Forcepoint CASB and Ping Federate

Integration Guide

Forcepoint

Integration Guide

Rabih Abou
14 September 2020
Public

Table of Contents

Version	Date	Author	Notes
0.1	07 April 2020	Rabih Abou Fakher	First draft
0.2	13 April 2020	Neelima Rai	Review
0.3	15 April 2020	Mattia Maggioli	Review
0.4	19 May 2020	Jonathan Knepher	Review
0.5	14 September 2020	Mattia Maggioli	Minor updates

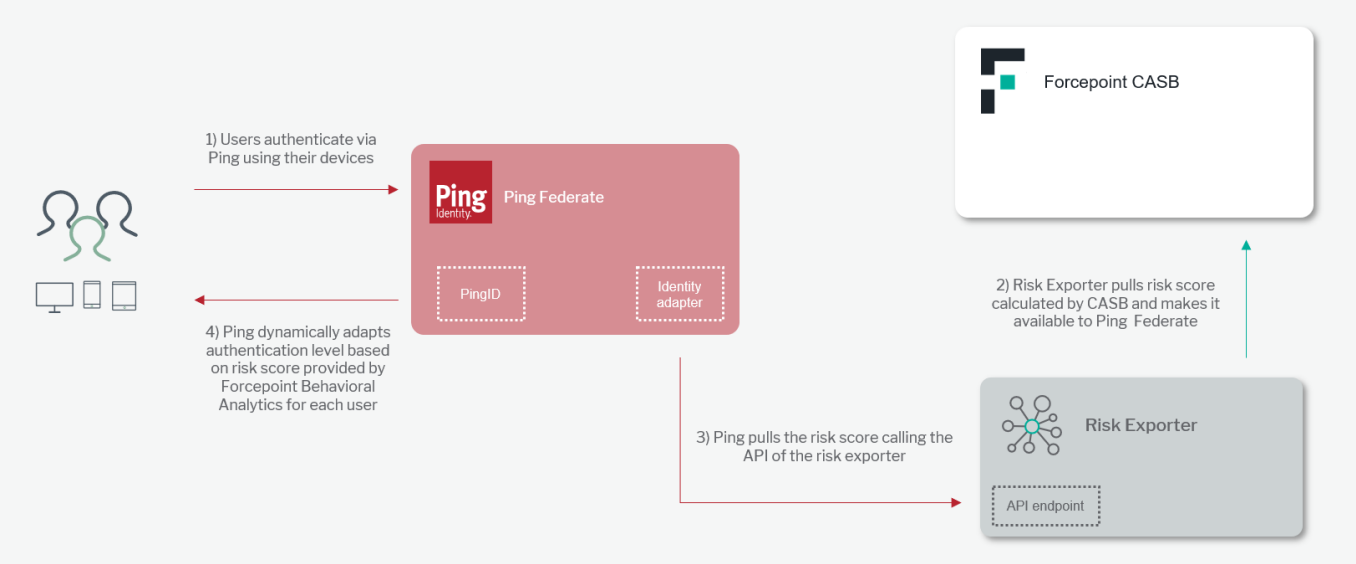
Summary

This guide provides step by step instructions to configure Forcepoint CASB and Ping Federate to enable dynamic authentication policies based on user risk.

The code and instructions provided enable administrators to automatically:

- → Provide the risk score calculated by Forcepoint CASB for each user to Ping Federate
- → Adjust authentication policies applied by Ping Federate to users based on their risk level

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



Caveats

These implementation instructions are tested with the following product versions:

- → Forcepoint CASB 2020 R2
- → Ping Federate 9.2 and 10.0

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

- → 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 must only 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 and the system needs to be 64-bit

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
- → **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 from the mapped risk score calculated by Forcepoint CASB. It is provided as a tar file with one associated configuration file.

Risk Exporter is deployed to a Linux machine with network connectivity to Forcepoint CASB and to 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 with CASB.

```
# # # 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: False
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: True
# 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 # # #
```

Note: The API port must be accessible through the firewall

4. Run the setup script with the command in the example below to install the program prerequisites. Run the command 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

   Username: fp-integrations

   Password: t1knmAkn19s

2. Run the command 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 CASP Risk Score
casb_risk_score_fetch_enable: True
# 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:
# Risk Score mapping example:
# risk_level_1: 0-19
# risk_level_2: 20-49
# risk_level_3: 50-79
# risk_level_4: 80-99
# risk_level_5: 100+
```

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 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 certifcates:

```
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 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 CASB Risk Exporter API.

1. Download and 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)

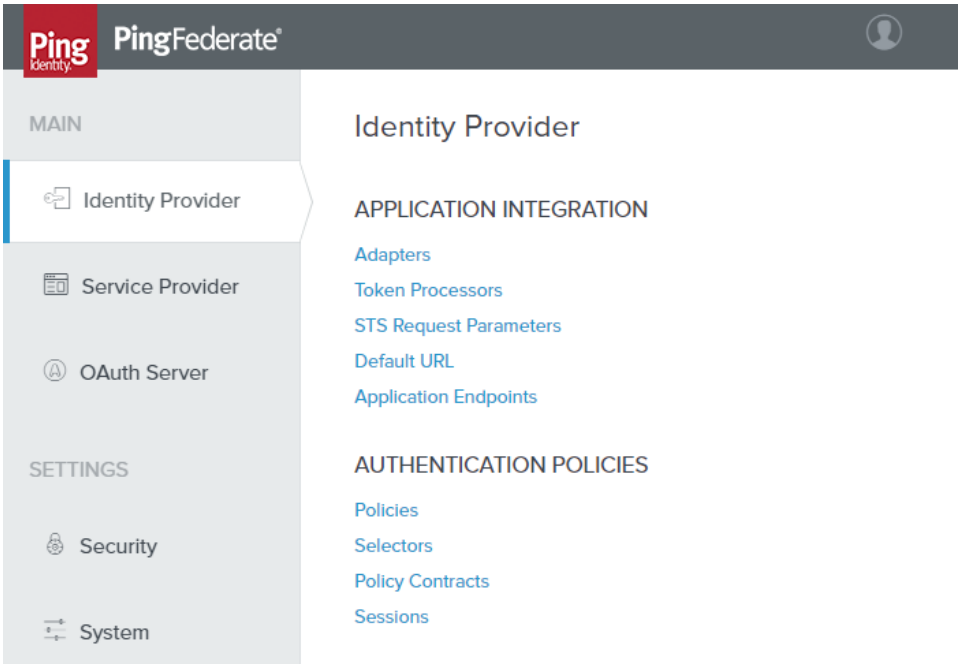
   wget --content-disposition *https://frcpnt.com/ping-connector-latest*
   tar -zxvf ping-connector.tar.gz

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

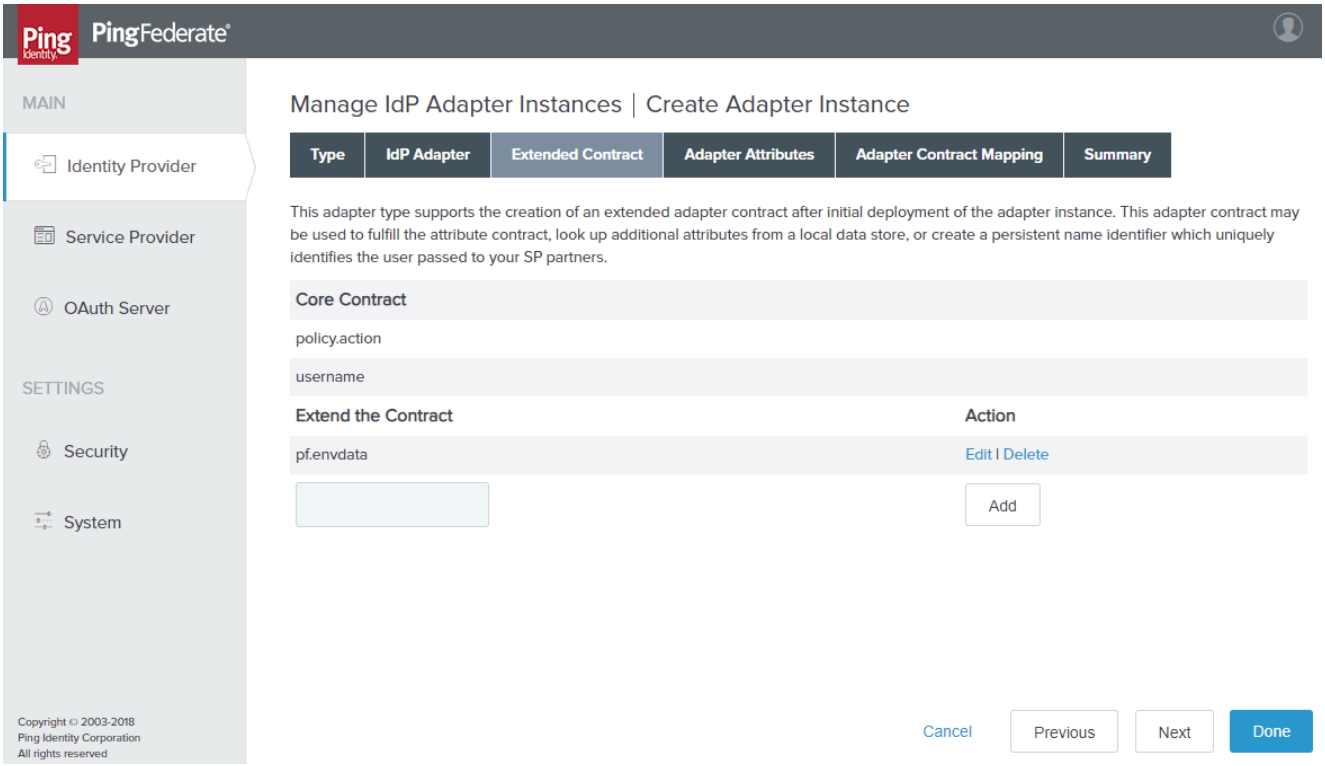
2. Edit the file /<pingfed-home>/pingfederate/server/default/data/config-store/org.sourceid.common.ExpressionManager.xml and change “evaluateExpressions” to true

```
<?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**



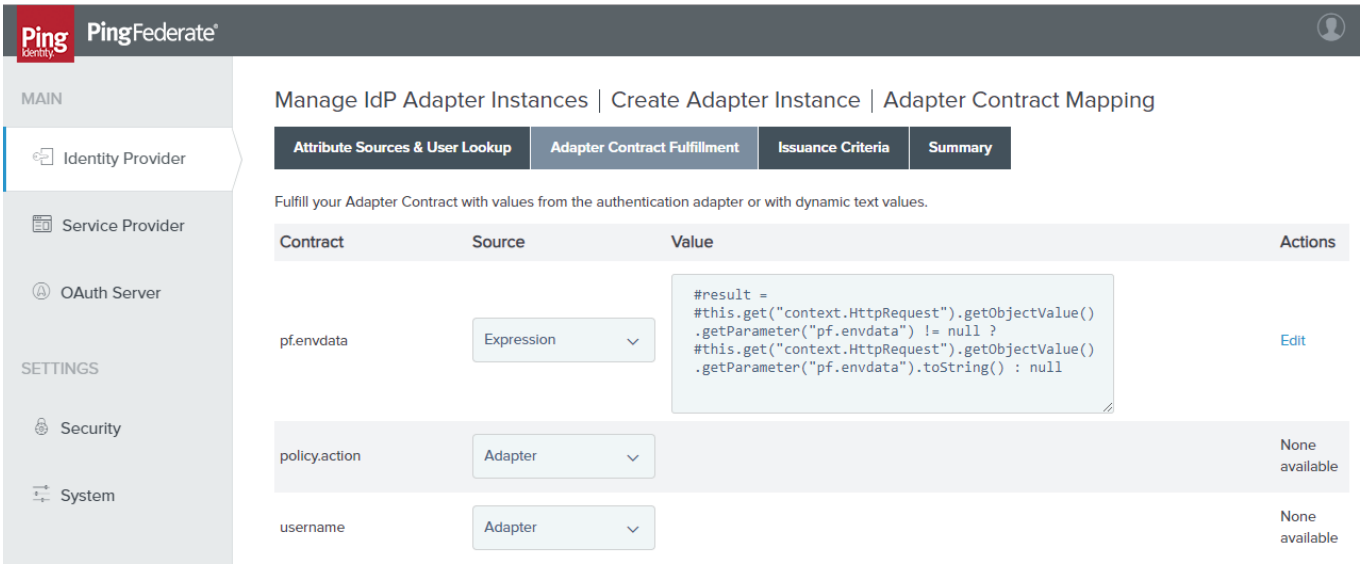
6. Select the existing **HTMLFormSimplePCV** adapter
7. Select **Extended Contract** and add **pf.envdata** in **Extend the Contract** field, then select **Done** to save the configuration of this IdP. Click **Save** on the next page to save these changes.



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



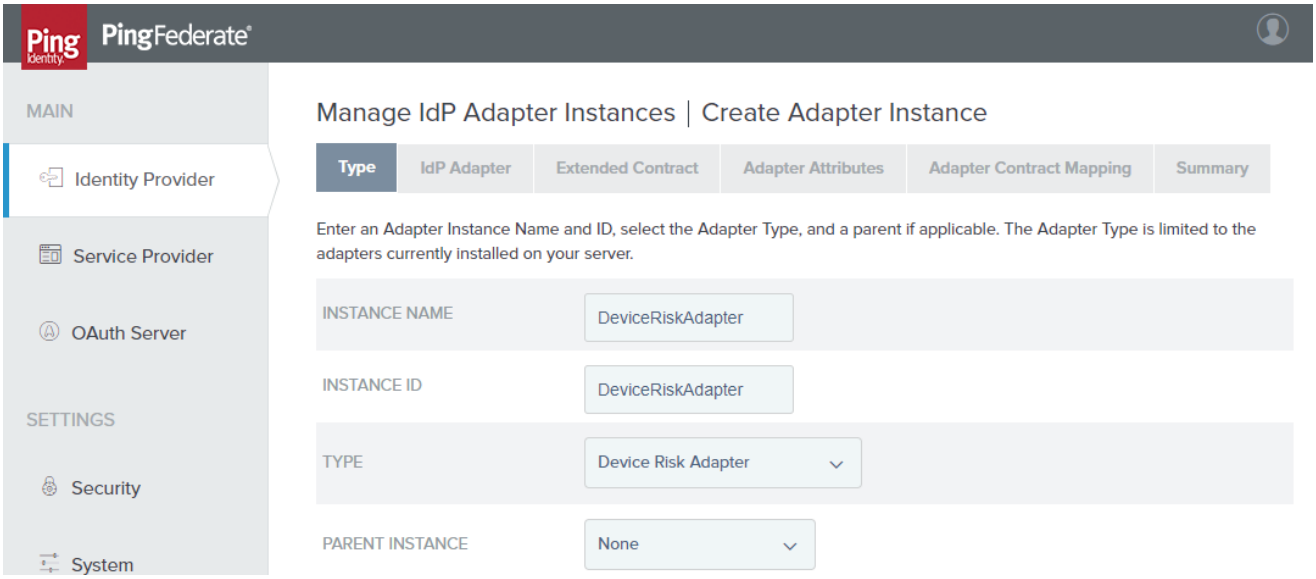
10. Click **Done** to save the new value. On the next page that appears, click **Done** again. Click **Save** on the next screen to finally save these changes.

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

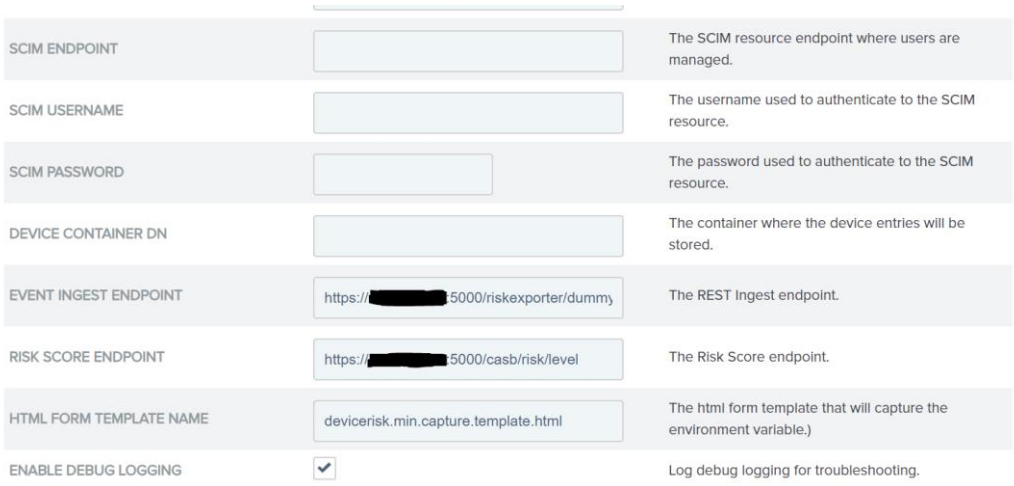
    Once done click **Next**.



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:

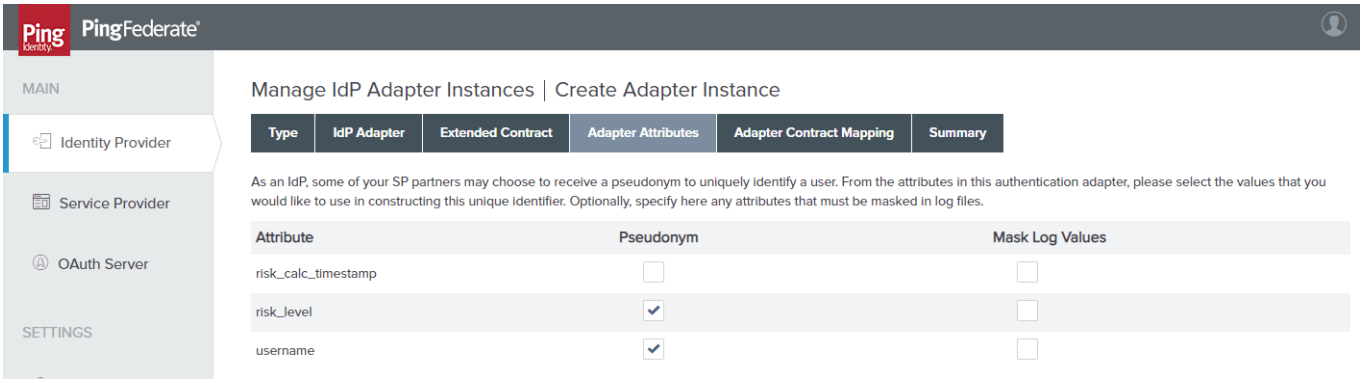
    Event ingest endpoint: https://<Risk Exporter hostname or IP>:5000/riskexporter/dummy/event
    Risk score endpoint: https://<Risk Exporter hostname or IP>:5000/casb/risk/level

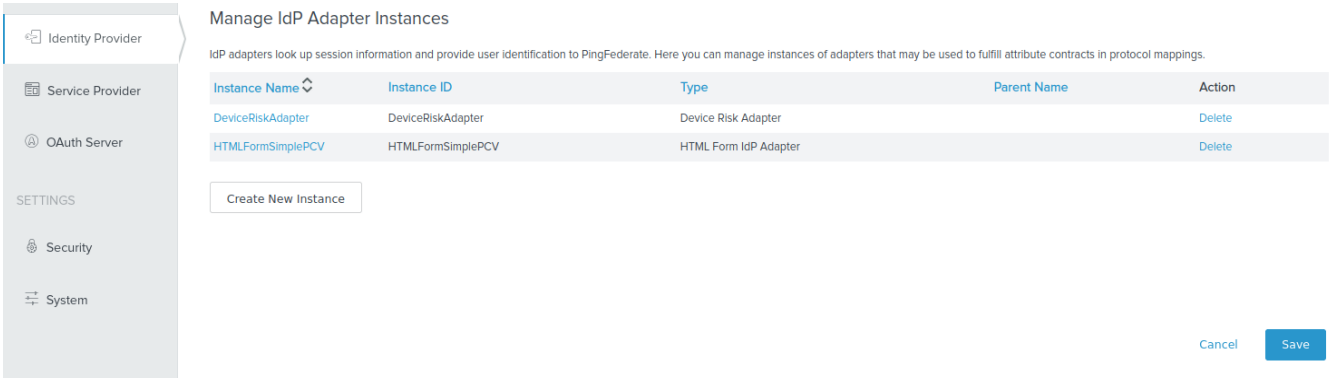
    Note: The cert for the Risk Exporter API above need to be trusted by ping host machine.



    Once done click **Next**.

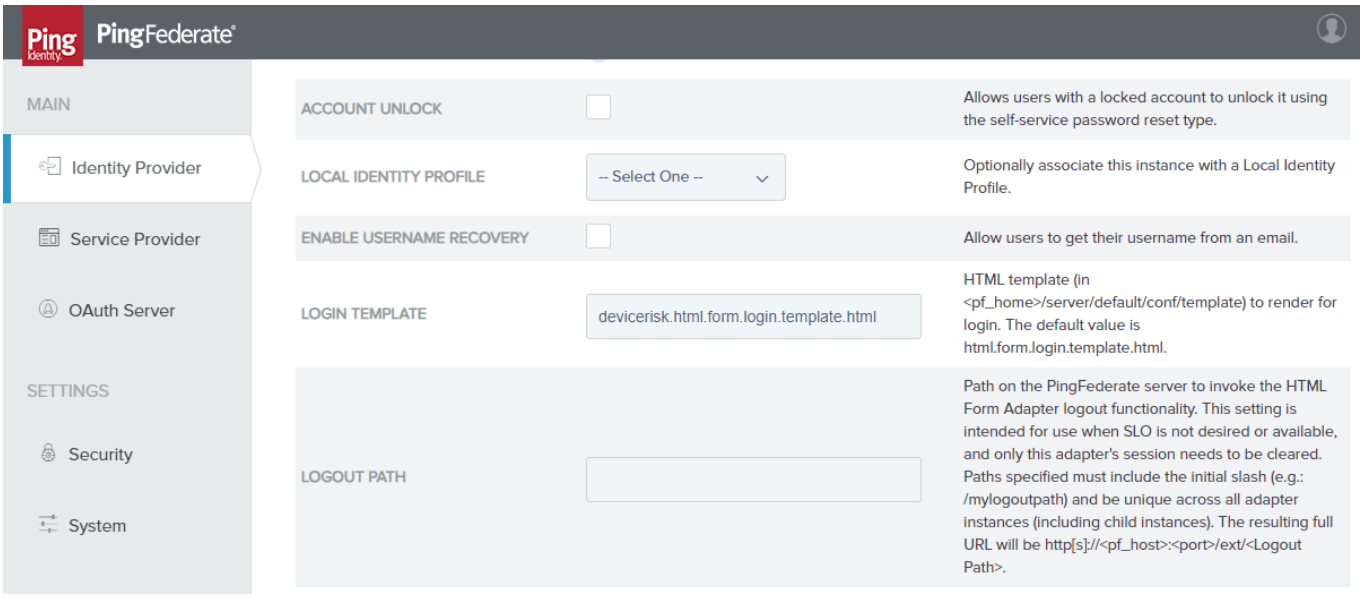
13. Click **Next** again and once on **Adapter Attributes** tab and tick the checkboxes **Pseudonym** for both "**risk_level**" and "**username**". Once done click **Next** in the next two pages that appear and once Summary page is reached, click **Done** and in the next page click **Save** to save the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



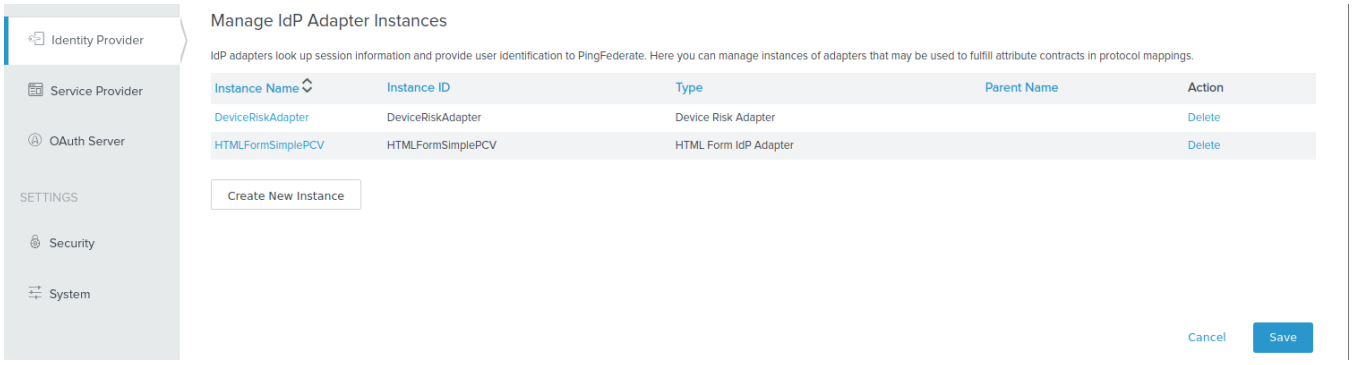


14. In the Identity Provider page go to **Integration** > **Adapters**, click **HTMLFromSimplePCV** then **IdP Adapter**
15. Scroll to the bottom of the page and click **Show Advanced Fields**
16. Find the **Login Template** field and replace the existing value with

devicerisk.html.form.login.template.html

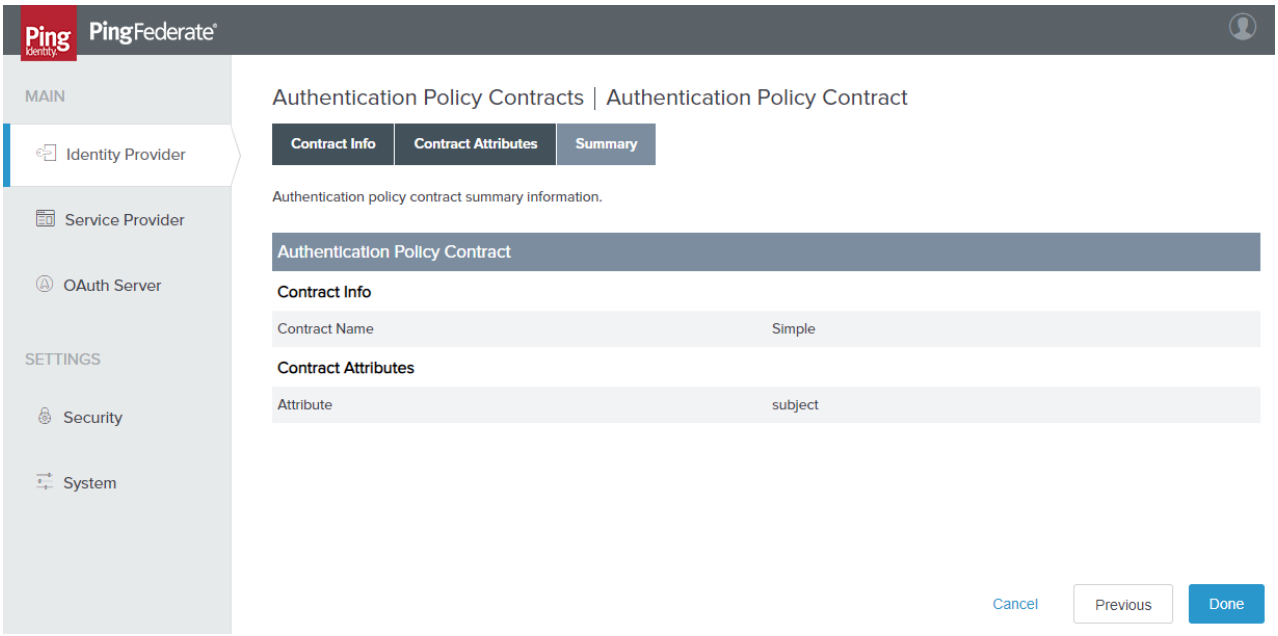


Once done click **Done.** In the next page, click **Save**



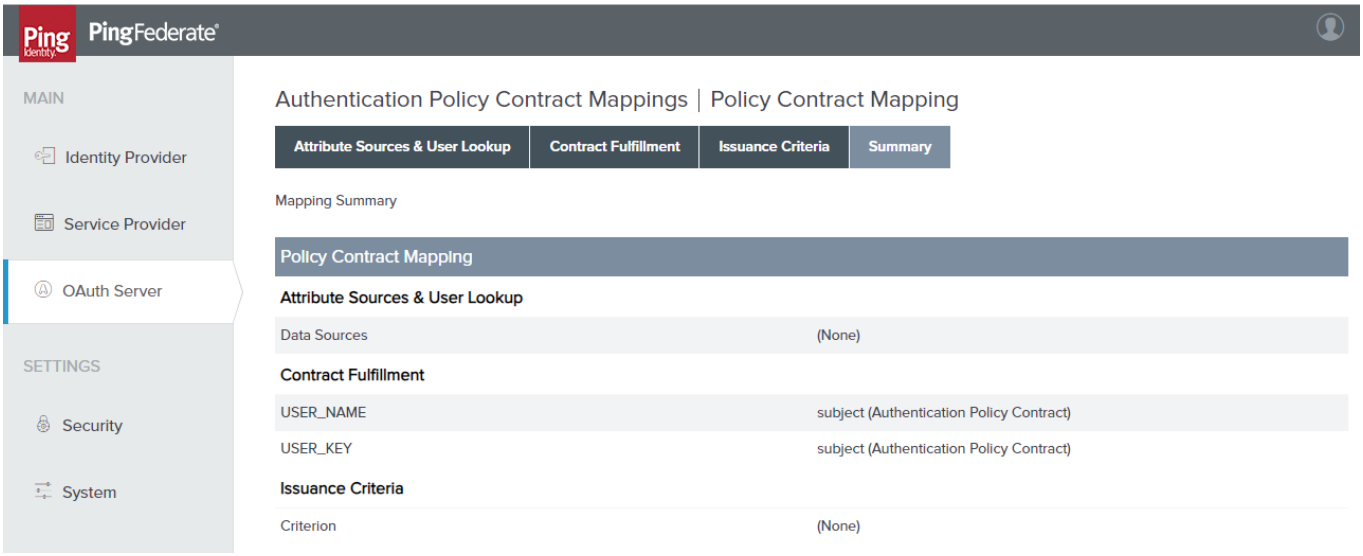
The next step is to define **Policy Contracts**: this will be used by Ping Federate to oppose different authentication steps to each user based on his risk level.

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**

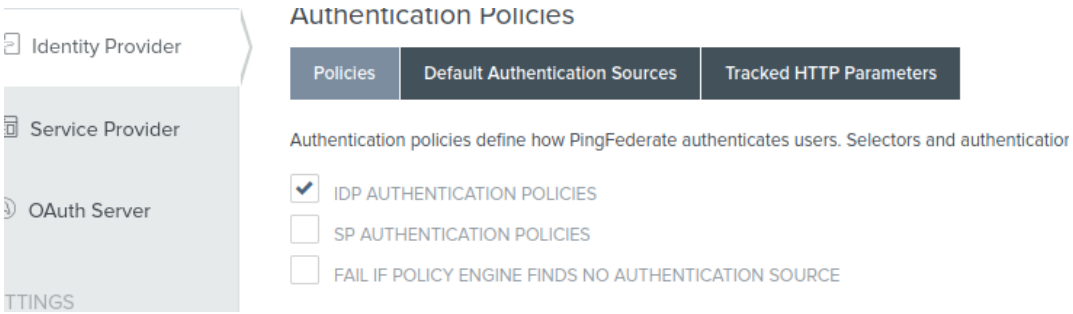


19. Click **OAuth Server > Grant Mapping > Authentication Policy Contract Mapping** and select from the drop-down menu the contract created in the previous step, then click **Add Mapping > Next** which takes you to the **Contract Fulfillment** tab
20. For both **USER_KEY** and **USER_NAME** contracts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



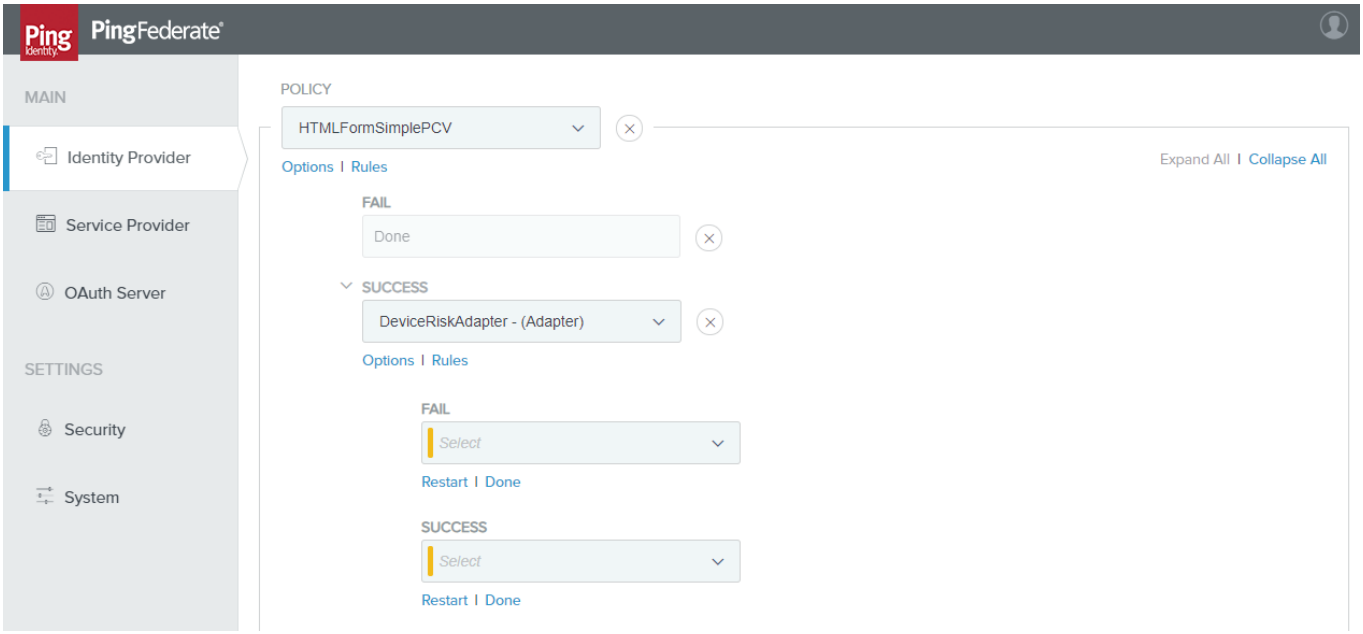
Click **Identity Provider > Authentication Policies > Policies**, and make sure that IDP AUTHENTICATION POLICIES checkbox is selected.



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

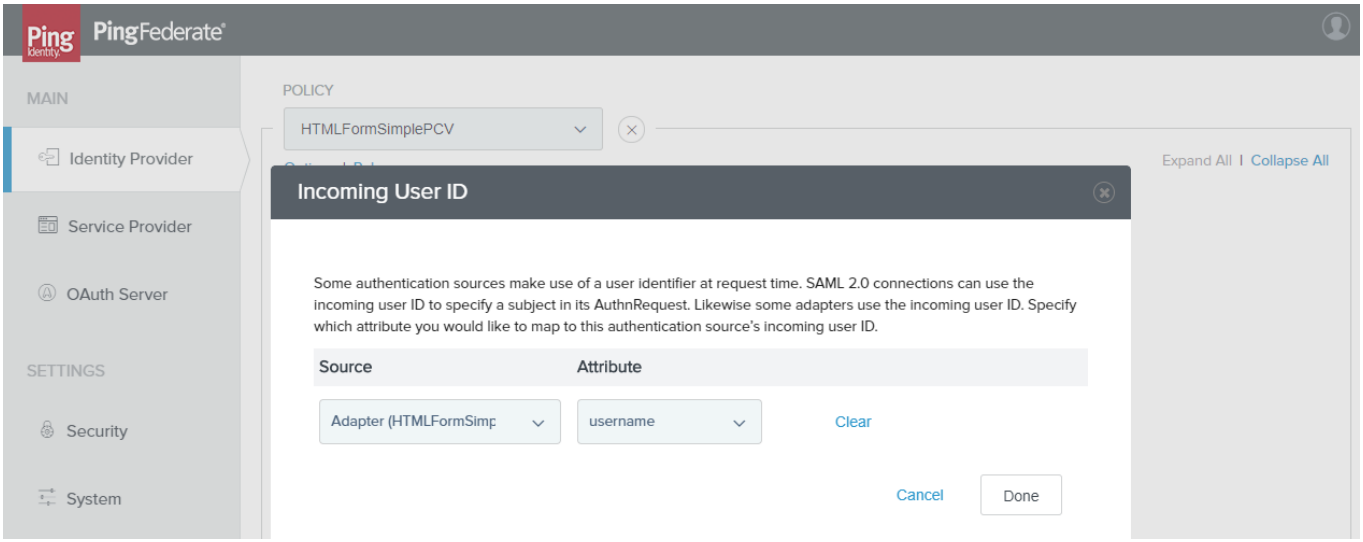
    For the **Fail** case click **Done**, while for the **Success** case click on the drop-down menu and select **DeviceRiskAdapter** from the **IdP Adapters** menu



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

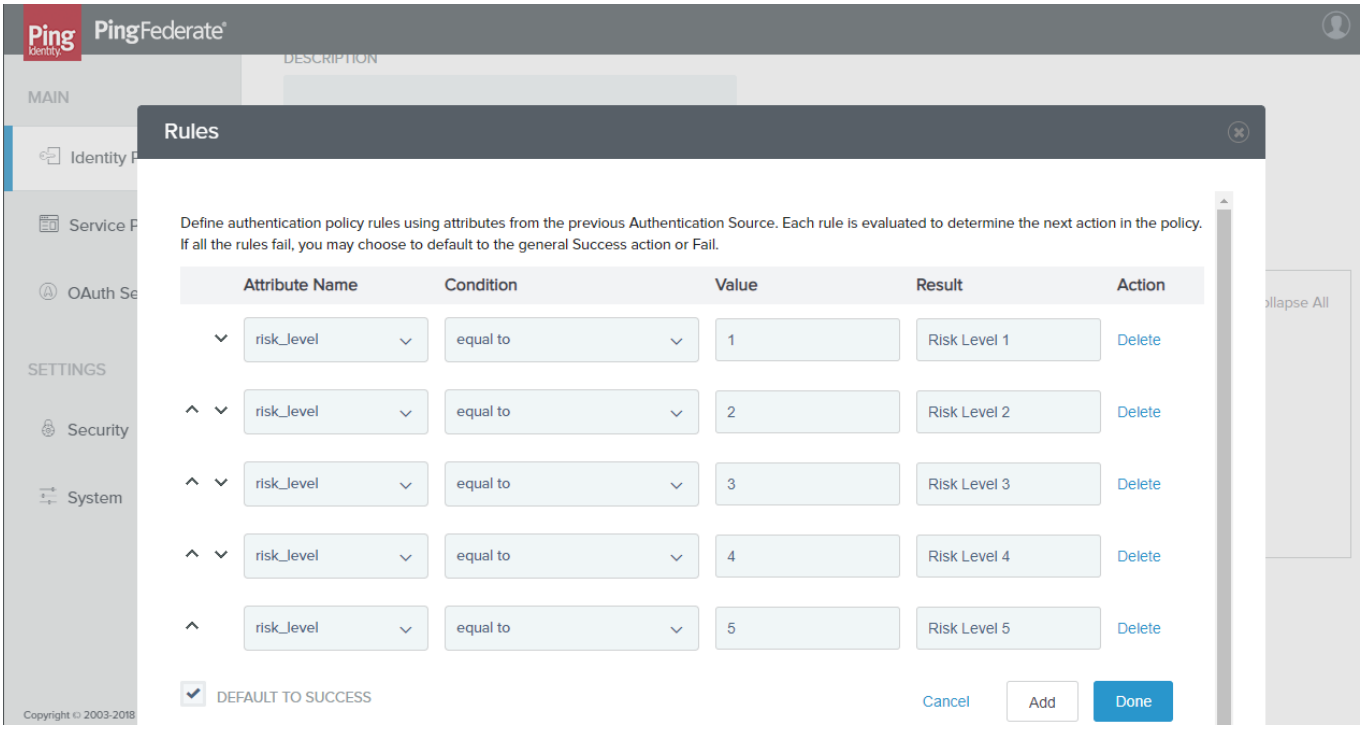
    Source: Adapter (HTMLFormSimplePCV)

    Attribute: username



Click **Done** once finished

23. Under the top **Success** drop-down now click **Rules** and add one line for each possible risk level value returned by Forcepoint CASB 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.



The **Risk Exporter** returns risk level -1 for entities that do not exist. 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 they have no risk level.

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**

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.

Example – Authentication policy rules
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 CASB.

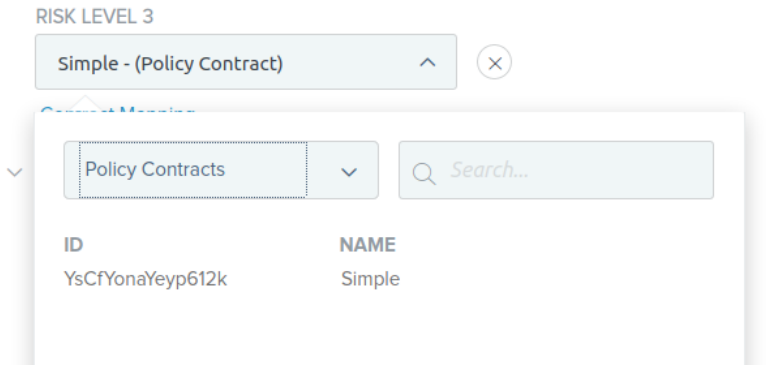
The configuration described in the next pages

→ 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 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)



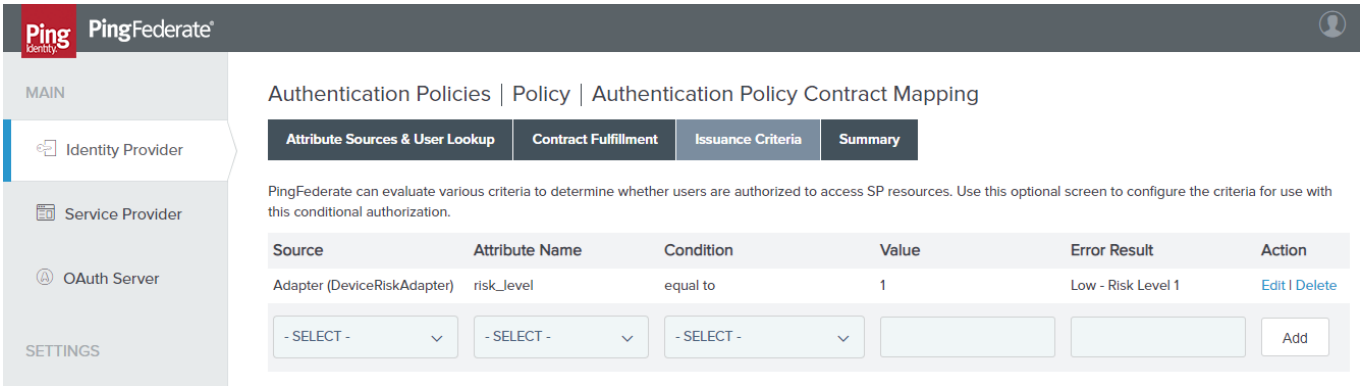
4. Click Contract Mapping
5. Click **Next** in the **Attribute Source & User Lookup** page and in the **Contract Fulfillment** page use the following choices in the drop-down menus then click **Next**

   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**.



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

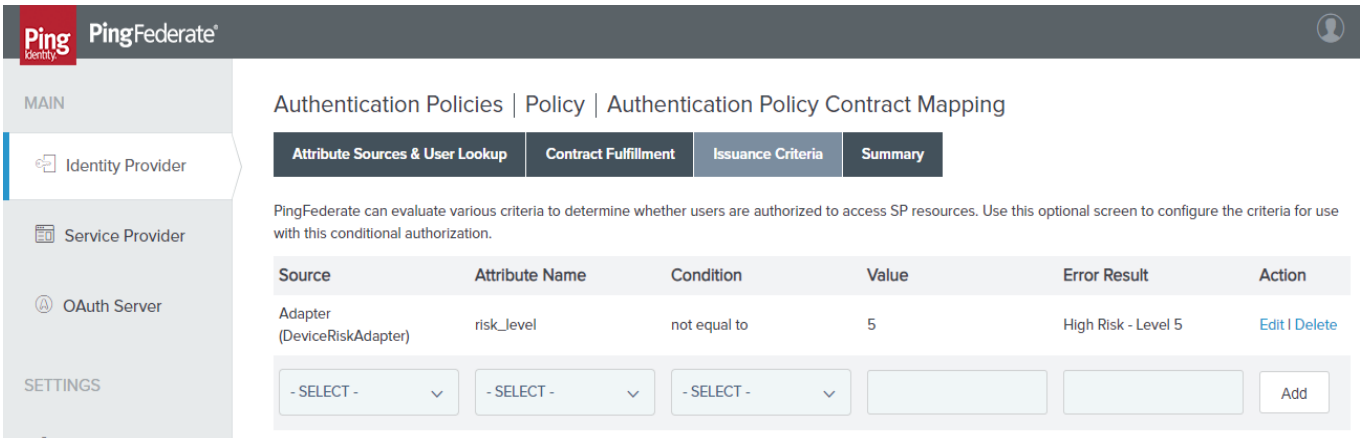
7. For **Risk Level 5** click **Contract Mapping** then **Next** in the **Attribute Source & User Lookup** page since no changes are to be made in this page. In the **Contract Fulfillment** page use the following choices in the drop-down menus then click **Next**

   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**.



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.

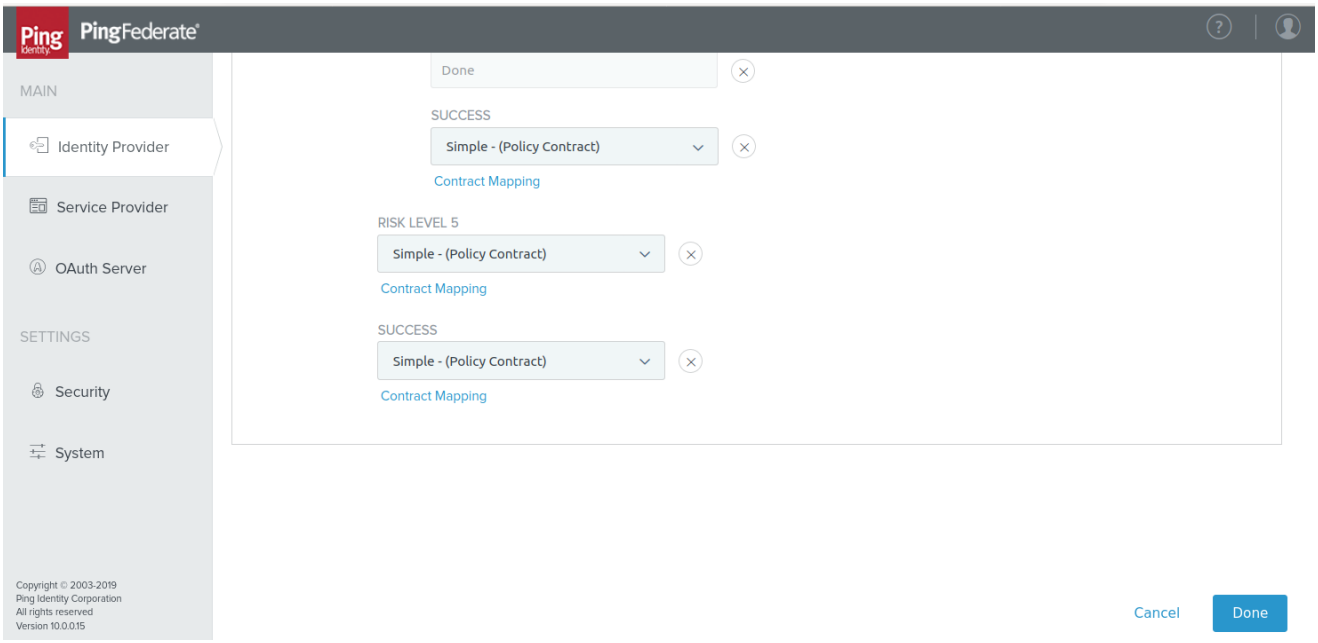
For **Success** under **DeviceRiskAdapter**, click on the drop-down menu and on the next drop-down menu select **Policy Contracts**, select **Simple** (created earlier in step 18 of Step 2 – Setup Ping Federate).

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**

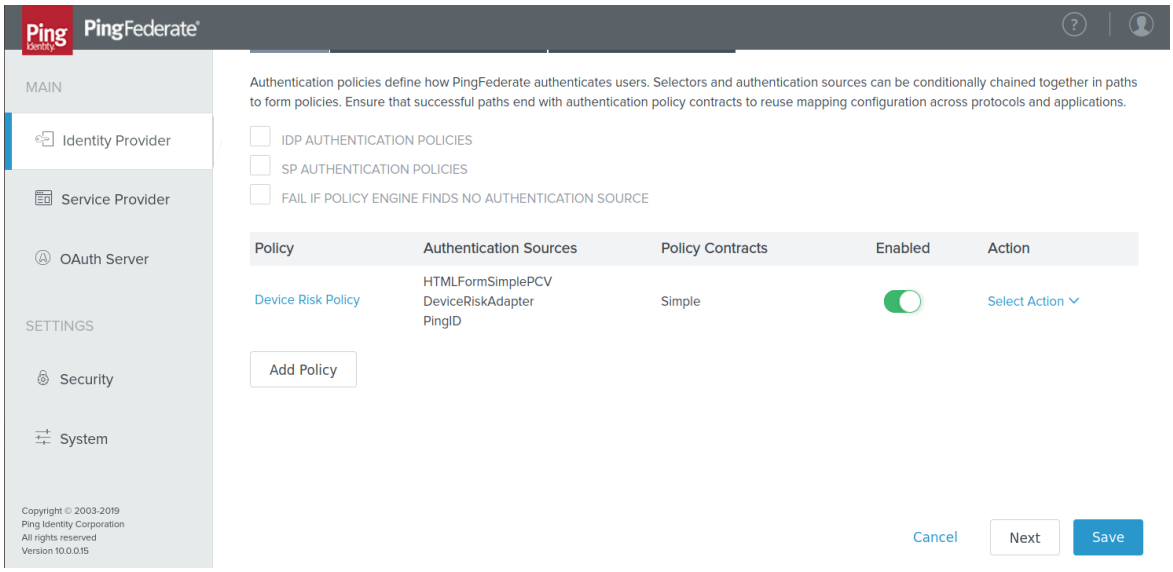
   Source: Adapter (HTMLFormSimplePCV)

   Value: username

2. Once done click **Next** then **Next**. Click **Done**.
3. Once all the risk levels are mapped correctly, click **Done** at the bottom of the page.



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



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 CASB and Ping Federate in use are listed as compatible

Forcepoint CASB 2020 R2
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 and 20GB free storage and the system needs to be 64-bit
→ The API port must be accessible through the firewall
→ The Risk Exporter API SSL certificate must be trusted by the Ping **host machine**
→ User must have sudo permissions to run **install.sh** and **setup.sh** scripts
→ Check that 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

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, execute the following command on the **host machine**:

curl -I https://<Risk Exporter hostname or IP>:5000

replacing the Risk Exporter hostname with the ones in use. Please check the first line of the result of the command 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/casb/healthcheck

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

casb-url available - OK!

casb-url 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.

Docker Implementation

Validate the prerequisites

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

→ Check the versions of Forcepoint CASB and Ping Federate in use are listed as compatible

Forcepoint CASB 2020 R2
Ping Federate 9.2 or 10.0

→ Verify the integration component correctly operates on a linux based machine with an existing docker engine and a minimum of 2GB RAM and 20GB free storage and the system needs to be 64-bit

→ The Risk Exporter API SSL certificate 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. Execute the following command on the host machine:

curl -I https://<Risk Exporter hostname or IP>:5000

replacing the Risk Exporter hostname with the ones in use. Please check the first line of the result of the command 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/casb/healthcheck

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

the healthcheck URL:

casb-url available - OK!

casb-url connection is successful - OK!

→ 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

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.