Installation and configuration of uApprove Jet Pack 3.2.0

This document contains the uApprove Jet Pack 3.2(in short, uApprove JP) deployment guide and the general manual.

The uApprove JP is an extension for the Shibboleth Identity Provider 3.x. It is intended to be made available part of the functions supported by the uApprove Jet Pack 2.5 on the Shibboleth Identity Provider. It allows users authenticating at an Identity Provider to release attribute selectively. See more information about the concept of the uApprove JP.

Notes about this guide:

- This guide assumes that the uApprove JP is installed on a Linux system. It's also possible to install it on a different operating system, like Windows. In this case, you may need to adapt some paths and commands accordingly.
- The guide shows paths and commands using variables like, e.g. `$IDP_HOME$` or `$UAPPROVE_INSTALL$`. You need to substitute these variables by the real paths, except where it is explicitly stated that you don't need to substitute them.

Table of Contents

Assumptions

- The Shibboleth Identity Provider is installed at `$IDP_HOME$` (e.g., `/opt/shibboleth-idp`).
- The Tomcat is installed at `$CATALINA_HOME$` (e.g., `/usr/java/tomcat`), and IdP's Instance is configured at `$CATALINA_BASE$`(e.g., `$CATALINA_HOME$`).
- The uApprove JP is downloaded and unpacked at `$UAPPROVE_INSTALL$` (e.g., `/usr/local/src/uApproveJP-#version#`).

Prerequisites

- Shibboleth Identity Provider 3.2.0 or later.

1 Basic Deployment

1.1 Library installation

Copying the libraries to the IdP's library directory:

```
# cp $UAPPROVE_INSTALL$/lib/*.jar $IDP_HOME$/edit-webapp/WEB-INF/lib
```

⚠ Assure that only one version of each library is present in IDP_HOME/edit-webapp/WEB-INF/lib.

1.2 Velocity template file

Copying the Velocity template file for view of attributes selection to the IdP's views directory:

```
# cp $UAPPROVE_INSTALL$/manual/examples/views/intercept/* $IDP_HOME$/views/intercept
```

1.3 CSS file

Copying the CSS file to the IdP's edit-webapp directory:

```
# cp $UAPPROVE_INSTALL$/manual/examples/edit-webapp/css/* $IDP_HOME$/edit-webapp/css
```

1.4 Custom of Configuration

Modify idp.consent.allowPerAttribute and idp.consent.compareValues to true in IDP_HOME/conf/idp.properties:

IDP_HOME/conf/idp.properties

```
...
# Flags controlling how built-in attribute consent feature operates
#idp.consent.allowDoNotRemember = true
#idp.consent.allowGlobal = true
idp.consent.allowPerAttribute = true

# Whether attribute values and terms of use text are compared
idp.consent.compareValues = true
...
```

Add bean named "shibboleth.FallbackLanguages" and bean named "shibboleth.CustomViewContext" for view of attribute selection as below in IDP_HOME/conf/global.xml:

IDP_HOME/conf/global.xml

```
...
<bean id="shibboleth.FallbackLanguages" parent="shibboleth.CommaDelimStringArray" c:_0="#{'%{idp.ui.fallbackLanguages:}'.trim()}" />

<util:map id="shibboleth.CustomViewContext">

    <entry key="OptionalAttributeFunction">

        <bean class="jp.gakunin.idp.consent.logic.impl.OptionalAttributeFunction" />

    </entry>

    <entry key="AttributeIntendedUseFunction">

        <bean class="jp.gakunin.idp.consent.logic.impl.AttributeIntendedUseFunction" p:defaultLanguages-ref="shibboleth.
FallbackLanguages" />

    </entry>

</util:map>

...
```

Modify <constructor-arg name="strategy"> in bean named "shibboleth.AttributeFilterService" to as below in IDP_HOME/system/conf/services-system.xml:

> ⚠ You must make the change again when you install the Shibboleth IdP again(upgrade, etc.) because the change will be overwritten.

IDP_HOME/system/conf/services-system.xml

```
...
<bean id="shibboleth.AttributeFilterService" class="net.shibboleth.ext.spring.service.ReloadableSpringService"

        depends-on="shibboleth.VelocityEngine"

        p:serviceConfigurations-ref="#{'%{idp.service.attribute.filter.resources:shibboleth.AttributeFilterResources}'.trim()}"

        p:failFast="%{idp.service.attribute.filter.failFast:%{idp.service.failFast:false}}"

        p:reloadCheckDelay="%{idp.service.attribute.filter.checkInterval:PT0S}"

        p:beanFactoryPostProcessors-ref="shibboleth.PropertySourcesPlaceholderConfigurer">

    <constructor-arg name="claz" value="net.shibboleth.idp.attribute.filter.AttributeFilter" />

    <constructor-arg name="strategy">

        <bean class="jp.gakunin.idp.attribute.filter.spring.impl.AttributeFilterServiceStrategy"

id="ShibbolethAttributeFilter"/>

    </constructor-arg>

</bean>

...
```

Modify the class definition in bean named "IsConsentRequiredPredicate" to as below in IDP_HOME/system/flows/intercept/attribute-release-beans.xml:

> ⚠ You must make the change again when you install the Shibboleth IdP again(upgrade, etc.) because the change will be overwritten.

IDP_HOME/system/flows/intercept/attribute-release-bean.xml

```
...
<bean id="IsConsentRequiredPredicate"
    class="jp.gakunin.idp.consent.logic.impl.IsConsentRequiredPredicate" />
...
```

1.5 Custom Templates

In case you want to customize the templates, follow section Custom View Templates.

At least, you should change to your organization's logo, since a placeholder logo is installed by default.

Modify to the federaion logo to as below:

1. Copying the logo file `organization-logo.png` to IdPs `edit-webapp/images/` directory:

```
$ ls $IDP_HOME$/edit-webapp/images/
dummylogo-mobile.png  dummylogo.png  organization-logo.png
```

2. Modify configuration in `$IDP_HOME$/messages/error-messages.properties`. Modify `idp.logo` to the filename of copied at above. Note that, the filename begin at `/images/`. In addition, modify the `idp.logo.alt-text`:

IDP_HOME/messages/error-messages.properties

```
idp.logo = /images/organization-logo.png

idp.logo.alt-text = Organization logo
```

3. Rebuild the `idp.war`:

```
$ cd $IDP_HOME$
$ sudo -u tomcat env JAVA_HOME="${JAVA_HOME}" bin/build.sh
Installation Directory: [/opt/shibboleth-idp]

Rebuilding /opt/shibboleth-idp/war/idp.war ...
...done

BUILD SUCCESSFUL
Total time: 16 seconds
```

1.6 Logging

Add as below to activate logging of the uApprove JP in `$IDP_HOME$/conf/logback.xml`:

IDP_HOME/conf/logback.xml

```
...
<!-- Logging level shortcuts. -->
<variable name="idp.loglevel.uApproveJP" value="INFO" />
...
<!-- ========================================================= -->
<!-- ============== Logging Categories and Levels ============== -->
<!-- ========================================================= -->
<logger name="jp.gakunin.idp" level="${idp.loglevel.uApproveJP:-INFO}"/>
...
```

1.7 Deployment

To activate the uApprove JP, the IdP must be re-deployed:

```
# cd $IDP_HOME$
# ./bin/build.sh
Installation Directory: [/opt/shibboleth-idp]

Rebuilding /opt/shibboleth-idp/war/idp.war ...
...done

BUILD SUCCESSFUL
Total time: 16 seconds
```

Copy idp.war to $CATALINA_BASE$/webapps:

```
# cp $IDP_HOME$/war/idp.war $CATALINA_BASE$/webapps/
```

Restart Tomcat:

```
# service tomcat7 restart
```

2 Advanced Deployment

This section contains advanced configuration topics.

2.1 How to save the consent information of user into Relational Database

You can be use the relational database (in short, RDB) to save the user's consent information.

2.1.1. MySQL Configuration

The following database parameters are example. Change as necessary the actual value. Please prepare a safe particular password.

Setting MySQL.

1. Create Database
   Create a database shibboleth used by the Shibboleth IdP:

```
db$ mysql -u root
mysql>
CREATE DATABASE shibboleth;
```

2. Create User
   Create a user shibboleth for access to database shibboleth from IdP. In addition, grant permissions to the database shibboleth to the created user:

```
db$ mysql -u root
mysql>
CREATE USER 'shibboleth'@'localhost' IDENTIFIED BY 'shibpassword';          # ←arbitrary password
GRANT INSERT, SELECT, UPDATE, DELETE ON shibboleth.* TO 'shibboleth'@'localhost';
```

3. Create Table

   Create a table StorageRecords used by JPAStorageService:

```
db$ mysql -u root
mysql>
use shibboleth;
CREATE TABLE `StorageRecords` (
    `context` varchar(255) NOT NULL,
    `id` varchar(255) NOT NULL,
    `expires` bigint(20) DEFAULT NULL,
    `value` longtext NOT NULL,
    `version` bigint(20) NOT NULL,
    PRIMARY KEY (`context`,`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

2.1.3. MySQL Connector/J Installation

1. Install the MySQL Connector/J (mysql-connector-java.jar) required to access the MySQL:

```
# yum install mysql-connector-java
```

2. The libraries are installed under /usr/share/java. Create a symbolic link from they to the lib directory of Tomcat:

```
# rpm -ql mysql-connector-java
...
/usr/share/java/mysql-connector-java.jar
...
# ln -s /usr/share/java/mysql-connector-java.jar $CATALINA_BASE$/lib/
```

2.1.4. idp.consent.StorageService Configuration

Modify idp.consent.StorageService to shibboleth.JPAStorageService:

IDP_HOME/conf/idp.properties

```
# Set to "shibboleth.StorageService" or custom bean for alternate storage of consent

idp.consent.StorageService = shibboleth.JPAStorageService
# Maximum number of consent storage records
# Set to -1 for unlimited server-side storage
idp.consent.maxStoredRecords = -1
```

2.1.5. shibboleth.JPAStorageService Configuration

Define the shibboleth.JPAStorageService set to idp.sesssion.StorageService at setting change of 2.1.4. idp.consent.StorageService Configuration.

Set the properties(p:url, p:username, p:assword) in bean named "Shibboleth.MySQLDataSource" according to of 2.1.1. MySQL Configuration:

IDP_HOME/conf/global.xml

```
<!-- Use this file to define any custom beans needed globally. -->
<bean id="shibboleth.JPAStorageService"
      class="org.opensaml.storage.impl.JPAStorageService"
      p:cleanupInterval="%{idp.storage.cleanupInterval:PT10M}"
      c:factory-ref="shibboleth.JPAStorageService.entityManagerFactory" />

<bean id="shibboleth.JPAStorageService.entityManagerFactory"
      class="org.springframework.orm.jpa.LocalContainerEntityManagerFactoryBean">
     <property name="packagesToScan" value="org.opensaml.storage.impl" />
     <property name="dataSource" ref="shibboleth.MySQLDataSource" />
     <property name="jpaVendorAdapter" ref="shibboleth.JPAStorageService.JPAVendorAdapter" />
     <property name="jpaDialect">
          <bean class="org.springframework.orm.jpa.vendor.HibernateJpaDialect" />
     </property>
</bean>

<bean id="shibboleth.JPAStorageService.JPAVendorAdapter"
      class="org.springframework.orm.jpa.vendor.HibernateJpaVendorAdapter"
      p:database="MYSQL" />

<bean id="shibboleth.MySQLDataSource"
      class="org.apache.tomcat.dbcp.dbcp.BasicDataSource"
      p:driverClassName="com.mysql.jdbc.Driver"
      p:url="jdbc:mysql://localhost:3306/shibboleth"
      p:username="shibboleth"
      p:password="shibpassword"
      p:maxActive="10"
      p:maxIdle="5"
      p:maxWait="15000"
      p:testOnBorrow="true"
      p:validationQuery="select 1"
      p:validationQueryTimeout="5" />
```

2.1.6. Restart Tomcat

Restart Tomcat:

```
# service tomcat7 restart
```

2.2 Templates

Custom View Templates

Feel free to customize the Velocity templates, CSS and image files located in IDP_HOME/edit-webapp/.
For convenience the Velocity is used, cf. Velocity User Guide.

2.3 Localization

Custom Messages

You might adjust/extend the provided resource bundles in IDP_HOME/messages.

Must be restarted Tomcat if was adjusted or extended.

There is template files for Japanese localization at $UAPPROVE_INSTALL$/manual/examples/messages/. It will be to display Japanese messages by copying the template files to IDP_HOME/messages/.

Must be restarted Tomcat to activate:

```
# cp $UAPPROVE_INSTALL$/manual/examples/messages/* $IDP_HOME$/messages/
# service tomcat7 restart
```

Relying Party Names and Descriptions

Currently only <AttributeConsumingService> element in metadata is supported to retrieve localized relying party names and descriptions.
For providing such names and descriptions extend the metadata for the SP like:

```
<EntityDescriptor entityID="https://sp.example.org/shibboleth">

    <!-- ... -->

    <SPSSODescriptor>

        <Extensions>

            <mdui:UIInfo xmlns:mdui="urn:oasis:names:tc:SAML:metadata:ui">

                <mdui:DisplayName xml:lang="en">Example SP</mdui:DisplayName>

                <!-- Service names in other languages -->

                <mdui:Description xml:lang="en">Some description of Example SP</mdui:Description>

                <!-- Service descriptions in other languages -->

            </mdui:UIInfo>

        </Extensions>

        <!-- ... -->

        <AttributeConsumingService index="1">

            <ServiceName xml:lang="en">Example SP</ServiceName>

            <!-- Service names in other languages -->

            <ServiceDescription xml:lang="en">Some description of Example SP</ServiceDescription>

            <!-- Service descriptions in other languages -->

        </AttributeConsumingService>

    </SPSSODescriptor>

</EntityDescriptor>
```

If they are both described <mdui: UIInfo> element takes precedence.

Also, it contains this information about SP of GakuNin except for some overseas SP.

2.4 Attribute In Attribute Requester’ s Metadata Plugin

Configuration Attribute In Requester's Metadata Matching Rule

This rule allows the release of an attribute if, via its metadata, the SP indicates a need/desire for the attribute. The attributes are indicated by means of <AttributeConsumingService> element within the<SPSSODescriptor> element. Attributes with isRequired=’true’ at <RequestedAttribute> is marked as required, and with isRequired=’false’ is marked as optional. See SAML metadata for more information.

Please be aware of the following:

- This filter requires the attribute requester’ s metadata be loaded and available.
- The requester’ s metadata must have an <SPSSODescriptor> role since that is the role that contains the listed attributes.
- This matching function only operates as a value rule and only really makes sense as a permit value rule.

Define the Namespace

In your attribute filter policy file you’ ll need to add the namespace declaration for this plugin. To do this:

- Add the attribute xmlns:uajpmf="http://www.gakunin.jp/ns/uapprove-jp/afp/mf" before the xmlns:xsi attribute on the root <AttributeFilterPolicyGroup> element.
- Add the following at the end of the whitespace delimited list of values for the xsi:schemaLocation attribute: http://www.gakunin.jp/ns/uapprove-jp/afp/mf http://www.gakunin.jp/schema/idp/gakunin-afp-mf-uapprovejp.xsd.

Define the Rule

This rule is defined by the <PermitValueRule xsi:type="uajpmf:AttributeInMetadata"> element with the following optional attribute:

onlyIfRequired	Boolean flag indicated that only those attributes which are marked as required should be released, those marked as optional will not be. Default value: **true**.
matchIfMetadataSilent	Boolean flag indicated that is marked as optional when the metadata has no <AttributeConsumingService> element. Default value: **false**.
onlyIfChecked	Boolean flag indicated that only those attributes which are marked as optional and user has permitted should be released. Default value: **false**. When set to false, its behavior is as same as AttributeInMetadata of Shibboleth IdP 3.2.0 or later.

How to write Permit Value Rule using the AttributeInMetadata Match Function:

```
<PermitValueRule xsi:type="uajpmf:AttributeInMetadata" onlyIfRequired="false" onlyIfChecked="true">
```

Attributes marked as optional will be displayed with checkbox. It is released when checkbox is checked only.

Example Permit Value Rule using the AttributeInMetadata Match Function:

```
<!--    ============================================================================
case 1: rule which compares metadata definitions with attributes mail,
     eduPersonPrincipalName, eduPersonAffiliation.

     Metadata which is marked as required, Everything is required information
     and always released.

     Metadate which is marked as optional:
     * mail attribute is required information and always released.
     * eduPersonPrincipalName attribute is optional information. In attribute
       selection window, it is displayed with checkbox. If the user checked the
       checkbox, it is released.
     * eduPersonAffiliation attribute is not released.

     No attributes are released when SP has no <AttributeConsumingService>
     element in metadata.
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="basic:ANY" />

    <afp:AttributeRule attributeID="mail">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             onlyIfRequired="false" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             onlyIfRequired="false"
                             onlyIfChecked="true" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata" />
    </afp:AttributeRule>

</afp:AttributeFilterPolicy>

<!--    ============================================================================
case 2: Example rule to add rule to SP which has no <AttributeConsumingService>
     element in metadata.

     When SP has no <AttributeConsumingService> element:
     * mail attribute is required information and always released.
     * eduPersonPrincipalName attribute is optional information. In attribute
       selection window, it is displayed with checkbox. If the user checked the
       checkbox, it is released.
     * eduPersonAffiliation attribute is not released.

     When SP has <AttributeConsumingService> element, it is the same as case 1.
        ============================================================================ -->
<afp:AttributeFilterPolicy id="PolicyforSPwithoutAttributeConsumingService">
    <afp:PolicyRequirementRule xsi:type="basic:ANY" />

    <afp:AttributeRule attributeID="mail">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             matchIfMetadataSilent="true"
                             onlyIfRequired="false" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonPrincipalName">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata"
                             matchIfMetadataSilent="true"
                             onlyIfRequired="false"
                             onlyIfChecked="true" />
    </afp:AttributeRule>

    <afp:AttributeRule attributeID="eduPersonAffiliation">
        <afp:PermitValueRule xsi:type="uajpmf:AttributeInMetadata" />
    </afp:AttributeRule>

</afp:AttributeFilterPolicy>
```

3 Troubleshooting

3.1 Troubleshooting

- Check IDP_HOME/logs/idp-process.log for ERROR or WARN messages.
- Check Tomcat’s log files located at $CATALINA_BASE$/logs for error messages.

3.2 Detailed logging

Enabling DEBUG or TRACE log level for the uApprove JP in IDP_HOME/conf/idp.properties:

IDP_HOME/conf/idp.properties
idp.loglevel.uApproveJP = DEBUG

A Notification of the using purpose of attributes on SP

A SP administrator can notify users of the using purpose (ex, use as initial value of user's profile) of attributes on the uApporve JP when they add the using purpose of attributes to their SP metadata.

A.1 Configuration

The notification of the using purpose of attributes on SP can be used to add uajpmd:description to <RequestedAttribute> element, or add <uajpmd:RequestedAttributeExtension> element in <Extensions> element in <SPSSODescriptor> element.

<uajpmd:RequestedAttributeExtension> element can describe by multiple languages. If both are set to one attribute, <uajpmd:RequestedAttributeExtension> element takes precedence.

uajpmd:description

This attribute is defined by the <ResquestedAttribute> element:

uajpmd:description	String indicated of the using purpose of attributes on SP

Example the <RequestedAttribute> element with uajpmd:descrption:

```
      <md:RequestedAttribute FriendlyName="mail"
               Name="urn:oid:0.9.2342.19200300.100.1.3"
               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
               uajpmd:description="The mail attribute is used as the initial value of the mail address field of the registration
form."/>
```

<uajpmd:RequestedAttributeExtension>

This element defines with the following attributes and one and more <uajpmd:Description> elements:

uajpmd:FriendlyName	The value of FriendlyName of the <RequestedAttribute> element to associate <uajpmd:RequestedAttributeExtension> element.

The <uajpmd:Description> element describes the using purpose of attributes on SP and defines with the following attributes:

xml:lang	The language used in the using purpose of attributes on SP

Example the <uajpmd:RequestedAttributeExtension>:

```
<md:EntitiesDescriptor Name="uapprovejp-dev-metadata.xml"
                    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
                    xmlns:ds="http://www.w3.org/2000/09/xmldsig#"
                    xmlns:shibmd="urn:mace:shibboleth:metadata:1.0"
                    xmlns:uajpmd="http://www.gakunin.jp/ns/uapprove-jp/metadata"
                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  ...
  <md:EntityDescriptor entityID="...">
    <md:SPSSODescriptor>
      ...

      <md:Extensions>
        ...
        <uajpmd:RequestedAttributeExtension FriendlyName="mail">
          <uajpmd:Description xml:lang="en">The mail attribute is used as the initial value of the mail address field of the
registration form.</uajpmd:Description>
          <uajpmd:Description xml:lang="ja">mail 属性を登録ページのメールアドレス欄の初期値として使用します</uajpmd:Description>
        </uajpmd:RequestedAttributeExtension>
        ...
      </md:Extensions>
      ...

      <md:AttributeConsumingService index="1">
             <md:ServiceName xml:lang="en">Sample Service</md:ServiceName>

        <md:RequestedAttribute FriendlyName="eduPersonPrincipalName"
                               Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6"
                               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
                               isRequired="true"/>
        <md:RequestedAttribute FriendlyName="mail"
                               Name="urn:oid:0.9.2342.19200300.100.1.3"
                               NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"/>
      </md:AttributeConsumingService>
      ...
  </md:SPSSODescriptor>
```