2016

SAFAX

External Clients

Policy Evaluation Deploy Policy

TU/e Security Group

Table of Contents

Introduction

SAFAX [1] is an extensible authorization framework offered as a service. SAFAX provides a novel XACML-based architectural framework tailored to the development of extensible authorization services for distributed and collaborative systems. The key design principle underlying SAFAX is that all components are loosely coupled services, thus providing the flexibility, extensibility and scalability needed to manage authorizations in cloud environments. SAFAX‘s architecture allows users to: *a)* deploy their access control policies in a standard format; *b)* in a single location; and *c)* augment policy evaluation with information from user selectable external trust services.

This document presents the steps to configure third-party software systems to access SAFAX programmatically. These software systems can also be considered clients of SAFAX as they consume SAFAX authorization service. To access SAFAX programmatically, we provide software clients that software systems can use to:

- Upload an XACML policy to an existing demo.
- Evaluate an XACML request using a client code.
- Evaluate an XACML request using username/password.

Clients can be written in any programming language that supports creating web service clients. For testing purposes, we describe the realization of the clients in Java.

List of Acronyms and Definitions

List of Acronyms

CH: Context Handler

PAP: Policy Administration Point

PEP: Policy Enforcement Point

PIP: Policy Information Point

SAFAX: eXtensible Authorization Framework As a Service

UUID: Universally Unique IDentifier

XACML: eXtensible Access Control Markup Language

List of Definitions

Administrator: a user who is responsible for the management of SAFAX.

General Steps

To access SAFAX programmatically, users need to register within SAFAX service. More specifically;

- They need to obtain a username/password by visiting SAFAX website.

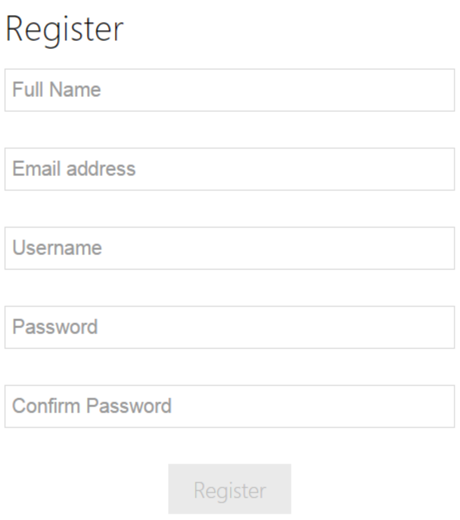
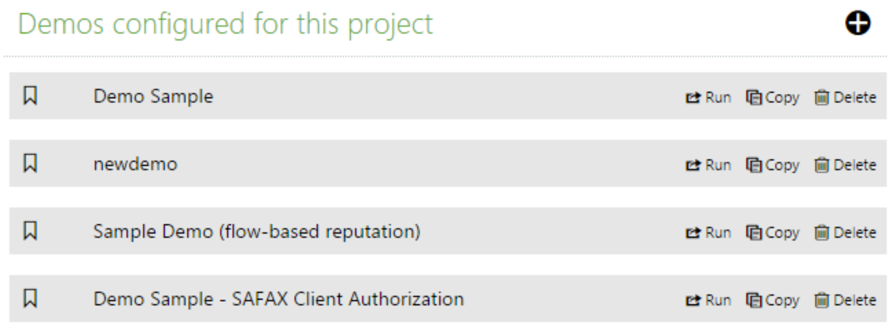


Figure 1 SAFAX Registration GUI Menu

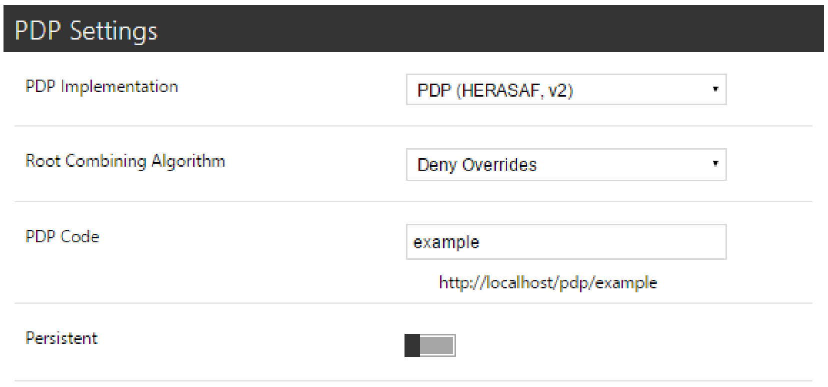
- After login with the username/password users can create a project and a demo within this project.


Demos configured for this project

Demo Sample	Run	Copy	Delete
newdemo	Run	Copy	Delete
Sample Demo (flow-based reputation)	Run	Copy	Delete
Demo Sample - SAFAX Client Authorization	Run	Copy	Delete

Figure 2 List of Demos in a Project

- For each demo, SAFAX creates a PDP instance and a PDP code. Demos can be configured according to user preferences.


PDP Settings

PDP Implementation	PDP (HERASAF, v2)
Root Combining Algorithm	Deny Overrides
PDP Code	example http://localhost/pdp/example
Persistent	

Figure 3 PDP Code and PDP Settings

Policy Evaluation

A third-party software system asks SAFAX to evaluate a policy against an XACML request. There are two options to use SAFAX evaluate web service:

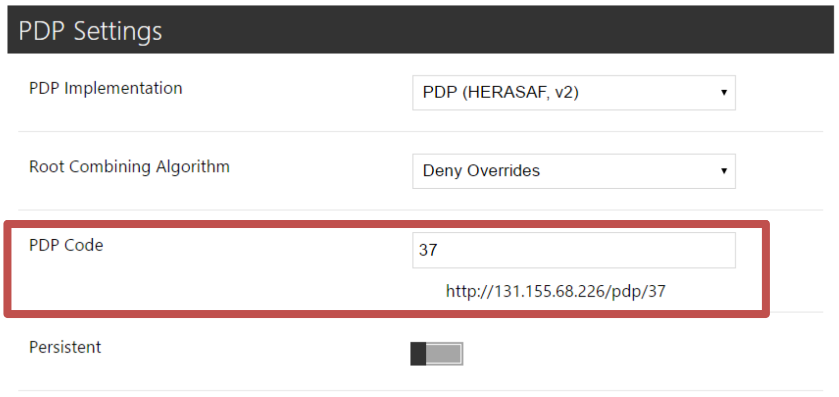
1. Use a client code
2. Use a username/password

Use a client code

A third-party software system consumes the following web service provided by SAFAX:

http://131.155.68.226/pep/evaluate/request/{pdpcode}

The *{pdpcode}* refers to the demo with respect to which the request should be evaluated. The *pdpcode* can be found by selecting the appropriate demo and then navigating to the *PDP Settings* (see Figure 4).



PDP Settings

PDP Implementation	PDP (HERASAF, v2)
Root Combining Algorithm	Deny Overrides
PDP Code	37 http://131.155.68.226/pdp/37
Persistent	

Figure 4 PDP Code

The third-party software system also needs to provide the following parameters:

- *clientcode*: this is the key through which SAFAX authenticates client and is provided by SAFAX administrators.
- *transactionid*: an immutable universally unique identifier (UUID). A UUID represents a 128-bit value. The *transactionid* is provided by client and for example, can be obtained using the *randomUUID()* method of Java UUID class.
- *request*: an XACML request.

In Java, this can be done by creating a Form object and attaching the three parameters to this object.

```
Form form=new Form();
form.add("transactionid",transactionid);
form.add("clientcode",clientkey);
form.add("request", req);
```

Figure 5 Java Code Snippets Adding Form Parameters

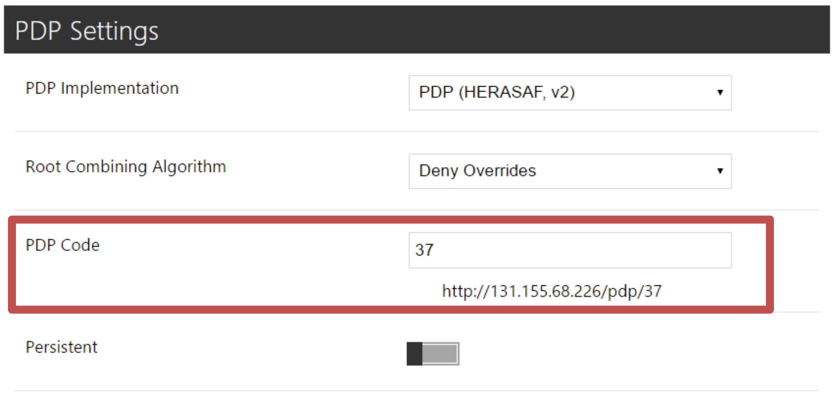
The client consumes the authorization service provided by SAFAX. SAFAX checks the client code provided by the client. If the code is valid, SAFAX evaluates the request and returns the XACML response to the third-party software system.

Use a username/password

A third-party software system consumes the following web service provided by SAFAX:

http://131.155.68.226/sfxservice/policy/client/load/request/{pdpcode}

The *{pdpcode}* refers to the demo with respect to which the request should be evaluated. The code can be found by choosing a demo and then navigating to the *PDP Settings*.



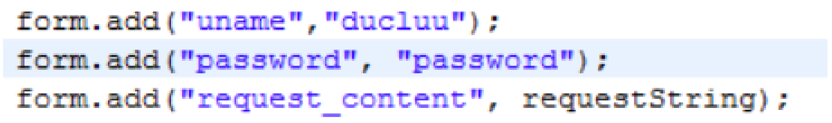
PDP Settings

PDP Implementation	PDP (HERASAF, v2)
Root Combining Algorithm	Deny Overrides
PDP Code	37 http://131.155.68.226/pdp/37
Persistent	

Figure 6 PDP Code

The third-party software system also needs to include the following form parameters:

- *uname*: this is the username provided by SAFAX.
- *password*: this is the password associated with a username.
- *request_content*: an XACML request.



```
form.add("uname","ducluu");
form.add("password", "password");
form.add("request_content", requestString);
```

Figure 7 Java Code Snippets Adding Form Parameters

The client consumes the authorization service provided by SAFAX. SAFAX checks the username and password provided by the client. SAFAX uses its own authorization service to check if the account associated with the client has the privilege to evaluate the request using the given PDP instance. If the account is allowed to evaluate the request against a policy contained in the given demo, SAFAX forwards the request to the PDP instance for evaluation and returns the XACML response to the client.

.

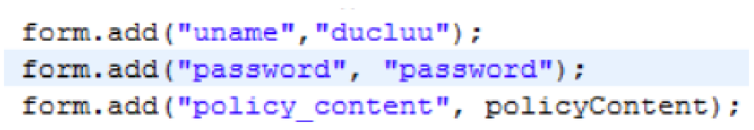
Deploy Policy

A third-party software system can deploy a policy to an existing demo within SAFAX by consuming the following web service provided by SAFAX:

http://131.155.68.226/sfxservice/policy/client/upload/xacml/{demoid}

The third-party software system needs to include the following form parameters:

- uname: this is the username.
- password: this is the password.
- policy_content: the content of the policy.



```
form.add("uname","ducluu");
form.add("password", "password");
form.add("policy_content", policyContent);
```

Figure 8 Java Code Snippets Adding Form Parameters

The client consumes the authorization service provided by SAFAX. In particular, the client sends a request to upload the given policy within a certain demo. SAFAX checks the username and password provided by the client. SAFAX uses its own authorization service to check if the account associated to the client is allowed to deploy the policy in the demo. If the account is authorized to deploy in the given demo, SAFAX forwards the request to the PAP service. SAFAX returns a notification to the client whether the policy is successfully deployed or not.

.

Java Client Example Codes

The previous sections have presented the steps describing how to consume SAFAX authorization services without limitation to a particular programming language. In this section, we present some sample clients implemented in Java and how they can be configured.

The example codes can be found in the SVN of the Security Group. We refer to the SAFAX Installation Guide [2] for the configuration of SVN and of the SAFAX project.

Three Java client applications are available:

1. PAPClient.java
2. SFXClient.java
3. SFXClientWithAuthorization.java

You can use Eclipse to open the project. Right-click any of the three client applications and choose *Run As -> Java Application*.

SFXClient.java

This client is used to evaluate an XACML request against the XACML policies defined in a given demo using a client code. The client requires the {*pdpcode}* corresponding to the demo to be used to evaluate the access request. Change the content of *serviceurl* at line 17 by using the appropriate {*pdpcode*}. This code can be obtained from the demo setting (see Figure 4).

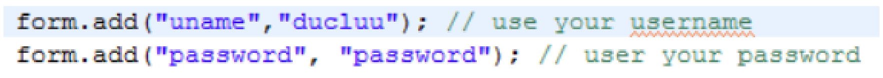
At line 20, a *transactionid* is automatically generated.

At line 25, use the *clientcode* provided by SAFAX administrators.

SFXClientWithAuthorization.java

This client is used to evaluate an XACML request against the XACML policies defined in a given demo using a username/password. The client requires the {*pdpcode}* to use the service. Change the content of *serviceurl* at line 17 by using the {*pdpcode*} corresponding to the demo to be used. This code can be obtained from the demo setting (see Figure 4).

At line 65 and 66 use your username and password provided by SAFAX.



```
form.add("uname","ducluu"); // use your username
form.add("password", "password"); // user your password
```

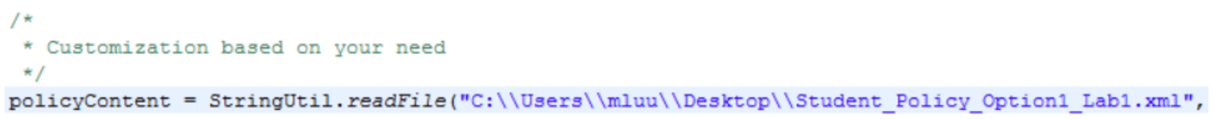
Figure 9 Username and Password Parameters

PAPClient.java

This client is used to upload a XACML policy to an existing demo in SAFAX. The client should specify the *demoid* based on actual need. In particular, the *demoid* indicates the demo in which a given XACML policy should be programmatically uploaded. This can be done by setting the content of the *serviceurl* variable at line 22.

http://131.155.68.226/sfxservice/policy/client/upload/xacml/{demoid}

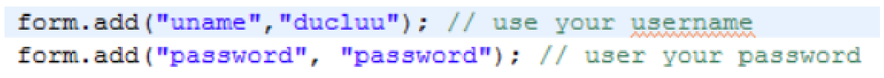
At line 27 change the first parameter of the *readFile()* to reference to the policy file.



```
/*
 * Customization based on your need
 */
policyContent = StringUtil.readFile("C:\\Users\\mluu\\Desktop\\Student_Policy_Option1_Lab1.xml",
```

Figure 10 Java Code Snippets *serviceurl* Variable

At line 36 and 37 use your username and password provided by SAFAX.



```
form.add("uname","ducluu"); // use your username
form.add("password", "password"); // user your password
```

Figure 11 Username and Password Parameters

References

[1] S. P. Kaluvuri and A. I. Egner and J. den Hartog and N. Zannone. SAFAX -- an extensible authorization service for cloud environments. Frontiers in ICT 2(9), 2015.

[2] SAFAX Installation Guide, 2015.