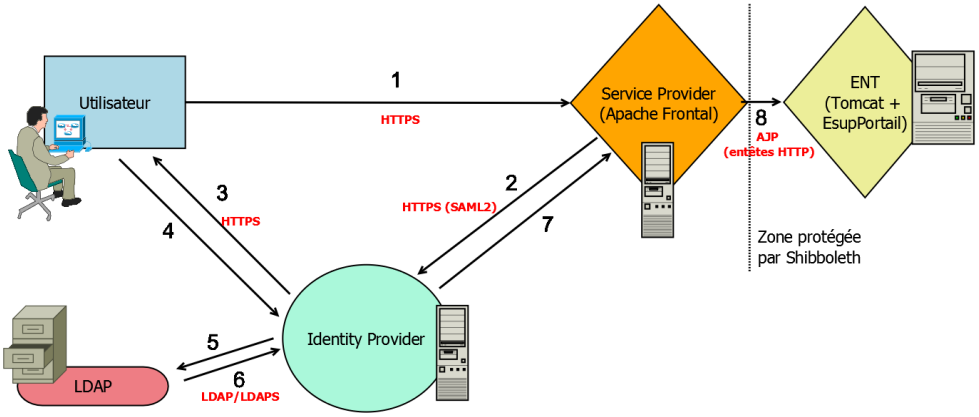
Shibboleth (esup 4)

Principe de Shibboleth



Légende du schéma

1 - L'utilisateur veut accéder à l'ENT, protéger par Shibboleth. La requête est interceptée par le SP.
2 - Le SP interroge l'IdP pour savoir si l'utilisateur à le droit d'accéder à la page.
3 - L'IdP demande à l'utilisateur ses informations de connexion.
4 - L'utilisateur indique son login et son mot de passe.
5 - L'IdP interroge un annuaire LDAP avec les informations fournies par l'utilisateur.
6 - Le LDAP confirme ou non que l'utilisateur existe bien et transmet à l'IdP ses informations.
7 - L'IdP retransmet les attributs de l'utilisateur authentifié au SP
8 - En fonction des attributs de l'utilisateurs, le SP va permettre l'accès à la page sécurisée : l'ENT.

Prérequis

Tomcat doit avoir été configuré et être branché sur un serveur Apache frontal. Bien vérifier également que les serveurs Tomcat présentent le paramètre suivant au niveau des différents connecteurs :

server.xml

```
tomcatAuthentication="false"
```

Il convient également d'avoir à disposition un IdP et un SP fonctionnels, et que le mapping des attributs LDAP ait été correctement réalisé.

Une version de Shibboleth 2.x est obligatoire, pour assurer le support SAML2.

La documentation Renater fournit des tutoriels détaillés sur l'installation et la configuration d'un IdP et d'un SP Shibboleth.

Configuration Apache - mod_shib

La mise en place du SP Shibboleth sur le serveur Apache va se faire via l'activation du module mod_shib, fourni avec le SP.

Il conviendra donc d'ajouter le chargement du module à la configuration du serveur httpd :

```
LoadModule mod_shib $SP_HOME/lib/shibboleth/mod_shib_xx.so
```

Il existe plusieurs versions fournies du mod_shib, dépendantes de la version du serveur Apache utilisé. Il conviendra donc de charger le module correspondant : mod_shib_13 (Apache httpd 1.3), mod_shib_20 (Apache httpd 2.0), mod_shib_22 (Apache httpd 2.2, version testée sous Windows), mod_shib_24 (Apache httpd 2.4, version testée sous Unix).

Une fois le module chargé, on va pouvoir le configurer. On va dans un premier temps protéger le handler Shibboleth qui permettra, entre autre, d'obtenir des informations sur le SP (Session en cours, statut, metadata...) :

```
...
UseCanonicalName On
ServerName localhost

<Location /Shibboleth.sso>
    SetHandler shib
</Location>
...
```

On peut également ajouter un script qui permettra de tester l'identification par Shibboleth, et qui affichera les informations transmises dans le header :

```
...
ScriptAlias /secure $APACHE_HOME/cgi-bin/printenv.pl
<Location /secure>
   AuthType shibboleth
   ShibRequestSetting requireSession 1
   ShibRequestSetting applicationId default
   require valid-user
</Location>
...
```

L'utilisation de ce script est expliquée dans la partie Tests de fonctionnement ci-dessous.

Enfin, il reste à protéger l'URL de login du portail pour forcer une authentification via Shibboleth quand un utilisateur accèdera à cette URL :

```
...
<Location /uPortal/Login >
    AuthType shibboleth
    ShibRequestSetting requireSession 1
    ShibRequestSetting applicationId default
    require valid-user
</Location>
...
```

Configuration du SP

L'identification se fait via des attributs LDAP remontés par l'IdP. Il faut donc s'assurer que le SP est bien configuré pour transmettre dans le remote user le bon attribut. Par défaut, l'attribut servant à l'authentification est l'uid (défini par le paramètre *environment.build.ldap.uidAttr* du filtre esup.properties). Il convient donc de modifier la configuration du SP (shibboleth2.xml) pour l'ajouter :

etc/shibboleth/shibboleth2.xml

```
<ApplicationDefaults entityID="<entityID du SP>"
                        REMOTE_USER="uid eppn persistent-id targeted-id">
```

Le tutoriel de la féderation Renater détaille ici certains éléments de configuration du SP, notamment les paramètres du nœud *ApplicationDefaults.* Parmi ceux-ci, le paramètre *SSO* (ancien *SessionInitiator)* va notamment permettre de définir la façon dont Shibboleth gère les demandes de session.

Par exemple, en définissant la balise de la façon suivante :

etc/shibboleth2.xml

```
<SSO discoveryProtocol="SAMLDS" discoveryURL="https://services-federation.renater.fr/test/wayf">
    SAML2 SAML1
</SSO>
```

Ici, le SP va utiliser un procole SAMLDS (Discovery Service), qui se fera via le WAYF précisé, pour l'identification. Le protocole SAML2 sera utilisé en priorité, et s'il n'est pas supporté, c'est SAML1 qui le sera.
Plus d'informations sur les valeurs possibles pour ce paramètre sont détaillées sur le wiki Shibboleth.

Il est également possible de filtrer dans ce fichier de configuration les IdP qui auront (whitelist) ou n'auront pas (blacklist) accès à l'application. La fédération Renater décrit ici la configuration à adopter.

Configuration uPortal

Une fois le serveur frontal configuré avec le SP, il faut configurer uPortal pour l'authentification via Shibboleth. L'authentification se fait par REMOTE_USER : l'utilisateur est identifié par Shibboleth, et est transmis à uPortal. Il va donc falloir faire les changements nécessaires.

Classes d'authentification

Les premières modifications sont à faire dans *uportal-war/src/main/resources/properties/security.properties*. Au début du fichier sont placées les classes d'authentification :

security.properties

```
...
## This is the factory that supplies the concrete authentication class
root=org.jasig.portal.security.provider.UnionSecurityContextFactory
root.cas=org.jasig.portal.security.provider.cas.CasAssertionSecurityContextFactory
#root.cas=org.jasig.cas3.extensions.clearpass.integration.uportal.
PasswordCachingCasAssertionSecurityContextFactory
root.simple=org.jasig.portal.security.provider.SimpleSecurityContextFactory
...
```

On va y ajouter la ligne suivante, qui va rajouter une classe d'identification d'un utilisateur remote :

security.properties

```
root.remote=org.jasig.portal.security.provider.RemoteUserSecurityContextFactory
```


Si l'authentification se fait par Shibboleth uniquement, il est préférable de commenter toutes les lignes de cette section, à l'exception de celle ajoutée et de la première ligne : *root=org.jasig.portal.security.provider.UnionSecurityContextFactory*.

Si plusieurs méthodes sont utilisées, uPortal essayera les différentes méthodes jusqu'à ce qu'une valide l'authentification.

Contexte utilisateur

Il va ensuite falloir préciser dans le contexte utilisateur qu'il s'agit d'un remote user. Cela se fait en éditant le fichier *uportal-war/src/main/resources /properties/contexts/userContext.xml* .
Il suffira d'éditer ce bean :

userContext.xml

```
...
<bean id="personManager" class="org.jasig.portal.security.provider.SimplePersonManager" />
...
```

Et de le remplacer par celui-ci :

userContext.xml

```
...
<bean id="personManager" class="org.jasig.portal.security.provider.RemoteUserPersonManager" />
...
```

Attributs utilisateur

Il va ensuite falloir modifier deux beans existants dans le fichier *uportal-war/src/main/resources/properties/contexts/personDirectoryContext.xml* pour configurer les attributs utilisés : requestAttributeSourceFilter et requestAdditionalDescriptors.
Ils doivent être modifiés de la façon suivante :

personDirectoryContext.xml

```
...
<!--
 | Servlet filter that creates an attribute for the serverName
 +-->
<bean id="requestAttributeSourceFilter" class="org.jasig.services.persondir.support.web.
RequestAttributeSourceFilter">
    <property name="additionalDescriptors" ref="requestAdditionalDescriptors" />
    <property name="usernameAttribute" value="remoteUser" />
    <property name="remoteUserAttribute" value="remoteUser" />
    <property name="serverNameAttribute" value="serverName" />
    <property name="processingPosition" value="BOTH" />
    <property name="headerAttributeMapping">
        <map>
            <!-- MODIFY THESE MAPPINGS TO EXPOSE HEADERS FROM SHIB AS USER ATTRIBUTES -->
            <entry key="cn">
                <list>
                    <value>cn</value>
                    <value>displayName</value>
                </list>
            </entry>
            <entry key="givenName" value="givenName" />
        </map>
    </property>
</bean>

<!--
 | Session-scoped descriptors object. One of these will exist for each user in their session. It will store the
 | attributes from the reques set by the requestAttributeSourceFilter
 +-->
<bean id="requestAdditionalDescriptors" class="org.jasig.services.persondir.support.
MediatingAdditionalDescriptors">
    <property name="delegateDescriptors">
        <list>
            <bean class="org.jasig.services.persondir.support.AdditionalDescriptors" scope="globalSession">
                <aop:scoped-proxy />
            </bean>
            <bean class="org.jasig.services.persondir.support.AdditionalDescriptors" scope="request">
                <aop:scoped-proxy />
            </bean>
        </list>
    </property>
</bean>
...
```

Lien de connexion

Si l'authentification est réalisée par Shibboleth uniquement, alors il convient de supprimer le lien de connexion CAS du bandeau. Pour ce faire,il convient de modifier le fichier *uportal-war/src/main/resources/layout/theme/universality/components.xsl* et de supprimer le bloc suivant :

components.xsl

```
...
<div id="portalCASLogin" class="fl-widget-content">

<a id="portalCASLoginLink" class="button" href="{$EXTERNAL_LOGIN_URL}" title=
"{upMsg:getMessage('sign.in.via.cas', $USER_LANG)}">
<span><xsl:value-of select="upMsg:getMessage('sign.in', $USER_LANG)"/><!-- <span
 class="via-cas"><xsl:value-of
select="upMsg:getMessage('with.cas',$USER_LANG)"/></span>--></span>
</a>


<p>
<xsl:value-of select="upMsg:getMessage('new.user.question', $USER_LANG)"/> 
<a id="portalCASLoginNewLink" href="{$CAS_NEW_USER_URL}" title="{upMsg:getMessage('create.new.portal.account',
$USER_LANG)}">
<xsl:value-of select="upMsg:getMessage('new.user', $USER_LANG)"/>
</a>.
</p>


</div>
...
```

Pour connecter un utilisateur, il suffira de le rediriger vers le lien */uPortal/Login* défini pour le mod_shib, que ce soit via un lien, un bouton, une image, etc...

Tests de fonctionnement

Ces tests vont permettre de vérifier le bon fonctionnement et le bon paramétrage de l'IdP et du SP Shibboleth avec, successivement, la récupération d'attributs LDAP par l'IdP et la transmission de ces attributs de l'IdP au SP .

> Ces étapes peuvent être suivies aussi bien avant l'installation qu'à tout moment pour déterminer l'origine d'un problème qui serait lié à l'authentification Shibboleth.

IdP

Avec l'IdP est fourni un utilitaire (aacli) qui va permettre de tester la récupération des attributs définis dans les fichiers de configuration de l'IdP *attribute-resolver* et *attribute-filter*. Il se trouve dans le répertoire /bin de l'IdP, et est décliné en deux versions : aacli.bat (Windows) et aacli.sh (Unix).

Deux paramètres sont nécessaires pour lancer l'utilitaire :
--*configDir* qui doit pointer vers le dossier contenant les fichiers de configuration de l'IdP (/conf en général)
--*principal* qui est l'identifiant utilisé pour le test.

D'autres paramètres de test peuvent être définis et sont explicités avec l'utilisation du paramètre --*help*.

Si un utilisateur est défini par un uid etudiant1 dans l'annuaire LDAP, et que l'IdP est configuré pour remonter les attributs uid et displayName, la commande :

```
(Unix)                $aacli.sh --configDir=<Path>conf --principal=etudiant1
(Windows)        aacli.bat --configDir=<Path>conf --principal=etudiant1
```

devrait produire une sortie semblable à cela :

```
<?xml version="1.0" encoding="UTF-8"?><saml2:AttributeStatement xmlns:saml2="urn:oasis:names:tc:SAML:2.0:
assertion">
   <saml2:Attribute FriendlyName="uid" Name="urn:oid:0.9.2342.19200300.100.1.1" NameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:type="xs:string">etudiant1</saml2:AttributeValue>
   </saml2:Attribute>
   <saml2:Attribute FriendlyName="displayName" Name="urn:oid:2.16.840.1.113730.3.1.241" NameFormat="urn:oasis:
names:tc:SAML:2.0:attrname-format:uri">
      <saml2:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:type="xs:string">Prenom Nom</saml2:AttributeValue>
   </saml2:Attribute>
</saml2:AttributeStatement>
```

Cela permet de vérifier que l'IdP peut bien se connecter au serveur LDAP et remonter des attributs.

SP

Pour vérifier que le SP est bien en état de fonctionnement, depuis la machine hébergeant le serveur Apache, on peut se rendre à l'url suivante : http://localhost/Shibboleth.sso/Status . Cette page devrait afficher des informations de métadonnées sur le SP, ce qui confirme son fonctionnement.

Pour vérifier que l'IdP transmet bien les éléments qu'il récupère au SP, on va utiliser la page de test mise en place lors de la configuration d'Apache : /secure.
Étant protégée par Shibboleth, elle va requérir une authentification, et on pourra vérifier les attributs transmis lors de cette identification.

Pour cela, il suffit d'accéder à http://localhost/secure pour afficher les attributs utilisateurs shibboleth résultant de l'exécution du script printenv.pl.

Attention : pour l'execution de ce script, Perl est nécessaire, et son chemin doit être précisé dans le script printenv.pl

Fourni normalement (à vérifier) avec la librairie sp shibboleth, voici le contenu de script :

```
 #!/usr/bin/perl
print "Content-type: text/plain\n\n";
print "Variables d'environnement positionnées par le SP shibboleth :\n\n";
foreach my $key (keys %ENV) {
    if ($key eq 'REMOTE_USER' || $key =~ /^Shib_/ || $key =~ /^[a-z]/) {
        printf "$key=$ENV{$key}\n";
    }
}
```

Une fois identifié, il est également possible d'accéder à la page de Session du SP qui affiche plus spécifiquement les attributs qui lui ont été transmis par l'IdP pour la session en cours, via l'URL http://localhost/Shibboleth.sso/Session .

Si ces attributs correspondent bien à ceux attendus, alors la communication entre IdP et SP est correcte.

Références

https://wiki.jasig.org/display/UPM40/Shibboleth
https://services.renater.fr/federation/docs/shibboleth
https://spaces.internet2.edu/display/ShibuPortal/Home