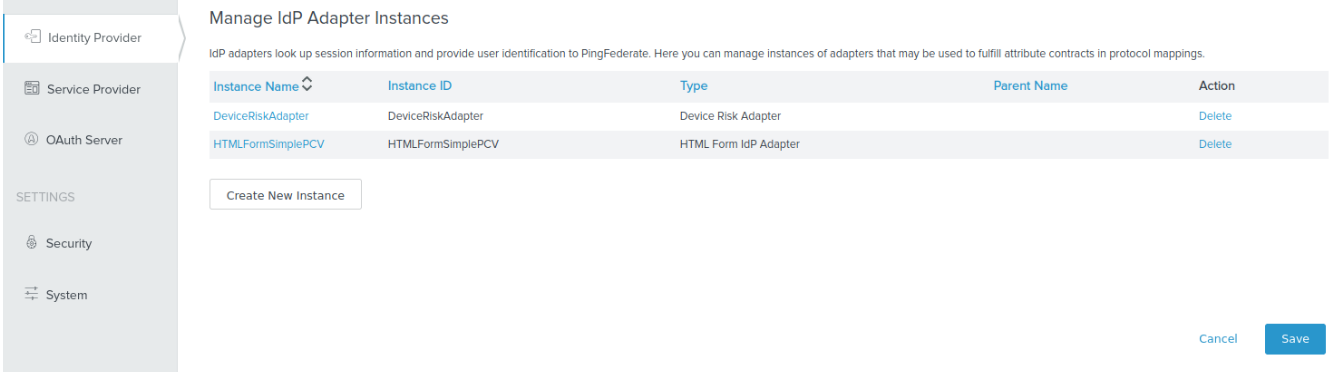
13. Click **Next** again and once on **Adapter Attributes** tab and tick the checkboxes **Pseudonym** for both "**risk_level**" and "**username**". Once done click **Next** in the next two pages that appear and once Summary page is reached, Click **Done** and in the next page click **Save** to save the changes.



Manage IdP Adapter Instances | Create Adapter Instance

As an IdP, some of your SP partners may choose to receive a pseudonym to uniquely identify a user. From the attributes in this authentication adapter, please select the values that you would like to use in constructing this unique identifier. Optionally, specify here any attributes that must be masked in log files.

Attribute	Pseudonym	Mask Log Values
risk_calc_timestamp	□	□
risk_level	☑	□
username	☑	□



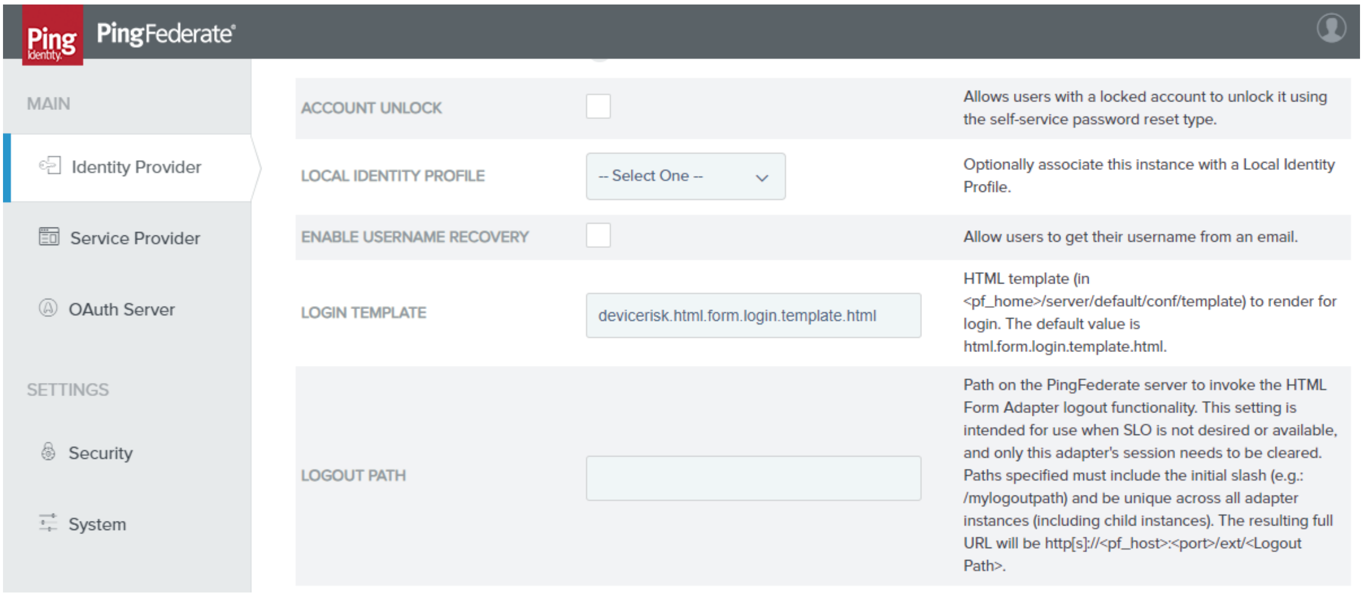
Manage IdP Adapter Instances

IdP adapters look up session information and provide user identification to PingFederate. Here you can manage instances of adapters that may be used to fulfill attribute contracts in protocol mappings.

Instance Name	Instance ID	Type	Parent Name	Action
DeviceRiskAdapter	DeviceRiskAdapter	Device Risk Adapter		Delete
HTMLFormSimplePCV	HTMLFormSimplePCV	HTML Form IdP Adapter		Delete

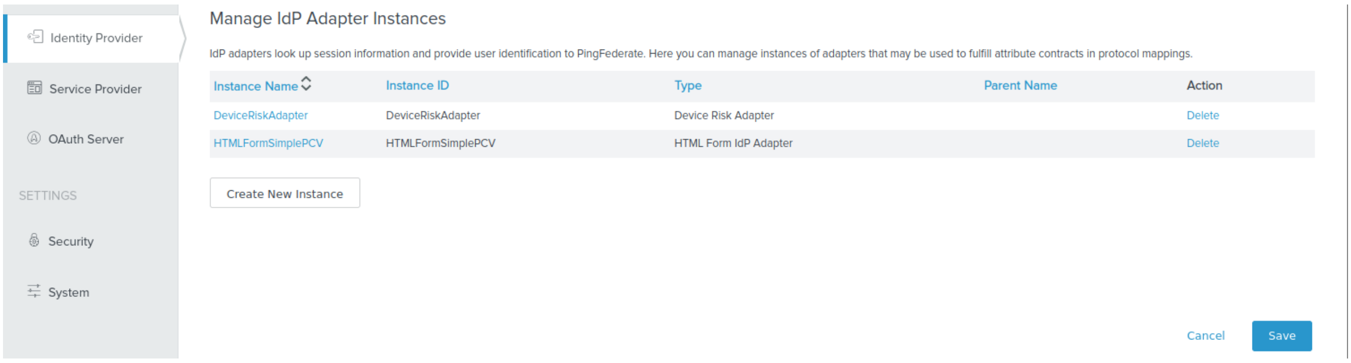
14. In the Identity Provider page go to Integration > Adapters, click HTMLFromSimplePCV then IdP Adapter
15. Scroll to the bottom of the page and click **Show Advanced Fields**
16. Find the **Login Template** field and replace the existing value with

    devicerisk.html.form.login.template.html



ACCOUNT UNLOCK	□	Allows users with a locked account to unlock it using the self-service password reset type.
LOCAL IDENTITY PROFILE	-- Select One --	Optionally associate this instance with a Local Identity Profile.
ENABLE USERNAME RECOVERY	□	Allow users to get their username from an email.
LOGIN TEMPLATE	devicerisk.html.form.login.template.html	HTML template (in <pf_home>/server/default/conf/template) to render for login. The default value is html.form.login.template.html.
LOGOUT PATH		Path on the PingFederate server to invoke the HTML Form Adapter logout functionality. This setting is intended for use when SLO is not desired or available, and only this adapter's session needs to be cleared. Paths specified must include the initial slash (e.g.: /mylogoutpath) and be unique across all adapter instances (including child instances). The resulting full URL will be http[s]://<pf_host>:<port>/ext/<Logout Path>.

Once done click **Done.** In the next page, click **Save**



Manage IdP Adapter Instances

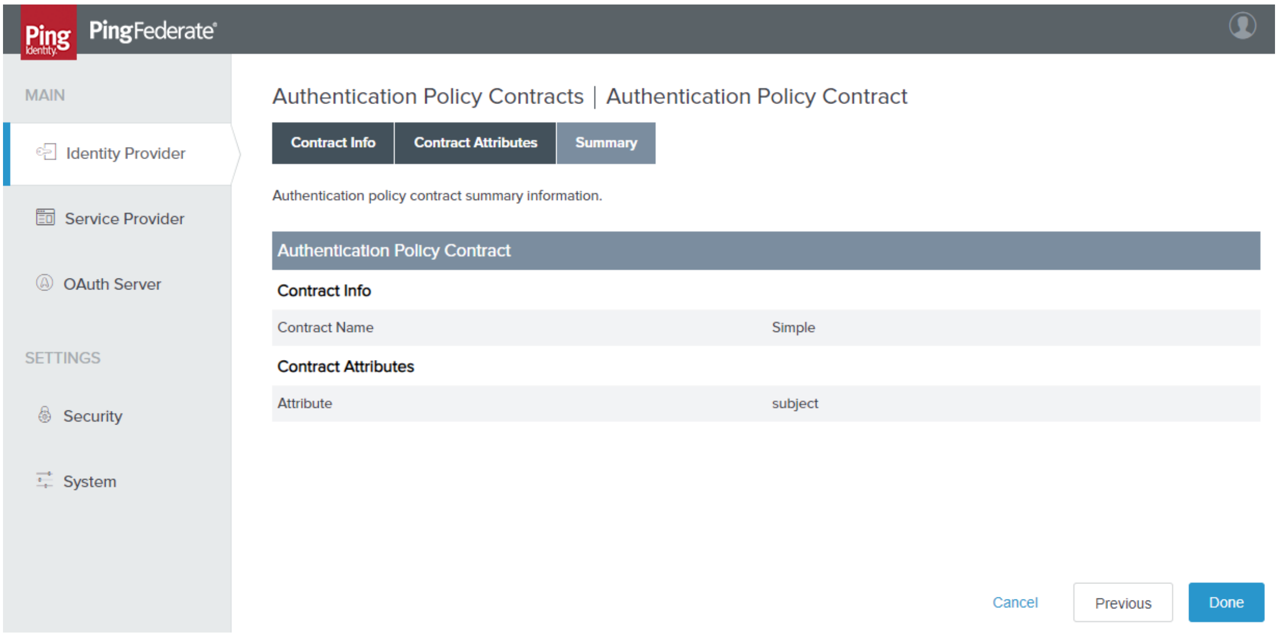
IdP adapters look up session information and provide user identification to PingFederate. Here you can manage instances of adapters that may be used to fulfill attribute contracts in protocol mappings.

Instance Name	Instance ID	Type	Parent Name	Action
DeviceRiskAdapter	DeviceRiskAdapter	Device Risk Adapter		Delete
HTMLFormSimplePCV	HTMLFormSimplePCV	HTML Form IdP Adapter		Delete

The next step is to define **Policy Contracts**: this will be used by Ping Federate to oppose different

authentication steps to each user based on his risk level provided by Forcepoint Behavioral Analytics.

17. Click Identity Provider > Authentication Policies > Policy Contracts > Create New Contract
18. Enter a name in the **Contact Name** field (e.g. “Simple” in the rest of this document), then **Next > Next > Done > Save**



Authentication Policy Contracts | Authentication Policy Contract

Authentication policy contract summary information.

Authentication Policy Contract

Contract Info

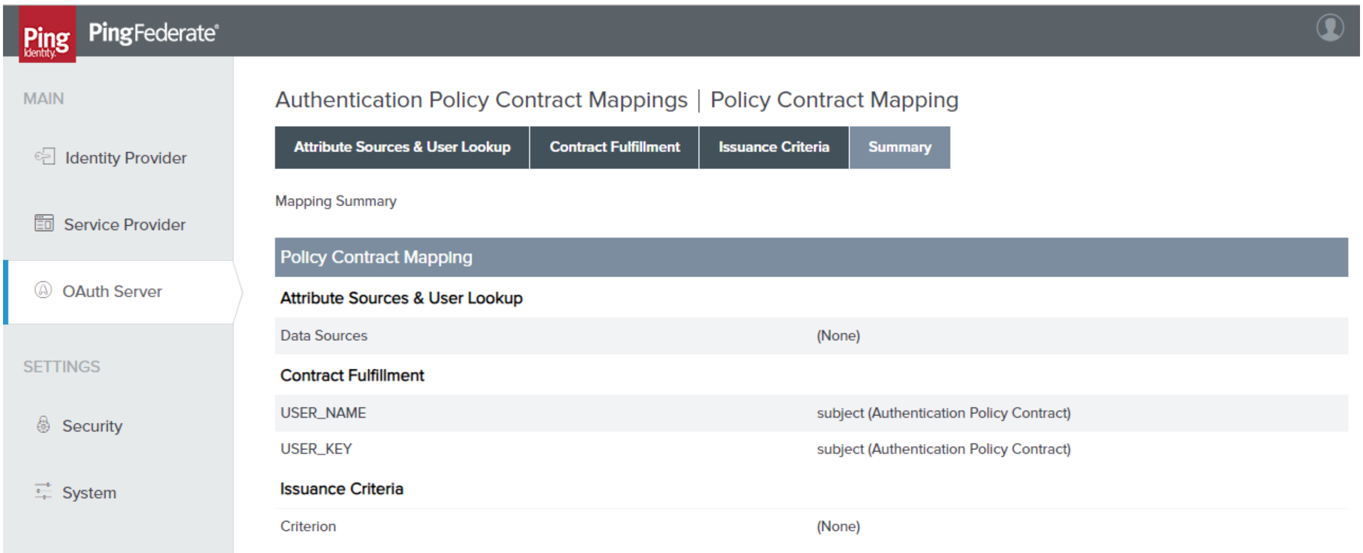
Contract Name	Simple

Contract Attributes

Attribute	subject

19. Click **OAuth Server > Grant Mapping > Authentication Policy Contract Mapping** and select from the drop-down menu the contract created in the previous step, then click **Add Mapping > Next** which takes you to the **Contract Fulfillment** tab
20. For both **USER_KEY** and **USER_NAME** contracts select **Authentication Policy Contract** from the drop-down menus; select **subject** as value

Once done click **Next**, do nothing in **Issuance Criteria** in the next screen but click **Next** then **Save**



Authentication Policy Contract Mappings | Policy Contract Mapping

Mapping Summary

Policy Contract Mapping

Attribute Sources & User Lookup

Data Sources	(None)

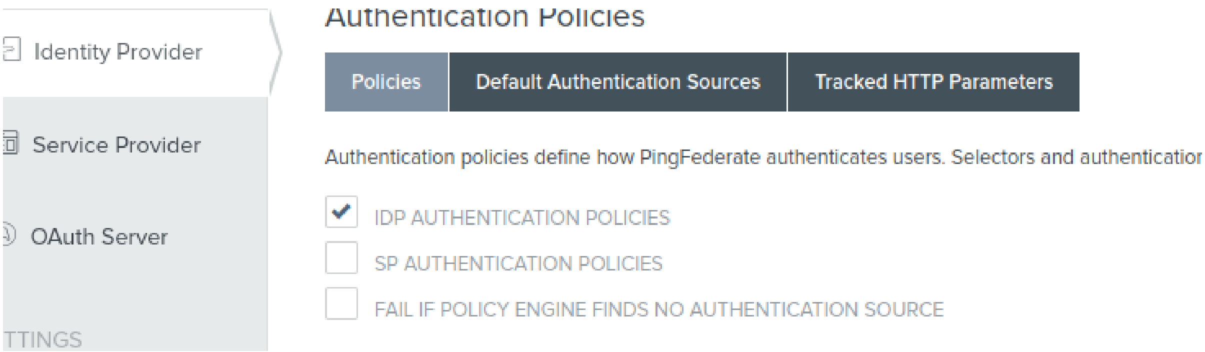
Contract Fulfillment

USER_NAME	subject (Authentication Policy Contract)
USER_KEY	subject (Authentication Policy Contract)

Issuance Criteria

Criterion	(None)

Click Identity Provider > Authentication Policies > Policies, and make sure that IDP AUTHENTICATION POLICIES checkbox is selected.



Authentication policies define how PingFederate authenticates users. Selectors and authenticatior

☑ IDP AUTHENTICATION POLICIES

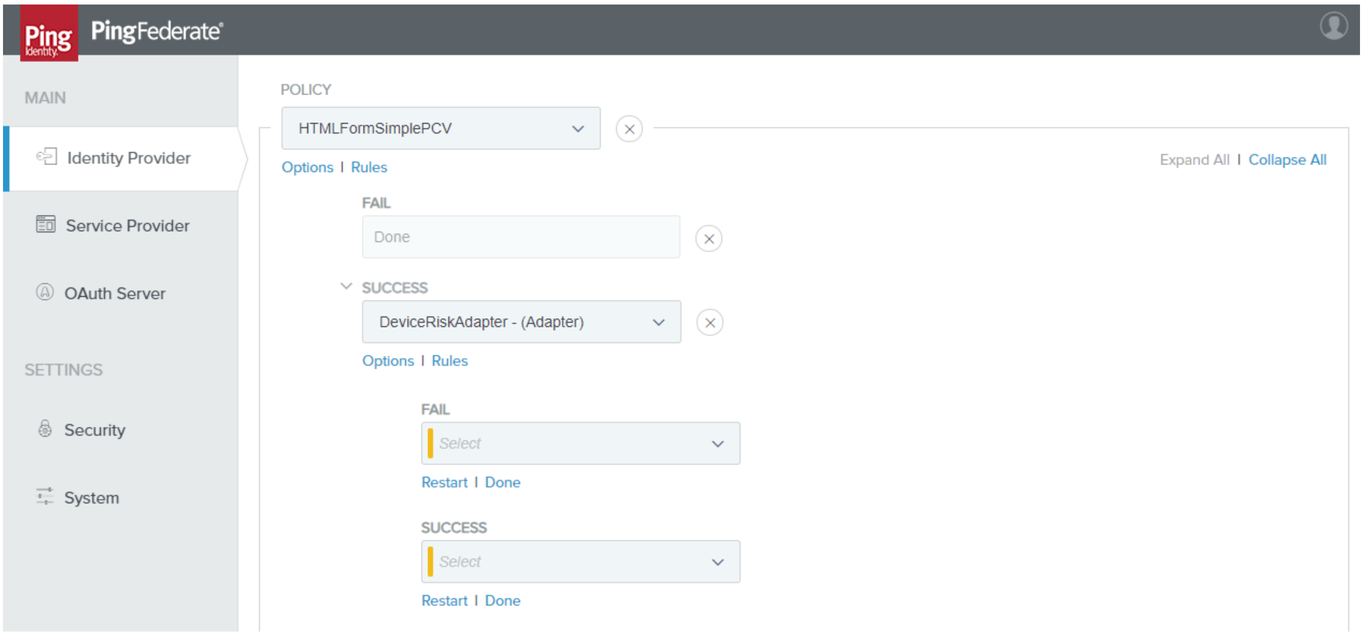
☐ SP AUTHENTICATION POLICIES

☐ FAIL IF POLICY ENGINE FINDS NO AUTHENTICATION SOURCE

21. Click Identity Provider > Authentication Policies > Policies > Add Policy, use the following values

    Name: Device Risk Policy
    Policy: pick HTMLFormSimplePCV from IdP Adapters in the drop-down menu

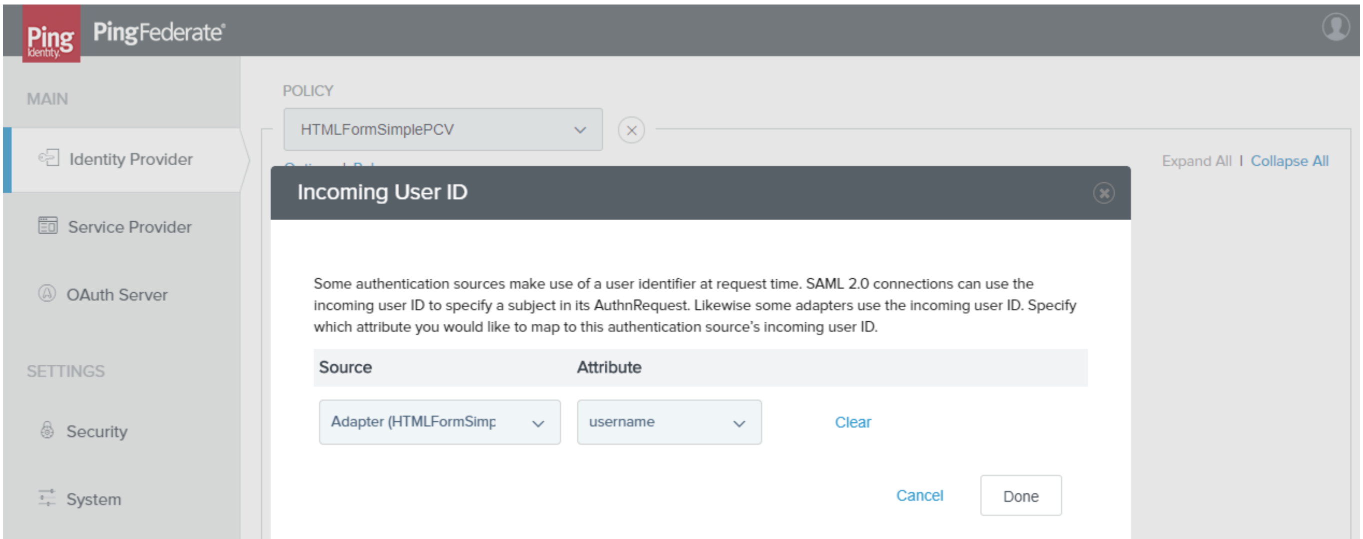
For the **Fail** case click **Done**, while for the **Success** case click on the drop-down menu and select DeviceRiskAdapter from the IdP Adapters menu



22. Under the top **Success** drop-down menu click **Options** and pick the following entries for each drop-down menu

    Source: Adapter (HTMLFormSimplePCV)

    Attribute: username

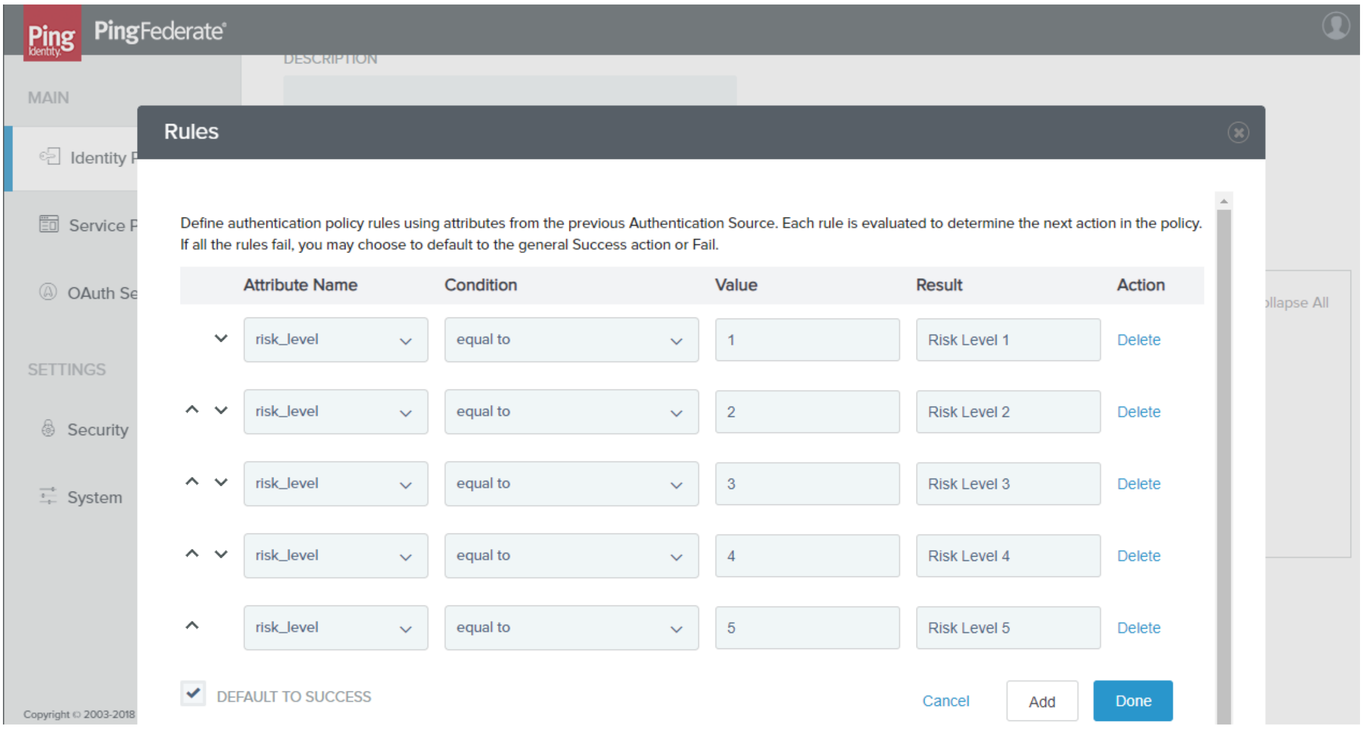


Some authentication sources make use of a user identifier at request time. SAML 2.0 connections can use the incoming user ID to specify a subject in its AuthnRequest. Likewise some adapters use the incoming user ID. Specify which attribute you would like to map to this authentication source's incoming user ID.

Source	Attribute	
Adapter (HTMLFormSimp	username	Clear

Click **Done** once finished

23. Under the top **Success** drop-down now click **Rules** and add one line for each possible value returned by Forcepoint Behavioral Analytics as in the picture below, each of these lines will be assigned with a different action so that Ping will oppose different challenges based on the **risk_level** value. User can add a new entry (rule) by clicking **Add** button.



Define authentication policy rules using attributes from the previous Authentication Source. Each rule is evaluated to determine the next action in the policy. If all the rules fail, you may choose to default to the general Success action or Fail.

Attribute Name	Condition	Value	Result	Action
risk_level	equal to	1	Risk Level 1	Delete
risk_level	equal to	2	Risk Level 2	Delete
risk_level	equal to	3	Risk Level 3	Delete
risk_level	equal to	4	Risk Level 4	Delete
risk_level	equal to	5	Risk Level 5	Delete

☑ DEFAULT TO SUCCESS

The **Risk Exporter** returns risk_level -1 for entities that have no risk level. This can be configured as a policy rule to utilize a specific risk level, or by ticking the **Default to success** box will let users authenticate normally if their risk level has not been calculated yet.

Please note that Ping Federate does not provide inequality operators for the risk_level value, therefore the system administrator must create a policy rule for each possible case (e.g. it is not possible to configure a rule for the case “risk_level > 2”)

Once finished, click **Done**

24. Typically, a system administrator may configure:

- Standard authentication steps for low risk users (e.g. risk_level = 1 or 2)
- Multi-factor authentication for medium risk users (e.g. risk_level = 3 or 4)
- More complex, or deny authentication for the most risky users (e.g. risk_level = 5)
- A custom action or one of the above options for users whose risk level has not been calculated yet (risk_level = -1)

The choice of how to map risk_level values to authentication steps is left to the system administrator, since there might already be in place custom authentication policies, and more importantly because the mapping decision differs from customer to customer based on their security policies.

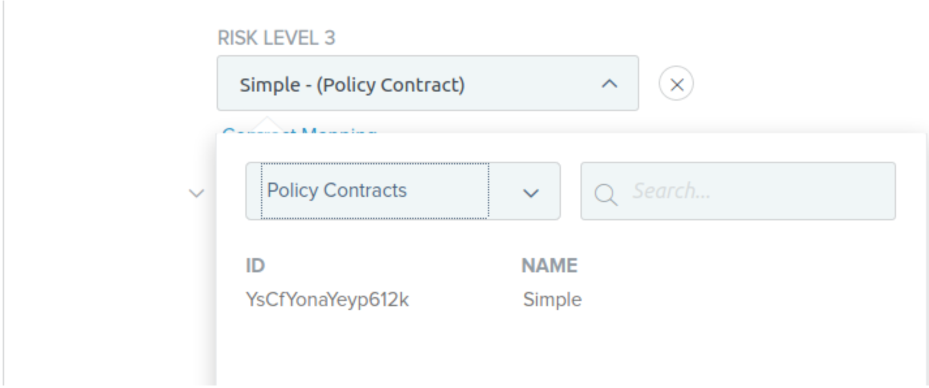
For system administrators with no previous authentication policies in place, here is an example of how to configure Ping Federate based on the risk level provided by Forcepoint Behavioral Analytics.

The configuration described in the next pages

- Allows standard access with username/password for users whose risk level is 1 to 4, and for users whose risk level has not been calculated yet
- Denies access to users whose risk_level equals to 5

The process to configure this is as follows:

1. Click **Expand All** to see all rules as a tree.
2. For the **Fail** case under **DeviceRiskAdapter** click **Done.**
3. Under **Risk Level 1**, click on the drop-down menu and on the next drop-down menu select **Policy Contracts,** select **Simple** (created earlier in step 18 of Step 2 – Setup Ping Federate)



RISK LEVEL 3

Simple - (Policy Contract)

Policy Contracts

ID	NAME
YsCfYonaYeyp612k	Simple

4. Click Contract Mapping
5. Click **Next** in the **Attribute Source & User Lookup** page and in the **Contract Fulfillment** page use the following choices in the drop-down menus then click **Next**

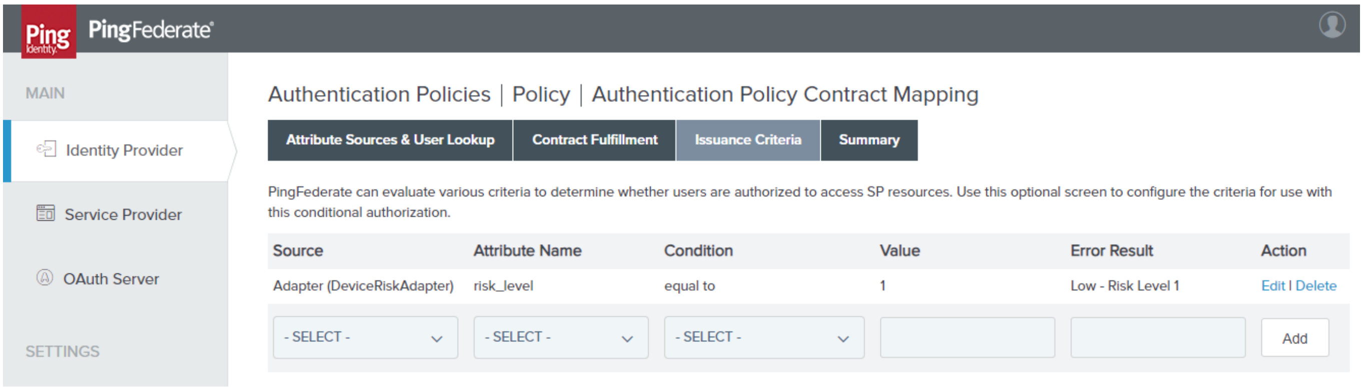
   Source: Adapter (HTMLFormSimplePCV)
   Value: username

6. In **Issuance Criteria** page, pick the following values in each drop-down menu:

   Source: Adapter (DeviceRiskLevel)
   Attribute Name: risk_level
   Condition: equal to
   Value: 1

Error Result: Low – Risk Level 1

Once done click **Add** then **Next**. Click **Done**.



Authentication Policies | Policy | Authentication Policy Contract Mapping

Attribute Sources & User Lookup | Contract Fulfillment | Issuance Criteria | Summary

PingFederate can evaluate various criteria to determine whether users are authorized to access SP resources. Use this optional screen to configure the criteria for use with this conditional authorization.

Source	Attribute Name	Condition	Value	Error Result	Action
Adapter (DeviceRiskAdapter)	risk_level	equal to	1	Low - Risk Level 1	Edit | Delete
- SELECT -	- SELECT -	- SELECT -			Add

Repeat steps 2 to 4 for each Risk Level 1 to 4.

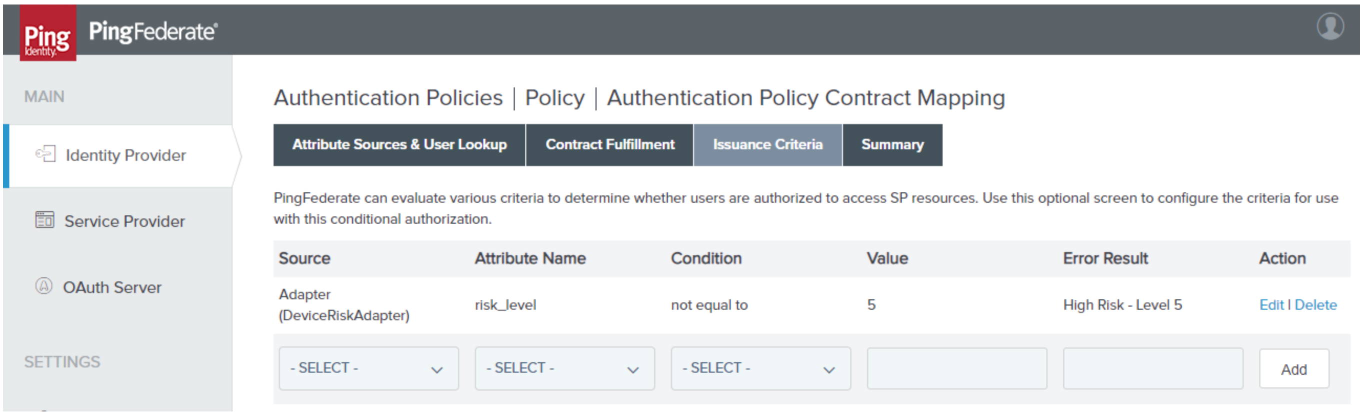
7. For Risk Level 5 click Contract Mapping then Next in the Attribute Source & User Lookup page since no changes are to be made in this page. In the Contract Fulfillment page use the following choices in the drop-down menus then click Next

   Source: Adapter (HTMLFormSimplePCV)
   Value: username

8. In the **Issuance Criteria** page, select the values in the drop-down menus as follows

   Source: Adapter (DeviceRiskLevel)
   Attribute Name: risk_level
   Condition: not equal to
   Value: 5
   Error Result: High – Risk Level 5

Once done click **Add** then **Next**. Click **Done**.



Authentication Policies | Policy | Authentication Policy Contract Mapping

PingFederate can evaluate various criteria to determine whether users are authorized to access SP resources. Use this optional screen to configure the criteria for use with this conditional authorization.

Source	Attribute Name	Condition	Value	Error Result	Action
Adapter (DeviceRiskAdapter)	risk_level	not equal to	5	High Risk - Level 5	Edit I Delete
- SELECT -	- SELECT -	- SELECT -			Add

By doing this, when Ping Federate processes the policy for a user with risk_level = 5, will route the user to the "**RISK LEVEL 5**" rule. This rule would authorize the login only if the **Issuance Criteria** is met, but since we configured **not equal to** as **Condition**, this will never be met thus no authentication for the user whose risk level is 5.

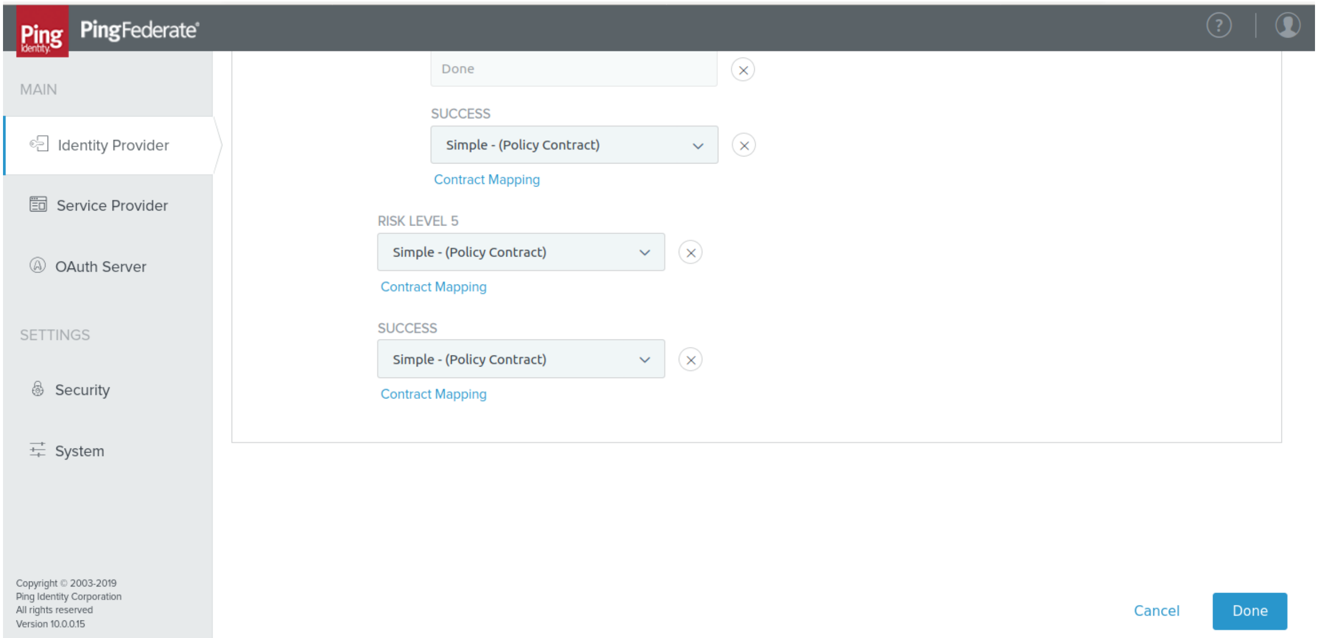
For **Success** under **DeviceRiskAdapter**, click on the drop-down menu and on the next drop-down menu select **Policy Contracts**, select **Simple** (created earlier in step 18 of Step 2 – Setup Ping Federate).

1. Click **Contract Mapping** then **Next** in the **Attribute Source & User Lookup** page since no changes are to be made in this page. In the **Contract Fulfillment** page use the following choices in the drop-down menus then click **Next**

   Source: Adapter (HTMLFormSimplePCV)
   Value: username

Once done click **Next** then **Next**. Click **Done**.

Once all the risk levels are mapped correctly, Click **Done** at the bottom of the page.



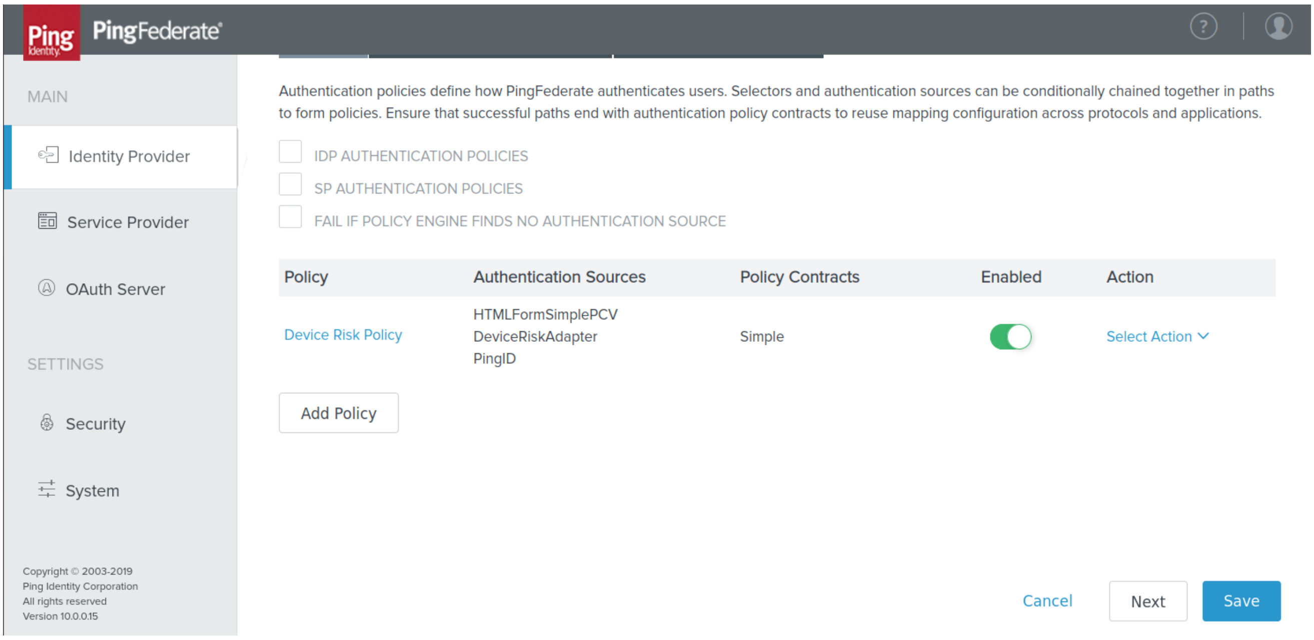
Done

SUCCESS: Simple - (Policy Contract)
Contract Mapping

RISK LEVEL 5: Simple - (Policy Contract)
Contract Mapping

SUCCESS: Simple - (Policy Contract)
Contract Mapping

In the next screen, Click **Save** to save the configuration for the Device Risk Policy



Authentication policies define how PingFederate authenticates users. Selectors and authentication sources can be conditionally chained together in paths to form policies. Ensure that successful paths end with authentication policy contracts to reuse mapping configuration across protocols and applications.

□ IDP AUTHENTICATION POLICIES
□ SP AUTHENTICATION POLICIES
□ FAIL IF POLICY ENGINE FINDS NO AUTHENTICATION SOURCE

Policy	Authentication Sources	Policy Contracts	Enabled	Action
Device Risk Policy	HTMLFormSimplePCV DeviceRiskAdapter PingID	Simple	●	Select Action

Step 3 – Ingesting failed login attempts

Information of failed login attempts are available from the **audit.log** file stored inside the Ping Federate host. In order to provide Forcepoint Behavioral Analytics with data of failed login attempts, log files must be accessible to the machine where the **Failed Logins Importer** component is installed. Typically, this is achieved by

- mapping an NFS share to the PingFederate audit logs directory in read-only mode

- configuring the NFS share so that is available only to the machine where the **Failed Logins Importer** component is installed

Once the audit logs folder is reachable from the machine hosting the **Failed Logins Importer**, proceed with the implementation of the actual component.

Implementation – Traditional

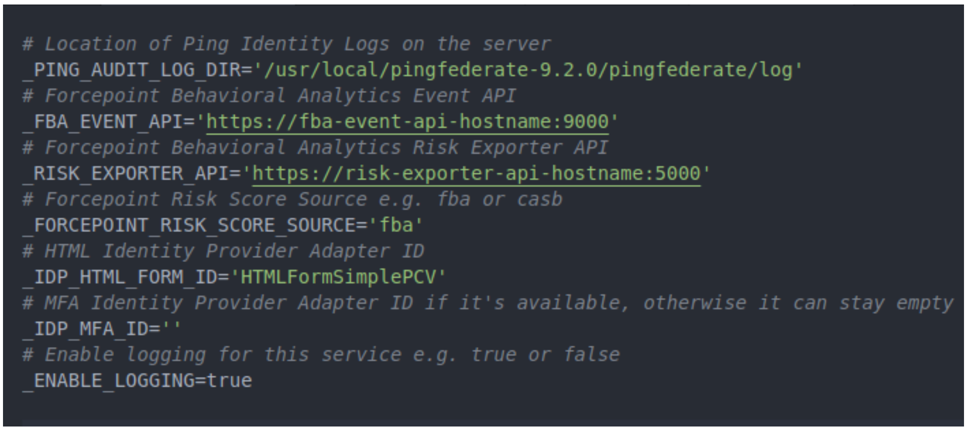
1. Unpack the **fp-fba-failed-logins-importer-ping-v1.tar.gz** file. The examples use the location **/opt/fp-fba-failed-logins-importer-ping-v1/**, however the administrator can change to another location if desired.

   wget --content-disposition https://frcpnt.com/fp-fba-failed-logins-importer-ping-latest
   tar -zxvf fp-fba-failed-logins-importer-ping-v1.tar.gz -C /opt/

2. Install script below will install the system prerequisites, run with a user with administrative privileges.

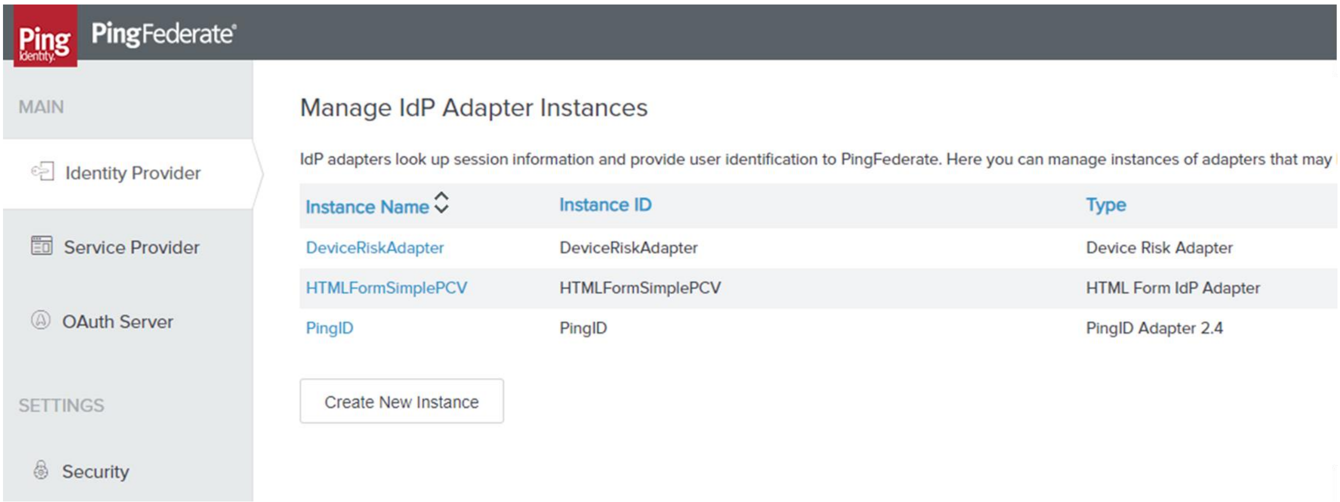
   /opt/fp-fba-failed-logins-importer-ping/deploy/install.sh

3. Move to the **/opt/fp-fba-failed-logins-importer-ping-v1/** folder and edit the file **user-config.sh** and change the values to match the hostnames/IP addresses, ports, paths, filenames, and credentials in your current environment relevant to your setup.



```
# Location of Ping Identity Logs on the server
_PING_AUDIT_LOG_DIR='/usr/local/pingfederate-9.2.0/pingfederate/log'
# Forcepoint Behavioral Analytics Event API
_FBA_EVENT_API='https://fba-event-api-hostname:9000'
# Forcepoint Behavioral Analytics Risk Exporter API
_RISK_EXPORTER_API='https://risk-exporter-api-hostname:5000'
# Forcepoint Risk Score Source e.g. fba or casb
_FORCEPOINT_RISK_SCORE_SOURCE='fba'
# HTML Identity Provider Adapter ID
_IDP_HTML_FORM_ID='HTMLFormSimplePCV'
# MFA Identity Provider Adapter ID if it's available, otherwise it can stay empty
_IDP_MFA_ID=''
# Enable logging for this service e.g. true or false
_ENABLE_LOGGING=true
```

The exact value for the **Adapter ID** entries can be found logging into web console of **Ping Federate** > **Identity Provider** > **Integration** > **Adapters**



Manage IdP Adapter Instances

IdP adapters look up session information and provide user identification to PingFederate. Here you can manage instances of adapters that may

Instance Name	Instance ID	Type
DeviceRiskAdapter	DeviceRiskAdapter	Device Risk Adapter
HTMLFormSimplePCV	HTMLFormSimplePCV	HTML Form IdP Adapter
PingID	PingID	PingID Adapter 2.4

If _ENABLE_LOGGING_ is set to true, logs files are created inside /opt/fp-fba-failed-logins-importer-ping/logs/
Note: The Forcepoint Behavioral Analytics event API and Risk Exporter API SSL certificates need to be trusted.

4. Setup script below will install the program prerequisites and run the program, run with a user with administrative privileges.

   /opt/fp-fba-failed-logins-importer-ping/deploy/setup.sh

Implementation – Docker

Please note: an NFS Server is required on the PingFederate Server.

1. Login into docker repository, you'll be asked to enter your username and password (provided below):

   docker login docker.frcpnt.com

   Username: fp-integrations
   Password: t1knmAkn19s

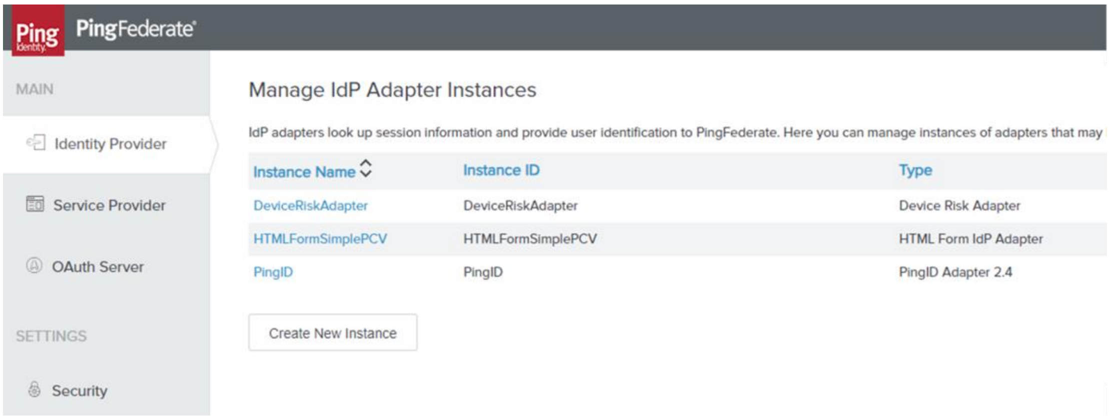
2. Run the below to download the image

   docker pull docker.frcpnt.com/fp-fba-failed-logins-importer-ping

3. Create a new file named **user-config.sh** and insert the following contents:

```
#!/usr/bin/env bash
# Leave this value below as it is
_PING_AUDIT_LOG_DIR='/mnt/ping/logs'
# Forcepoint Behavioral Analytics Event API e.g. https://fba-event-api-hostname:9000
_FBA_EVENT_API=''
# Forcepoint Behavioral Analytics Risk Exporter API e.g. https://risk-exporter-api-hostname:5000
_RISK_EXPORTER_API=''
# Forcepoint Risk Score Source e.g. fba or casb
_FORCEPOINT_RISK_SCORE_SOURCE=''
# HTML Identity Provider Adapter ID e.g. HTMLFormSimplePCV
_IDP_HTML_FORM_ID=''
# MFA Identity Provider Adapter ID if it's available, otherwise it can stay empty
_IDP_MFA_ID=''
# Enable logging for this service e.g. true or false
_ENABLE_LOGGING=true
```

The exact value for the **Adapter ID** entries can be found logging into web console of **Ping Federate** > **Identity Provider** > **Integration** > **Adapters**



Manage IdP Adapter Instances

IdP adapters look up session information and provide user identification to PingFederate. Here you can manage instances of adapters that may

Instance Name	Instance ID	Type
DeviceRiskAdapter	DeviceRiskAdapter	Device Risk Adapter
HTMLFormSimplePCV	HTMLFormSimplePCV	HTML Form IdP Adapter
PingID	PingID	PingID Adapter 2.4

Add the relevant values to each line before saving.

4. Run the container with either one of the following commands, depending on your scenario

- if **user-config.sh** file is located locally then run the following command, replacing the red part with the full path of user-config.sh file and the current hostname or IP address of the machine hosting PingFederate with the audit log directory path.

```
docker run --detach --cap-add=SYS_ADMIN --security-opt apparmor:unconfined \
--name fp-fba-failed-logins-importer-ping \
--volume <user-config.shl-full-path>:/usr/fp-fba-failed-logins-importer-ping/user-config.sh \
--env PING_NFS_MAPPING=<ping-server-host-name:nfs_log_dir> \
docker.frcpnt.com/fp-fba-failed-logins-importer-ping
```

if **user-config.sh** file is accessed by a URL, then run the below, replacing the red part with the URL of the user-config.sh file to download and the current hostname or IP address of the machine hosting PingFederate with the audit log directory path.

```
docker run --detach --cap-add=SYS_ADMIN --security-opt apparmor:unconfined \
--name fp-fba-failed-logins-importer-ping \
--env CONFIG_FILE_URL_LOCATION=<config-file-url> \
--env PING_NFS_MAPPING=<ping-server-host-name:nfs_log_dir> \
docker.frcpnt.com/fp-fba-failed-logins-importer-ping
```

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 Behavioral Analytics and Ping Federate in use are listed as compatible

  Forcepoint Behavioral Analytics 3.1.0
  Ping Federate 9.2 or 10.0

- Verify the integration component correctly operates on a clean CentOS 7.x or Ubuntu 18.04 machine with at least 2GB RAM, 20GB free storage and the system needs to be 64-bit
- The Forcepoint Behavioral Analytics event API and Risk Exporter API SSL certificates need to be trusted by the Ping host machine
- The Forcepoint Behavioral Analytics event API port needs to be accessible through the firewall
- User must have sudo permissions in order to run install.sh and setup.sh scripts
- Check the user can download the necessary files with the below commands:

```
wget --content-disposition https://frcpnt.com/fp-riskexporter-api-latest
wget --content-disposition https://frcpnt.com/ping-connector-latest
wget --content-disposition https://frcpnt.com/fp-fba-failed-logins-importer-ping-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 (which has ping Federate appliance) has network connectivity to the Risk Exporter API and Forcepoint Behavioral Analytics API.

  Execute the following commands on the host machine:

```
curl -I https://<Risk Exporter hostname or IP>:5000
curl -I https://<Forcepoint Behavioral Analytics Streaming Ingest Public API Hostname>:9000
```

  replacing the Risk Exporter hostname and Forcepoint Behavioral Analytics Streaming Ingest

Public API Hostname with the ones in use. Please check the first line of the result of both the commands above is:

HTTP/1.0 200 OK

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 API is configured and running properly: Check there are no errors on the following page:

  https://<ping federate host machine ip>:5000/fba/healthcheck

  replacing the <ping federate host machine ip> with the one in use. Check that the following messages appear on the healthcheck URL:

  kafka available - OK!
  Kafka connection is successful - OK!

- Check the risk exporter logs file: From the home directory of ***/fp-riskexporter-api/logs/*** check the log file **risk-score-api.log**

  Check there are no error messages in this log file.

- Check the fba failed logins importer log file: From the home directory of *fp-fba-failed-logins-importer-ping/logs/* check for the log file **fba-ping-events.log**

  Check there are no error messages in this log file.

Docker Implementation

Validate the prerequisites

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

- Check the versions of Forcepoint Behavioral Analytics and Ping Federate in use are listed as compatible

Forcepoint Behavioral Analytics 3.1.0
Ping Federate 9.2 or 10.0

- Verify the integration component correctly operates on a linux based machine with an existing docker setup and meets the minimum hardware requirements of 2GB RAM, 20GB free storage and the system needs to be 64-bit
- The Forcepoint Behavioral Analytics event API and Risk Exporter API SSL certificates need to be trusted by the ping 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 host machine (which has ping Federate appliance) has network connectivity to the Risk Exporter API and Forcepoint Behavioral Analytics API.
  Execute the following commands on the host machine:
  curl -I https://<Risk Exporter hostname or IP>:5000
  curl -I https://<Forcepoint Behavioral Analytics Streaming Ingest Public API Hostname>:9000

Replacing the Risk Exporter hostname and Forcepoint Behavioral Analytics Streaming Ingest Public API Hostname with the ones in use. Please check the first line of the result of both the commands above is:

HTTP/1.0 200 OK

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 API is configured and running properly: Check there are no errors on the following page:

  https://<ping federate host machine ip>:5000/fba/healthcheck

  Replacing the <ping federate host machine ip> with the one in use. Check that the following messages appear on the healthcheck URL:

```
kafka available - OK!
Kafka connection is successful - OK!
```

- Check the logs for fp-fba-failed-logins-importer-ping container:

  ```
  docker logs fp-fba-failed-logins-importer-ping | tail
  ```

  Check the output is similar to the one below and has no errors:

  ```
  crond[41]: line /usr/fp-fba-failed-logins-importer-ping/source/run-fba-ping-mfa-batch-events.sh
  crond[41]: wakeup dt=60
  crond[41]: file root:
  crond[41]: line run-parts /etc/periodic/15min
  crond[41]: line run-parts /etc/periodic/hourly
  crond[41]: line run-parts /etc/periodic/daily
  crond[41]: line run-parts /etc/periodic/weekly
  crond[41]: line run-parts /etc/periodic/monthly
  crond[41]: line /usr/fp-fba-failed-logins-importer-ping/source/run-fba-ping-batch-events.sh
  crond[41]: line /usr/fp-fba-failed-logins-importer-ping/source/run-fba-ping-mfa-batch-events.sh
  ```

- Check the logs for fp-riskexporter-api container:

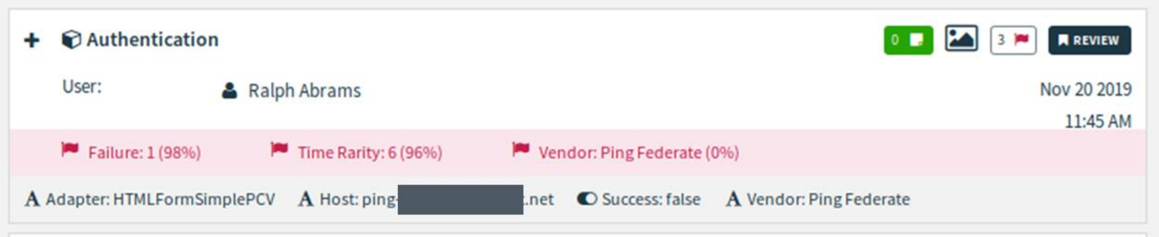
  ```
  docker logs fp-riskexporter-api | tail
  ```

  Check the output is similar to the one below and has no errors:

  ```
  Configs Initialized
  ```

Appendix – Sample events from different types of login failures

- Failed login event from authentication attempt with username and password



+ Authentication 0 3 REVIEW

User: Ralph Abrams

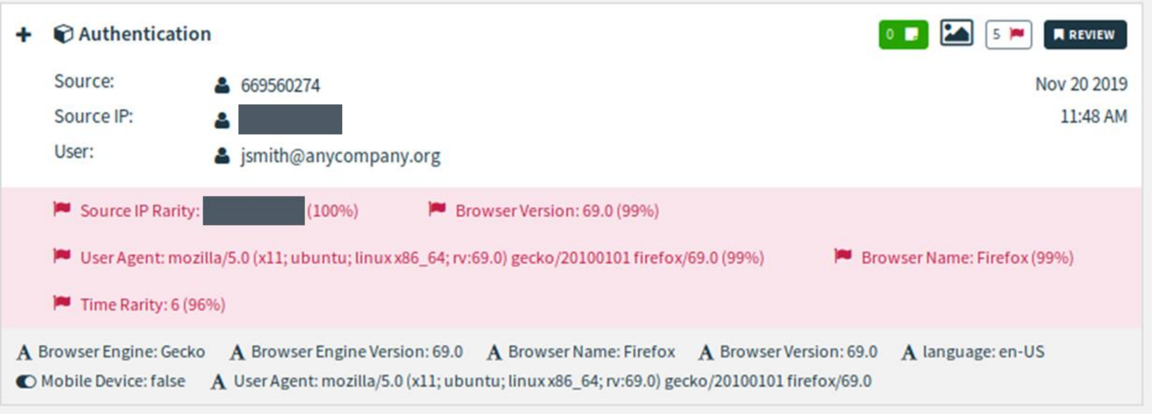
Nov 20 2019 11:45 AM

Failure: 1 (98%) Time Rarity: 6 (96%) Vendor: Ping Federate (0%)

Adapter: HTMLFormSimplePCV Host: ping-[redacted].net Success: false Vendor: Ping Federate

- Failed to verify during MFA stage after username and password authentication is successful, two events are sent:

  First, an event from a success login with username and password (data provided by the **Identity Adapter**):



+ Authentication 0 5 REVIEW

Source: 669560274

Source IP: [redacted]

User: jsmith@anycompany.org

Nov 20 2019 11:48 AM

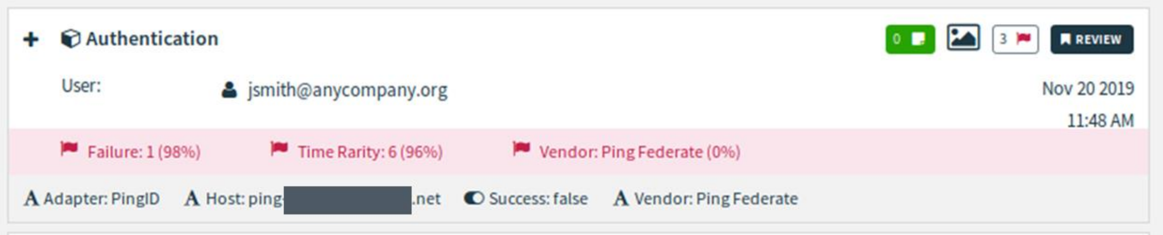
Source IP Rarity: [redacted] (100%) Browser Version: 69.0 (99%)

User Agent: mozilla/5.0 (x11; ubuntu; linux x86_64; rv:69.0) gecko/20100101 firefox/69.0 (99%) Browser Name: Firefox (99%)

Time Rarity: 6 (96%)

Browser Engine: Gecko Browser Engine Version: 69.0 Browser Name: Firefox Browser Version: 69.0 language: en-US Mobile Device: false User Agent: mozilla/5.0 (x11; ubuntu; linux x86_64; rv:69.0) gecko/20100101 firefox/69.0

Then, an event from the failed attempt at MFA stage (from the **audit.log** file):



+ Authentication 0 3 REVIEW

User: jsmith@anycompany.org

Nov 20 2019 11:48 AM

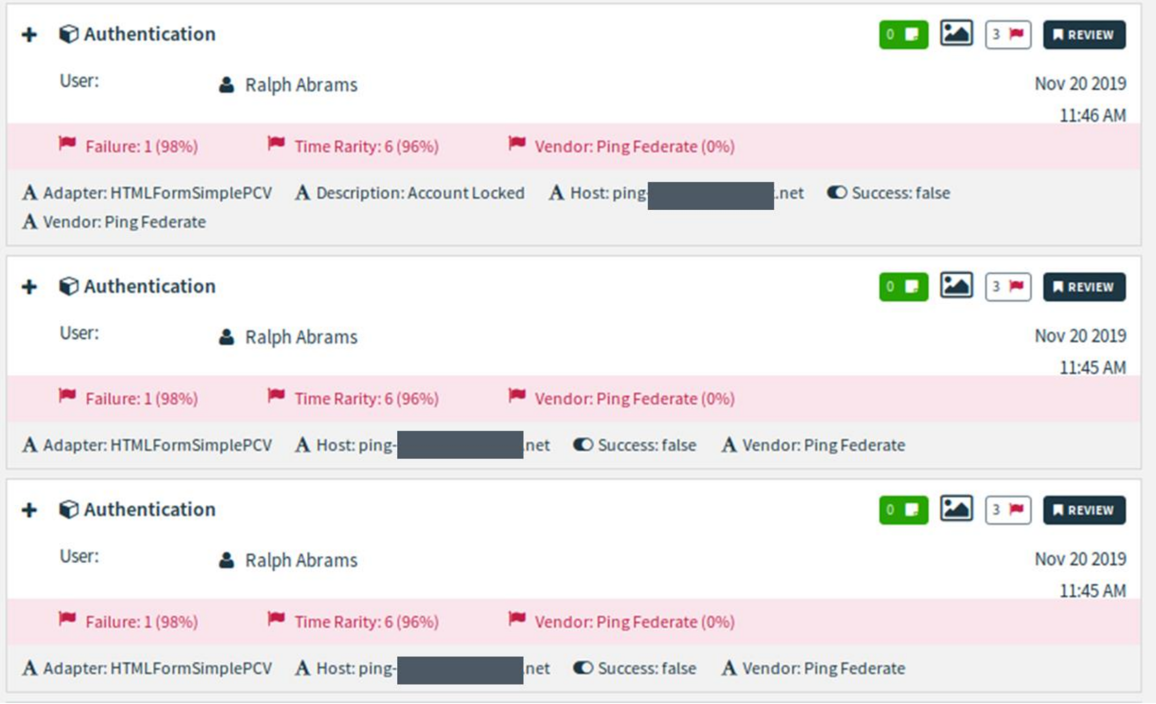
Failure: 1 (98%) Time Rarity: 6 (96%) Vendor: Ping Federate (0%)

Adapter: PingID Host: ping[redacted].net Success: false Vendor: Ping Federate

In the case of MFA, failure is derived from two scenarios: the wrong response to the

Authentication challenge, or the lack of response (e.g. the case of a malicious user who desists
from logging in once opposed with MFA).

- Failed authentication due to lockout after multiple failed attempts (please note the events are listed from the most recent to the last recent one)



Authentication

User: Ralph Abrams

Nov 20 2019
11:46 AM

Failure: 1 (98%) Time Rarity: 6 (96%) Vendor: Ping Federate (0%)

Adapter: HTMLFormSimplePCV Description: Account Locked Host: ping[redacted].net Success: false
Vendor: Ping Federate

Authentication

User: Ralph Abrams

Nov 20 2019
11:45 AM

Failure: 1 (98%) Time Rarity: 6 (96%) Vendor: Ping Federate (0%)

Adapter: HTMLFormSimplePCV Host: ping-[redacted].net Success: false Vendor: Ping Federate

Authentication

User: Ralph Abrams

Nov 20 2019
11:45 AM

Failure: 1 (98%) Time Rarity: 6 (96%) Vendor: Ping Federate (0%)

Adapter: HTMLFormSimplePCV Host: ping-[redacted].net Success: false Vendor: Ping Federate