Installation et CONFIGURATION de Shibboleth IdP

Normalement, quelqu'un qui lit ce document n'a pas besoin d'explication sur ce qu'est Shibboleth, car il le sait déjà.
Shibboleth sert à faire ce qu'on appelle de la fédération d'identités, permettant de connecter des utilisateurs à des applications à l'intérieur d'organisations, ou entre plusieurs organisations indépendantes.

Normalement, quelqu'un qui lit ce document n'a pas besoin de savoir ce qu'est la fédération d'identités, parce qu'il en est un expert. Dans ce domaine, on rencontrera des termes tels que SSO, SAML, Ticket, IdP (Identity Provider), SP (Service Provider), etc ; mais vous êtes experts en tout ceci, si vous lisez ce document.
Supposons que dans mon entreprise, j'ai plusieurs applications (supposition « Toto-logique » puisqu'on a toujours 10000+ applications inutiles et inefficaces remplies de bugs dans les entreprises). Chaque application possède sa propre BD d'utilisateurs/mots de passe. En plus, on demande souvent aux utilisateurs de ne pas avoir les mêmes mots de passe pour les différentes applications, et de les changer tous les matins. Chaque système a son propre mécanisme de gestion et d'authentification des utilisateurs. En implémentant une fédération d'identité, on pourrait s'affranchir de ces multitudes de comptes par système. Tout sera géré au niveau du fournisseur d'identités, et les applications (fournisseur de services) vont «rediriger » les requêtes d'accès vers le IdP. Si on a des applications accessibles dans le cloud (SaaS) ou des partenaires qui nous donnent accès à leurs ressources, les accès à ces applications pourraient aussi se faire via notre IdP, et c'est ce que nous implémenterons avec notre Shi-bboleth. Il y a quelques mois, j'avais fait des schémas qui expliquent un chouya ADFS et GIA, c'est ici : http://www.yerbynet.com/Cours/ADFS.pdf

Une fois qu'on a compris ce que shibboleth est (ou pas), on va passer à son installation (mais si, ou pas).
C'est une installation classique, d'ailleurs assez simple, mais la CONFIGURATION (hwo hwo hwo hwoa – non, pas le ho ho ho du père Noël, mais plutôt celui du film d'horreur): c'est une usine à gaz, un labyrinthe de fichiers de configuration. C'est comme un lion qui essaye de chasser du thon au large des Seychelles. C'est pire que Nagios, Exim, MPLS, VPWS, TC, LLTM, et Radius (réunis). C'est presque pareil que de redistribuer 90k de préfixes dans des VRF qui sont configurés sur des équipements qui n'en supportent que 5k. Comme le dirait le chef du parti, c'est du « fourrage ».
Il faut se préparer à passer du temps avec lui avant qu'il ne vous accepte.

Il ne me reste plus qu'à vous dire : merci, bonjour ! (moi étant compris dans le vous).

I - Installation

Nous installons Shibboleth3.3.1 avec Tomcat8 et Java8 sur du CentOS7 qui roule Apache2.4.

0- Installation de Tomcat :

http://www.yerbynet.com/Cours/TomcatCentos7.html

0'- Installation de MySQL (optionnel) :

Installer mariadb (***yum install mariadb-server mariadb mysql-connector-java***) et sécuriser le un minimum avec la commande ***mysql_secure_installation***.
Puis, ajouter la ligne ci-dessous dans le fichier « /etc/my.cnf » :

```
bind-address=127.0.0.1
```

Vous pouvez créer le fichier « /root/.my.cnf » et ajouter les credentials du root de mysql.
Partie totalement optionnelle, et qui permet juste de ne pas avoir à entrer le mot de passe root à chaque fois qu'on veut faire des requêtes MySQL.

```
[client]
user=root
password=motdepasse_mysql_de_root
```

Relance de MySQL :

0''- Installation de NTP (optionnel) :

Installer ntp (***yum install ntp***) et faites le écouter seulement sur sa loopback en ajoutant dans le fichier « /etc/ntp.conf » les lignes suivantes :

```
interface ignore wildcard
interface listen 127.0.0.1
interface listen::1
```

1- Préparation de l'environnement :

Veuillez noter que mon serveur shibboleth s'appelle « shib-idp » et que mon domaine est « yerbynet.com ». La plupart des exemples et des config sont donc basés sur shib-idp.yerbynet.com.

[root@shib-idp tomcat]# ***yum install java-1.8.0-openjdk-devel*** (Ai-je vraiment installé ce paquet?)

[root@shib-idp ~]# ***mkdir /root/inst-shib***
[root@shib-idp ~]# ***cd /root/inst-shib***

[root@shib-idp ~]# ***vi /etc/profile.d/shib.sh***

```
IDP_VERSION="3.3.1"
SHIB_HOME=/opt/shibboleth-idp
SHIB_INST_HOME=/root/inst-shib/shibboleth-identity-provider-$IDP_VERSION

IDP_HOME=/opt/shibboleth-idp
```

JAVA_HOME=/usr/lib/jvm/java
TC_HOME=/opt/tomcat

export SHIB_HOME IDP_HOME JAVA_HOME SHIB_INST_HOME IDP_VERSION TC_HOME

[root@shib-idp inst-shib]# ***. /etc/profile.d/shib.sh***

[root@shib-idp inst-shib]# ***wget https://shibboleth.net/downloads/identity-provider/latest/shibboleth-identity-provider-${IDP_VERSION}.tar.gz***

2- Installation par les sources :

[root@shib-idp inst-shib]# ***tar zxvf shibboleth-identity-provider-${IDP_VERSION}.tar.gz***

[root@shib-idp inst-shib]# ***cd $SHIB_INST_HOME***

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***pwd***
/root/inst-shib/shibboleth-identity-provider-3.3.1

~~{// Passez par ici si et seulement si vous avez l'erreur avec JAVA_HOME~~

~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***./bin/install.sh***~~
~~Error: JAVA_HOME is not defined correctly.~~
~~We cannot execute java~~
~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***echo $JAVA_HOME***~~

~~[root@osboxes shibboleth-identity-provider-3.3.1]#~~

~~[root@osboxes shibboleth-identity-provider-3.3.1]# whereis java~~
~~java: /usr/bin/java /usr/lib/java /etc/java /usr/share/java /usr/share/man/man1/java.1.gz~~
~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***ls -l /usr/bin//java***~~
~~lrwxrwxrwx. 1 root root 22 1 juin 09:56 /usr/bin//java -> /etc/alternatives/java~~
~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***ls -l /etc/alternatives/java***~~
~~lrwxrwxrwx. 1 root root 73 1 juin 09:56 /etc/alternatives/java -> /usr/lib/jvm/java-1.8.0-openjdk-1.8.0.131-3.b12.el7_3.x86_64/jre/bin/java~~
~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***ls -l /usr/lib/jvm/java***~~
~~lrwxrwxrwx. 1 root root 26 8 juin 12:15 /usr/lib/jvm/java -> /etc/alternatives/java_sdk~~

~~[root@osboxes shibboleth-identity-provider-3.3.1]# ***vi /root/.bash_profile***~~

~~# Get the aliases and functions~~
~~if [-f ~/.bashrc]; then~~
~~. ~/.bashrc~~
~~fi~~

~~# User specific environment and startup programs~~

~~PATH=$PATH:$HOME/bin~~

~~export PATH~~
~~**export JAVA_HOME="/usr/lib/jvm/java"**~~
~~[root@osboxes shibboleth-identity-provider-3.3.1]#~~

~~#}~~

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***sh ./bin/install.sh***

Source (Distribution) Directory (press <enter> to accept default): [/root/inst-shib/shibboleth-identity-provider-3.3.1]

*Installation Directory: [**/opt/shibboleth-idp**]*

*Hostname: [**shib-idp.yerbynet.com**]*

*SAML EntityID: [**https://shib-idp.yerbynet.com/idp/shibboleth**]*

*Attribute Scope: [**yerbynet.com**]*

Backchannel PKCS12 Password:
Re-enter password:
Cookie Encryption Key Password:
Re-enter password:
Warning: /opt/shibboleth-idp/bin does not exist.
Warning: /opt/shibboleth-idp/dist does not exist.
Warning: /opt/shibboleth-idp/doc does not exist.
Warning: /opt/shibboleth-idp/system does not exist.
Warning: /opt/shibboleth-idp/webapp does not exist.
Generating Signing Key, CN = shib-idp.yerbynet.com URI = https://shib-idp.yerbynet.com/idp/shibboleth ...
...done
Creating Encryption Key, CN = shib-idp.yerbynet.com URI = https://shib-idp.yerbynet.com/idp/shibboleth ...
...done
Creating Backchannel keystore, CN = shib-idp.yerbynet.com URI = https://shib-idp.yerbynet.com/idp/shibboleth ...
...done
Creating cookie encryption key files...
...done
Rebuilding /opt/shibboleth-idp/war/idp.war ...
...done

BUILD SUCCESSFUL
Total time: 1 minute 59 seconds

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***ls -l /opt/shibboleth-idp/war/idp.war***
-rw-r--r--. 1 root root 38802396 13 jun 11:55 /opt/shibboleth-idp/war/idp.war

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***openssl pkcs12 -in $IDP_HOME/credentials/idp-backchannel.p12 -out $IDP_HOME/credentials/idp-backchannel.key -nocerts -nodes***
Enter Import Password:
MAC verified OK

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***cd /opt/shibboleth-idp/***
[root@shib-idp shibboleth-idp]# ***ls -l credentials/***

[root@shib-idp shibboleth-idp]# ***vi /opt/tomcat/conf/Catalina/localhost/idp.xml***

```
<Context docBase="/opt/shibboleth-idp/war/idp.war"
        privileged="true"
        antiResourceLocking="false"
        antiJARLocking="false"
        swallowOutput="true" >

    <!-- Work around lack of Max-Age support in IE/Edge -->
    <CookieProcessor alwaysAddExpires="true" />

</Context>
```

[root@shib-idp shibboleth-identity-provider-3.3.1]# ***cd $IDP_HOME***
[root@shib-idp shibboleth-idp]# ***cd edit-webapp/WEB-INF/lib/***
[root@shib-idp lib]# ***pwd***
/opt/shibboleth-idp/edit-webapp/WEB-INF/lib

[root@shib-idp lib]# ***wget https://build.shibboleth.net/nexus/service/local/repositories/thirdparty/content/javax/servlet/jstl/1.2/jstl-1.2.jar***

[root@shib-idp lib]# ***/opt/shibboleth-idp/bin/build.sh -Didp.target.dir=/opt/shibboleth-idp***

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

BUILD SUCCESSFUL
Total time: 2 seconds

[root@shib-idp shibboleth-idp]# ***wget -O /opt/tomcat/lib/javax.servlet.jsp.jstl-api-1.2.1.jar 'http://search.maven.org/remotecontent?filepath=javax/servlet/jsp/jstl/javax.servlet.jsp.jstl-api/1.2.1/javax.servlet.jsp.jstl-api-1.2.1.jar'***
[root@shib-idp shibboleth-idp]# ***wget -O /opt/tomcat/lib/javax.servlet.jsp.jstl-1.2.1.jar http://search.maven.org/remotecontent?filepath=org/glassfish/web/javax.servlet.jsp.jstl/1.2.1/javax.servlet.jsp.jstl-1.2.1.jar***

[root@shib-idp shibboleth-idp]# ***chown -Rv tomcat:tomcat /opt/shibboleth-idp***
[root@shib-idp shibboleth-idp]# ***chmod 600 /opt/shibboleth-idp/conf/ldap.properties /opt/shibboleth-idp/conf/attribute-resolver.xml***

3- Ajustements de Tomcat, Apache et Firewalld :

Pour une machine de 4G de RAM, je peux faire ceci :

[root@shib-idp tomcat]# ***cd $TC_HOME***
[root@shib-idp tomcat]# ***vi bin/setenv.sh***

```
export CATALINA_OPTS="$CATALINA_OPTS -Xmx3072M"
export CATALINA_OPTS="$CATALINA_OPTS -Xms3072M"
export CATALINA_OPTS="$CATALINA_OPTS -Didp.home=/opt/shibboleth-idp"
export CATALINA_OPTS="$CATALINA_OPTS -XX:+UseG1GC"
```

[root@shib-idp tomcat]# ***vim conf/server.xml***

```
  <!-- Commenter ces 3 lignes
  <Connector port="8080" protocol="HTTP/1.1"
             connectionTimeout="20000"
             redirectPort="8443" />
 -->

  <!-- Commenter aussi ca :
  <Connector port="8009" protocol="AJP/1.3" redirectPort="8443" />
  -->

  <!-- Et ajouter ca :         -->
  <Connector port="8009" address="127.0.0.1"
    enableLookups="false" redirectPort="443"
    protocol="AJP/1.3" maxPostSize="100000"
    tomcatAuthentication="false" />
```

[root@shib-idp tomcat]# ***ln -s /usr/share/java/mysql-connector-java.jar /opt//tomcat/lib/***

[root@shib-idp tomcat]# ***systemctl restart tomcat***

[root@shib-idp tomcat]# ***netstat -tlnp***

```
tcp6       0      0 127.0.0.1:8009          :::*                    LISTEN      13615/java
tcp6       0      0 127.0.0.1:8005          :::*                    LISTEN      13615/java
tcp        0      0 0.0.0.0:3306            0.0.0.0:*               LISTEN      8452/mysqld
```

[root@shib-idp tomcat]# ***yum install httpd mod_ssl openldap-clients***
[root@shib-idp tomcat]# ***systemctl enable httpd***
[root@shib-idp tomcat]# ***systemctl start httpd***
[root@shib-idp tomcat]# ***netstat -tlnp***

[root@shib-idp tomcat]# ***firewall-cmd --add-service=http***
success
[root@shib-idp tomcat]#***firewall-cmd --add-service=https***
success
[root@shib-idp tomcat]# ***firewall-cmd --permanent --add-service=http***
success
[root@shib-idp tomcat]# ***firewall-cmd --permanent --add-service=https***
success
[root@shib-idp tomcat]# ***firewall-cmd --add-port=8443/tcp***
[root@shib-idp tomcat]# ***firewall-cmd --permanent --add-port=8443/tcp***

[root@shib-idp tomcat]# ***cd /etc/httpd/conf.d/***

[root@shib-idp conf.d]# ***vi ssl.conf***

```
<VirtualHost _default_:443>

      #DocumentRoot "/var/www/html"

      ServerName shib-idp.yerbynet.com:443

      ErrorLog logs/ssl_error_log
      TransferLog logs/ssl_transfer_log
          CustomLog logs/ssl_request_log \
        "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
      LogLevel warn

      SSLEngine on
      SSLProtocol all -SSLv2 -SSLv3
      SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA:!RC4:!LOW
      SSLHonorCipherOrder on
      SSLCertificateFile /etc/pki/tls/certs/localhost.crt
      SSLCertificateKeyFile /etc/pki/tls/private/localhost.key

      ProxyRequests Off

      <Location /idp>
            Require all granted
            SSLOptions      +StdEnvVars +ExportCertData
            SSLVerifyClient optional_no_ca
            SSLVerifyDepth  10
      </Location>

      <Proxy ajp://localhost:8009/idp/*>
            Require all granted
      </Proxy>

      ProxyPass /idp ajp://localhost:8009/idp retry=5
```

```
    <Files ~ "\.(cgi|shtml|phtml|php3?)$">
        SSLOptions +StdEnvVars
    </Files>

    <Location /idp/profile/SAML2/SOAP/ECP>
        AuthType Basic
        AuthName "YerbyNET - Shibboleth Identity Provider - ECP profile"
        AuthBasicProvider ldap
        AuthLDAPURL ldap://ad.yerbynet.com/cn=Users,dc=priv,dc=yerbynet,dc=com?
sAMAccountName
        AuthLDAPBindDN "cn=roger_user,cn=Users,dc=priv,dc=yerbynet,dc=com"
        AuthLDAPBindPassword "roger_pass"
        Require valid-user
        SSLRequireSSL
    </Location>

   BrowserMatch "MSIE [2-5]" \
      nokeepalive ssl-unclean-shutdown \
      downgrade-1.0 force-response-1.0

</VirtualHost>
```

[root@shib-idp conf.d]# ***vi default.conf***

```
<VirtualHost _default_:80>

    ServerName shib-idp.yerbynet.com:80

    ErrorLog logs/default_error_log
    TransferLog logs/default_transfer_log
    CustomLog logs/default_request_log \
      "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"
    LogLevel warn

    Redirect Permanent / https://shib-idp.yerbynet.com/

</VirtualHost>
```

[root@shib-idp conf.d]# ***vi ssl8443.conf***

```
Listen 8443 https
<VirtualHost _default_:8443>

    ServerName shibboleth.yerbynet.com:8443

    ErrorLog logs/ssl8843_error_log
    TransferLog logs/ssl8843_transfer_log
    LogLevel warn
```

```
    SSLEngine on
    SSLProtocol all -SSLv2 -SSLv3
    SSLCipherSuite HIGH:MEDIUM:!aNULL:!MD5:!SEED:!IDEA:!RC4:!LOW
    SSLHonorCipherOrder on
    SSLCertificateFile /opt/shibboleth-idp/credentials/idp-backchannel.crt
    SSLCertificateKeyFile /opt/shibboleth-idp/credentials/idp-backchannel.key
    SSLVerifyClient         optional_no_ca
    SSLVerifyDepth          10

    ProxyRequests Off

    <Proxy ajp://localhost:8009>
          Allow from all
    </Proxy>

    ProxyPass /idp ajp://localhost:8009/idp retry=5

    <Files ~ "\.(cgi|shtml|phtml|php3?)$">
       SSLOptions +StdEnvVars
    </Files>
    <Directory "/var/www/cgi-bin">
       SSLOptions +StdEnvVars
    </Directory>

    SetEnvIf User-Agent ".*MSIE.*" \
           nokeepalive ssl-unclean-shutdown \
           downgrade-1.0 force-response-1.0
    CustomLog logs/ssl8443_request_log \
           "%t %h %{SSL_PROTOCOL}x %{SSL_CIPHER}x \"%r\" %b"

    SSLOptions      -StdEnvVars +ExportCertData

</VirtualHost>
```

[root@shib-idp conf.d]# ***setsebool -P httpd_can_network_connect on***
[root@shib-idp conf.d]# ***httpd -t***
[root@shib-idp conf.d]# ***systemctl restart httpd***
[root@shib-idp conf.d]# ***netstat -tlnp***

```
tcp        0      0 127.0.0.1:3306          0.0.0.0:*               LISTEN      19112/mysqld
tcp6       0      0 127.0.0.1:8009          :::*                    LISTEN      13615/java
tcp6       0      0 127.0.0.1:8005          :::*                    LISTEN      13615/java
tcp6       0      0 :::80                   :::*                    LISTEN      13766/httpd
tcp6       0      0 :::8443                 :::*                    LISTEN      13766/httpd
tcp6       0      0 :::443                  :::*                    LISTEN      13766/httpd
```

La partie la plus simple est faite.

Allez prendre un café ou un thé avant de poursuivre, une petite bière pourrait aussi faire l'affaire, mais surtout pas un whisky.

Vous pouvez procéder à quelques petits tests avant de continuer. Accéder à cette URL :
- https://shib-idp.yerbynet.com/idp/shibboleth : donne un document xml contenant l'entityID de votre shibboleth et des certificats générés lors de l'installation (fichier « $IDP-HOME/metadata/idp-metadata.xml »)

Vous pouvez avoir besoin de prendre un petit breuvage avant de continuer ou même un bon petit dodo.

II – La bonne Grosse CONFIG de shibboleth

Voici des fichiers que j'ai eu à modifier dans le cadre de ma config de Shibboleth 3.3.1 (ou pas).

[root@shib-idp conf.d]# ***cd $IDP_HOME***

Liste des fichiers à modifier :
conf/idp.properties
conf/ldap.properties
conf/attribute-resolver.xml
conf/saml-nameid.properties
conf/saml-nameid.xml
conf/metadata-providers.xml
conf/relying-party.xml
conf/attribute-filter.xml
conf/global.xml
conf/access-control.xml
metadata/idp-metadata.xml
conf/authn/password-authn-config.xml
conf/c14n/subject-c14n.xml
conf/intercept/consent-intercept-config.xml
messages/messages.properties
edit-webapp/images/logo.png

La liste des fichiers n'est pas exhaustive, j'en oublie sûrement.

Avant de continuer, vous pourrez avoir besoin de jeter un coup d'œil pour avoir la liste de tous les fichiers susceptibles d'être modifiés, ici :
https://wiki.shibboleth.net/confluence/display/IDP30/ConfigurationFileSummary

Vous pourriez avoir besoin de passer en mode debug jusqu'à ce que ça marche.
Ça se fait dans le fichier « $IDP_HOME/conf/logback.xml ».

```
<variable name="idp.loglevel.idp" value="DEBUG" />
<variable name="idp.loglevel.ldap" value="DEBUG" />
<variable name="idp.loglevel.messages" value="DEBUG" />
```

```
<variable name="idp.loglevel.encryption" value="DEBUG" />
<variable name="idp.loglevel.opensaml" value="DEBUG" />
<variable name="idp.loglevel.props" value="DEBUG" />
```

Vous pourriez avoir besoin de suivre les logs avec ceci :
tail -f /opt/shibboleth-idp/logs/idp-process.log /opt/tomcat/logs/catalina.out

0- « conf/access-control.xml »

Avec ce shibboleth version 3, on a une petite interface admin qui nous permet quelques petites opérations de check de status, ou de reload de metadata ou de service. Encore faudrait-il y avoir accès.

Pour ce faire, il nous faut 2 petites choses faciles et rapides à faire.

i- Fichier « $IDP_HOME/conf/idp.properties » et vérifier que ces 3 lignes existent (normalement, c'est le cas par défaut) :

```
idp.status.accessPolicy= AccessByIPAddress
idp.resolvertest.accessPolicy= AccessByIPAddress
idp.reload.accessPolicy= AccessByIPAddress
```

ii- Fichier « $HOME_IDP/conf/access-control.xml »

```
    <entry key="AccessByIPAddress">
      <bean id="AccessByIPAddress" parent="shibboleth.IPRangeAccessControl"
        p:allowedRanges="#{ {'127.0.0.1/32', '::1/128', 'IPDuServeur', 'Net1', 'Net2'} }" />
    </entry>
```

Petit test avant d'aller de l'avant :
- URL : https://shib-idp.yerbynet.com/idp/status : donne le statut de votre shibboleth (version de Java, l'OS, version de shibboleth, last reload, …).
- Refaites aussi le test précédent svp.

0' – Fichier « $IDP_HOME/metadata/idp-metadata.xml »

Dans ce fichier, je pense que j'ai dé-commenté ou ajouté les 4 lignes commençant par « SingleLogoutService ».
Ça me donne quelque chose de ce genre :

```
    <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:1.0:bindings:SOAP-binding"
Location="https://shib-idp.yerbynet.com:8443/idp/profile/SAML1/SOAP/ArtifactResolution"
index="1"/>
    <ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://shib-idp.yerbynet.com:8443/idp/profile/SAML2/SOAP/ArtifactResolution"
index="2"/>
```

```
        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/Redirect/SLO"/>
        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/POST/SLO"/>
        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
SimpleSign" Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/POST-SimpleSign/SLO"/>
        <SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Location="https://shib-idp.yerbynet.com:8443/idp/profile/SAML2/SOAP/SLO"/>

        <SingleSignOnService Binding="urn:mace:shibboleth:1.0:profiles:AuthnRequest"
Location="https://shib-idp.yerbynet.com/idp/profile/Shibboleth/SSO"/>
        <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/POST/SSO"/>
        <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST-
SimpleSign" Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/POST-SimpleSign/SSO"/>
        <SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://shib-idp.yerbynet.com/idp/profile/SAML2/Redirect/SSO"/>
```

Ce fichier peut être customisé un peu plus avec les infos de votre organisation, mais bon, il faut tout faire marcher avant.

1- Branchement LDAP + tests avec testshib.org

[root@shib-idp shibboleth-idp]# ***vi conf/ldap.properties***

Les lignes que j'ai rajoutées ou modifiées sont :

```
idp.authn.LDAP.authenticator     = bindSearchAuthenticator
idp.authn.LDAP.ldapURL           = ldaps://ldap1.yerbynet.com ldaps://ldap2.yerbynet.com
idp.authn.LDAP.useStartTLS       = false
idp.authn.LDAP.useSSL            = true
idp.authn.LDAP.sslConfig         = certificateTrust
idp.authn.LDAP.baseDN            = cn=Users,dc=priv,dc=yerbynet,dc=com
idp.authn.LDAP.subtreeSearch     = true
idp.authn.LDAP.userFilter        = (sAMAccountName={user})
idp.authn.LDAP.bindDN            = cn=roger_user,cn=Users,dc=priv,dc=yerbynet,dc=com
idp.authn.LDAP.bindDNCredential              = roger_pass
idp.authn.LDAP.dnFormat          =%s@priv.yerbynet.com
```

Puis, rendez-vous ici : http://www.testshib.org/
Il faudra intégrer Testshib comme SP, et faire des tests jusqu'à ce que ça marche avant de continuer.

C'est très simple, il faut juste :
- enregistrer votre shibboleth sur leur site http://www.testshib.org/register.html (uploader votre fichier $IDP-HOME/metadata/idp-metadata.xml)
- configurer votre shibboleth comme décrit là http://www.testshib.org/configure.htm ; il s'agit de modifier le fichier $IDP-HOME/conf/metadata-providers.xml en ajoutant ces lignes :

```
<MetadataProvider id="HTTPMetadataTESTSHIB"
        xsi:type="FileBackedHTTPMetadataProvider"
        backingFile="%{idp.home}/metadata/testshib-providers.xml"
        metadataURL="http://www.testshib.org/metadata/testshib-providers.xml"/>
```

- redémarrer votre IdP avec Tomcat : ***systemctl restart tomcat***

Pour tester, vous pouvez URLer ceci : https://shib-idp.yerbynet.com/idp/profile/SAML2/Unsolicited/SSO?providerId=https://sp.testshib.org/shibboleth-sp ou bien aller sur https://sp.testshib.org/.
Refaites aussi les tests précédents svp.

Si ça fonctionne, vous méritez 2 grands verre de whisky, suivi d'un bon gros dodo. Malheureusement, je ne donne pas dans les stupéfiants (en tout cas pas encore), donc je ne saurai pas quoi recommander.

2- Résolution des attributs LDAP « conf/attribute-resolver.xml »

Dans ce fichier, on définit comment est-ce qu'on convertit nos attributs LDAP en Shibboleth, avant qu'ils ne soient envoyés à nos fournisseurs de services qui en demandent.

Le fichier intéressant est « attribute-resolver-full.xml », recopiez le sur « attribute-resolver.xml ».

[root@shib-idp shibboleth-idp]# ***cp conf/attribute-resolver-full.xml conf/attribute-resolver.xml***

[root@shib-idp shibboleth-idp]# ***vi conf/attribute-resolver.xml***

i- Enlever les commentaires pour la définition des attributs dans les 3 sections : « Schema: Core schema attributes », « Schema: inetOrgPerson attributes », et « Schema: eduPerson attributes ».

ii- Remplacer l'ID de l'attribut « mail » par « email » :

```
<AttributeDefinition xsi:type="Simple" id="mail" sourceAttributeID="mail">
```

devient

```
<AttributeDefinition xsi:type="Simple" id="email" sourceAttributeID="mail">
```

iii- Modifier l'attribut « uid » pour le faire pointer vers quelque chose de vraiment unique dans votre LDAP. Exemple pour AD :

```
<AttributeDefinition xsi:type="Simple" id="uid" sourceAttributeID="uid">
<AttributeDefinition id="uid" xsi:type="Simple" sourceAttributeID="sAMAccountName">
```

iv- Ajouter la définition de l'attribut commonName :

```
<AttributeDefinition xsi:type="Simple" id="commonName" sourceAttributeID="cn">
  <Dependency ref="myLDAP" />
  <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-def:cn"
    encodeType="false" />
```

```
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:2.5.4.3"
friendlyName="displayName" encodeType="false" />
  </AttributeDefinition>
```

v- Décommenter et adapter le data-connector LDAP. Ça ressemble à ceci :

```
    <DataConnector id="myLDAP" xsi:type="LDAPDirectory"
        ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
        baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
        principal="%{idp.attribute.resolver.LDAP.bindDN}"
        principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
        useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:true}"
        trustFile="%{idp.attribute.resolver.LDAP.trustCertificates}"
        connectTimeout="%{idp.attribute.resolver.LDAP.connectTimeout}"
        responseTimeout="%{idp.attribute.resolver.LDAP.responseTimeout}">
        <FilterTemplate>
        <![CDATA[
            (sAMAccountName=$requestContext.principalName)
        ]]>
        </FilterTemplate>
        <ConnectionPool
          minPoolSize="%{idp.pool.LDAP.minSize:3}"
          maxPoolSize="%{idp.pool.LDAP.maxSize:10}"
          blockWaitTime="%{idp.pool.LDAP.blockWaitTime:PT3S}"
          validatePeriodically="%{idp.pool.LDAP.validatePeriodically:true}"
          validateTimerPeriod="%{idp.pool.LDAP.validatePeriod:PT5M}"
          expirationTime="%{idp.pool.LDAP.idleTime:PT10M}"
          failFastInitialize="%{idp.pool.LDAP.failFastInitialize:false}" />
        <LDAPProperty name="java.naming.referral" value="follow"/>
    </DataConnector>
```

vi- Enlever les attributs non commentés et non pertinents pour votre installation.

vii- Redémarrer votre Shibboleth/Tomcat et scruter les logs.

viii- Refaire les tests qui fonctionnaient et s’assurer qu’ils fonctionnent toujours avant de continuer.

ix- Bonus attribut: Exemple d’attribut scripté

```
   <AttributeDefinition xsi:type="ScriptedAttribute" id="isMemberOf">
    <Dependency ref="eduPersonEntitlement" />
       <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-def:isMemberOf"
encodeType="false" />
            <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.5.1.1"
friendlyName="isMemberOf" encodeType="false" />
    <Script><![CDATA[
        logger = Java.type("org.slf4j.LoggerFactory").getLogger("net.shibboleth.idp.attribute");

        if (eduPersonEntitlement.getValues().get(0) == "enfant" ) {
```

```
            isMemberOf.addValue("Authorized");
        }
        else {
            isMemberOf.addValue("Unauthorized");
        }
    logger.info("Values of eduPersonEntitlement were: " + eduPersonEntitlement.getValues().get(0));
        ]]>
    </Script>
</AttributeDefinition>
```

2’- Connexion de attribute-resolver avec MySQL :

[root@shib-idp shibboleth-idp]# ***mysql***

MariaDB [(none)]> ***create database shibboleth;***

MariaDB [(none)]> ***grant all on shibboleth.* to shibboleth_user@'localhost' identified by "lemotdepassedeshibboleth-oui-oui";***

MariaDB [(none)]> ***use shibboleth;***

MariaDB [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`)
) DEFAULT CHARSET=utf8;

MariaDB [shibboleth]> ***CREATE TABLE shibpid (***
localEntity VARCHAR(255) NOT NULL,
peerEntity VARCHAR(255) NOT NULL,
persistentId VARCHAR(50) NOT NULL,
principalName VARCHAR(50) NOT NULL,
localId VARCHAR(50) NOT NULL,
peerProvidedId VARCHAR(50) NULL,
creationDate TIMESTAMP NOT NULL,
deactivationDate TIMESTAMP NULL,
PRIMARY KEY (localEntity(50), peerEntity(50), persistentId(50))
) DEFAULT CHARSET=utf8;

[root@shib-idp shibboleth-idp]# ***wget https://github.com/REANNZ/arcs-shibext/releases/download/1.8.3/arcs-shibext-1.8.3.jar***
[root@shib-idp shibboleth-idp]# ***mv arcs-shibext-1.8.3.jar $IDP_HOME/edit-webapp/WEB-INF/lib/***
[root@shib-idp shibboleth-idp]# ***cd $IDP_HOME***
[root@shib-idp shibboleth-idp]# ***sh $IDP_HOME/bin/build.sh***

i- Moodification du fichier « IDP_HOME/conf/saml-nameid.properties »
Générer un salt convenable (plus de 16 caractères) avec la commande ***openssl rand -base64 36***
Décommenter et ajuster ces lignes :

```
idp.persistentId.sourceAttribute = uid
idp.persistentId.salt = <valeur-salt-générée>
idp.persistentId.generator = shibboleth.StoredPersistentIdGenerator
idp.persistentId.dataSource = shibboleth.JPAStorageService.DataSource
```

ii- Modification de « IDP_HOME/conf/saml-nameid.xml »
Juste décommenter cette ligne :

```
    <ref bean="shibboleth.SAML2PersistentGenerator" />
```

iii- Modification de « $IDP_HOME/conf/c14n/subject-c14n.xml »
Juste décommenter cette ligne :

```
    <ref bean="c14n/SAML2Persistent" />
```

iv- Modification de « IDP_HOME/conf/global.xml »
Ajouter ces beans :

```
  <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="persistenceUnitName" value="storageservice" />
    <property name="packagesToScan" value="org.opensaml.storage.impl" />
    <property name="dataSource" ref="shibboleth.JPAStorageService.DataSource" />
    <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">
    <property name="database" value="MYSQL" />
  </bean>

  <bean id="shibboleth.JPAStorageService.DataSource"
    class="org.apache.tomcat.jdbc.pool.DataSource" destroy-method="close" lazy-init="true"
    p:driverClassName="com.mysql.jdbc.Driver"
    p:url="jdbc:mysql://localhost:3306/shibboleth?
autoReconnect=true&amp;sessionVariables=wait_timeout=31536000"
```

```
      p:validationQuery="/* ping */ SELECT 1;"
      p:testOnBorrow="true"
      p:username="shibboleth_user"
      p:password="lemotdepassedeshibboleth-oui-oui" />
```

v- Modification de « IDP_HOME/conf/idp.properties»
Décommenter et modifier les lignes suivantes :

```
idp.session.StorageService = shibboleth.JPAStorageService
idp.consent.StorageService = shibboleth.JPAStorageService
idp.replayCache.StorageService = shibboleth.JPAStorageService
idp.artifact.StorageService = shibboleth.JPAStorageService
```

vi- Retour à « IDP_HOME/conf/attribute-resolver.xml »
Ajouter ce data-connector :

```
  <DataConnector id="StoredId"
        xsi:type="StoredId"
        generatedAttributeID="persistentId"
        sourceAttributeID="uid"
        salt="%{idp.persistentId.salt}">
    <Dependency ref="uid" />
    <ApplicationManagedConnection
        jdbcDriver="com.mysql.jdbc.Driver"
        jdbcURL="jdbc:mysql://localhost:3306/shibboleth?autoReconnect=true"
        jdbcUserName="shibboleth"
        jdbcPassword="lemotdepassedeshibboleth-oui-oui"" />
  </DataConnector>
```

Ajouter un nouvel attribut « eduPersonTargetedID » dépendant de « StoreID ».

```
<AttributeDefinition id="eduPersonTargetedID" xsi:type="SAML2NameID"
        sourceAttributeID="persistentId">
  <Dependency ref="StoredId" />
  <AttributeEncoder xsi:type="SAML1XMLObject" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
encodeType="false" />
  <AttributeEncoder xsi:type="SAML2XMLObject" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.10"
friendlyName="eduPersonTargetedID" encodeType="false" />
</AttributeDefinition>
```

Un petit restart de tomcat pour vérifier que tout fonctionne toujours avec tests et inspections de logs.

vii- On va définir quelques attributs statiques et on en aura fini avec ces conneries.
Toujours dans « $IDP_HOME/conf/attribute-resolver.xml », voici un exemple que moi, je peux ajouter. Vous, trouvez ce que vous devez ajouter.

```
  <DataConnector id="staticAttributes" xsi:type="Static">
    <Attribute id="o">
      <Value>Entreprise YerbyNET</Value>
```

```
        </Attribute>
        <Attribute id="c">
            <Value>BF</Value>
        </Attribute>
        <Attribute id="co">
            <Value>Burkina Faso</Value>
        </Attribute>
        <Attribute id="schacHomeOrganization">
            <Value>yerbynet.com</Value>
        </Attribute>
        <Attribute id="eduPersonAffiliation">
            <Value>member</Value>
        </Attribute>
        <Attribute id="eduPersonScopedAffiliation">
            <Value>member</Value>
        </Attribute>
        <Attribute id="homeOrganization">
            <Value>www.yerbynet.com</Value>
        </Attribute>
    </DataConnector>

<AttributeDefinition id="eduPersonAffiliation" xsi:type="Simple">
    <Dependency ref="staticAttributes" />
    <DisplayName xml:lang="en">Affiliation type</DisplayName>
    <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-
def:eduPersonAffiliation" />
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
friendlyName="eduPersonAffiliation" />
</AttributeDefinition>

 <AttributeDefinition xsi:type="Scoped" id="eduPersonScopedAffiliation" scope="%{idp.scope}"
sourceAttributeID="eduPersonAffiliation">
        <Dependency ref="staticAttributes" />
        <AttributeEncoder xsi:type="SAML1ScopedString" name="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation" encodeType="false" />
        <AttributeEncoder xsi:type="SAML2ScopedString" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9"
friendlyName="eduPersonScopedAffiliation" encodeType="false" />
    </AttributeDefinition>

<AttributeDefinition id="friendlyCountryName" xsi:type="Simple" sourceAttributeID="co">
    <Dependency ref="staticAttributes" />
    <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-def:co"
encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:0.9.2342.19200300.100.1.43"
friendlyName="co" encodeType="false" />
</AttributeDefinition>

<AttributeDefinition id="countryName" xsi:type="Simple" sourceAttributeID="c">
    <Dependency ref="staticAttributes" />
```

```
    <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-def:c"
encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:2.5.4.6" friendlyName="c"
encodeType="false" />
</AttributeDefinition>

<AttributeDefinition id="schacHomeOrganization" xsi:type="Simple"
sourceAttributeID="schacHomeOrganization">
    <Dependency ref="staticAttributes" />
    <AttributeEncoder xsi:type="SAML1String" name="urn:mace:dir:attribute-
def:schacHomeOrganization" encodeType="false" />
    <AttributeEncoder xsi:type="SAML2String" name="urn:oid:1.3.6.1.4.1.25178.1.2.9"
friendlyName="schacHomeOrganization" encodeType="false" />
</AttributeDefinition>
```

viii- Un petit restart de tomcat pour vérifier que tout fonctionne toujours, accompagné de tests et inspections de logs.
À partir d'ici, le plus gros est passé. C'était la partie la plus longue et complexe.
Le reste consiste à ajouter des relying-parties (généralement des fournisseurs de services), et quels attributs est-ce qu'on relâche en fonction des fournisseurs.

Svp, refaites quelques tests avant de continuer. Merci.

3- « conf/attribute-filter.xml »

Quel fournisseur de services a droit à quels attributs. Donc, ce fichier filtre des attributs en fonction des fournisseurs.
Ce fichier va être modifié lorsque vous connectez un nouveau fournisseur de services. Généralement, le fournisseur doit vous dire les attributs dont il a besoin. Il se peut que vous ne soyez pas d'accord pour les lui livrer (parce que trop personnel par exemple), mais ca, c'est une autre histoire.
Dans notre cas, on va releaser un attribut à tous (ANY), et on va releaser quelques attributs pour test-shibboleth. Ca nous permettra de tester, et de voir comment testshib réagit. Supprimez les 2 exemples du fichier « attribute-filter.xml », et ajouter ceux-ci :

```
    <!--  Release the transient ID to anyone -->
    <AttributeFilterPolicy id="releaseTransientIdToAnyone">
        <PolicyRequirementRule xsi:type="ANY" />
        <AttributeRule attributeID="transientId">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>
    </AttributeFilterPolicy>
```

```
<!-- Release givenName to sp.testshib.org →
    <AttributeFilterPolicy id="releaseToSPtestshib">
        <PolicyRequirementRule xsi:type="Requester" value="https://sp.testshib.org/shibboleth-sp" />
        <AttributeRule attributeID="eduPersonAffiliation">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>
```

```
    <AttributeRule attributeID="eduPersonEntitlement">
        <PermitValueRule xsi:type="ANY" />
    </AttributeRule>
    <AttributeRule attributeID="eduPersonTargetedID">
          <PermitValueRule xsi:type="ANY" />
    </AttributeRule>
  <AttributeRule attributeID="givenName">
      <PermitValueRule xsi:type="ANY" />
  </AttributeRule>
</AttributeFilterPolicy>
```

Pour vos autres fournisseurs, vous aurez sûrement à modifier ce fichier pour relâcher les attributs dont ils ont besoin.

4- « conf/metadata-providers.xml »

C'est ici que vous mettrez les metadata de vos fédérations. Ce fichier a par exemple déjà été modifié pour y ajouter testshib.org.

Voici quelques exemples de configurations :

```
<MetadataProvider id="PremierFederateur" xsi:type="FileBackedHTTPMetadataProvider"
            metadataURL="http://shib2.net/CoreServices/metadata_signed_sha512.xml"
            backingFile="/opt/shibboleth-idp/metadata/shib2net_metadata_signed.xml">
            <MetadataFilter xsi:type="SignatureValidation"
                certificateFile="${idp.home}/credentials/md-signer.crt"
                requireSignedRoot="true" />
</MetadataProvider>
```

```
<MetadataProvider id="FederationMetadata2" xsi:type="FileBackedHTTPMetadataProvider"
      xmlns="urn:mace:shibboleth:2.0:metadata"
      xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
      xsi:schemaLocation="urn:mace:shibboleth:2.0:metadata
      http://shibboleth.net/schema/idp/shibboleth-metadata.xsd"
      metadataURL="http://federation.example.org/metadata.xml"
      backingFile="%{idp.home}/metadata/federation-metadata2.xml">
    <MetadataFilter xsi:type="SignatureValidation" requireSignedRoot="true"
      certificateFile="%{idp.home}/credentials/fed-signing-key.pem"/>
</MetadataProvider>
```

Avec ces 2 exemples, ces 2 fichiers « $IDP_HOME/credentials/md-signer.crt » et « $IDP_HOME/credentials/fed-signing-key.pem » doivent être créés. Il faudra les récupérer (télécharger ou récupérer sur des disquettes livrer par mobylettes) depuis vos 2 relying-parties (shib2.net et shibboleth.net).

Une petite relance de Tomcat s'impose.

Les 2 fichiers « /opt/shibboleth-idp/metadata/shib2net_metadata_signed.xml » et « %{idp.home}/metadata/federation-metadata2.xml » doivent avoir été créés, ainsi que celui de testshib qui existait déjà.

5- conf/relying-party.xml

On pourrait (très peu probable) avoir besoin de modifier ce fichier pour spécifier les paramètres SAML qu'on souhaiterait supporter pour un relying-party.

Voici un exemple de modification que j'ai faite.
J'ai ajouté des lignes semblables dans la liste « shibboleth.RelyingPartyOverrides ».

```
    <bean id="shibboleth.adfsgw" parent="RelyingPartyByName"
c:relyingPartyIds="#{{'http://adfs.exemple1.com/adfs/services/trust',
'https://examen.exple2.net/shibboleth',
                    'https://exple3.org/shibboleth'}}">
      <property name="profileConfigurations">
        <list>
            <bean parent="SAML2.SSO" p:authenticationFlows="#{{'Password'}}"
            p:postAuthenticationFlows="#{{'attribute-release', 'terms-of-use'}}"
            p:signAssertions="true"
            p:encryptAssertions="false" />
            <bean parent="SAML2.ECP" />
            <bean parent="SAML2.Logout" />
            <bean parent="SAML2.AttributeQuery" />
            <bean parent="SAML2.ArtifactResolution" />
            <bean parent="Liberty.SSOS" />
        </list>
      </property>
    </bean>
```

6- Cosmétiques et derniers petits réglages

i- Si vous utilisez AD, il serait intéressant de faire cette modification dans le fichier
« $IDP_HOME/conf/authn/password-authn-config.xml »

```
<util:constant id="shibboleth.authn.Password.Lowercase" static-field="java.lang.Boolean.~~FALSE~~"/>
<util:constant id="shibboleth.authn.Password.Lowercase" static-field="java.lang.Boolean.TRUE"/>
```

ii- Si votre LDAP n'est pas un AD, cette ligne ci-dessous n'est pas necessaire dans votre fichier attribute-resolver.xml

```
~~<LDAPProperty name="java.naming.referral" value="follow"/>~~
```

iii- « conf/intercept/consent-intercept-config.xml ». Dans ce fichier, on va essayer d'ordonner l'affichage des attributs dans la fenêtre de consentement. Ouvrez le, et faites quelques chose semblable à ceci :

```
<util:list id="shibboleth.consent.attribute-release.AttributeDisplayOrder">
<value>uid</value>
<value>displayName</value>
<value>commonName</value>
<value>givenName</value>
<value>surname</value>
<value>email</value>
<value>eduPersonPrincipalName</value>
<value>organizationName</value>
<value>postalAddress</value>
<value>telephoneNumber</value>
<value>mobileNumber</value>
</util:list>
```

iv - « conf/idp.properties »

```
# Whether attribute values and terms of use text are compared
#idp.consent.compareValues = ~~false~~
idp.consent.compareValues = true
# Maximum number of consent records for space-limited storage (e.g. cookies)
#idp.consent.maxStoredRecords = ~~10~~
idp.consent.maxStoredRecords = -1
```

v- « messages/messages.properties »
Customisation des messages et uploader votre logo ici « edit-webapp/images/logo.png ».

vi- Renouvellement quotidien de la clé secrète du DataSealer ⇒ crontab « /etc/cron.d/shibboleth »

```
21 01 * * * tomcat IDP_HOME=/opt/shibboleth-idp JAVA_HOME=/usr/lib/jvm/java /opt/shibboleth-idp/bin/seckeygen.sh --versionfile /opt/shibboleth-idp/credentials/sealer.kver --storefile /opt/shibboleth-idp/credentials/sealer.jks --storepass motdepasse  --alias secret
```

Petite commande pour lister les clés : ***keytool -v -list -keystore sealer.jks -storepass <secret> -storetype JCEKS***
Par défaut, il y aura jusqu'à 30 clés.

7- Redémarrage et derniers tests

Redémarrer tout ce que vous pouvez redémarrer, et re-tester tout. Si ca fonctionne, enlevez le mode DEBUG de vos logs.
Test de reload de service via URL :
URL1 : *https://shib-idp.yerbynet.com/idp/profile/admin/reload-service?id=shibboleth.LoggingService*
URL2 : *https://shib-idp.yerbynet.com/idp/profile/admin/reload-service?id=shibboleth.AttributeFilterService*
Test de refresh de metadata via URL :
URL : *https://shib-idp.yerbynet.com/idp/profile/admin/reload-metadata?id=HTTPMetadataTESTSHIB*
Cette commande ***$IDP_HOME/bin/aacli.sh*** permet de faire des tests en ligne de commandes.
Exemple :

$IDP_HOME/bin/aacli.sh -n roger_user -r https://sp.testshib.org/shibboleth-sp -u https://shib-idp.yerbynet.com/idp

Chef(s), vous méritez un bon champagne.

Bon bon bon !
Je pense qu'on a fait le tour. Et vous, vous avez suffisamment été félicité(e)(s). J'espère que vous avez bien abusé du champagne, whisky et bière que je vous offrais.
Quant à ce document, il dépasse les 20 pages, donc, dans un monde où personne n'a le temps, il n'est plus intéressant.
Quant à moi, j'en ai marre. Je ne vais quand même pas relire plus de 20 pages de documentations techniques extrêmement ennuyantes à la recherche de fautes. Ça me rappelle l'histoire de l'écrivain qui a arrêté l'écriture parce qu'il s'ennuyait quand il se relisait. Je relirai et éventuellement je corrigerai la prochaine fois que j'installe un autre shibboleth.

Donc sur ce, « merci bonjour ! ».

Roger YERBANGA
www.yerbynet.com

Sources :

https://shibboleth.net/about/
https://wiki.shibboleth.net/confluence/display/IDP30/Installation#Installation-BeforeYouBegin
https://wiki.shibboleth.net/confluence/display/SHIB2/IdPLinuxNonRoot
https://wiki.shibboleth.net/confluence/display/IDP30/ApacheTomcat8
https://wiki.shibboleth.net/confluence/display/IDP30/SecurityAndNetworking#SecurityAndNetworking-Apache
https://wiki.shibboleth.net/confluence/display/IDP30/ReloadableServices
https://tuakiri.ac.nz/confluence/display/Tuakiri/Installing+a+Shibboleth+3.x+IdP
https://wiki.shibboleth.net/confluence/display/IDP30/SecretKeyManagement
https://wiki.shibboleth.net/confluence/display/IDP30/AttributeFilterLegacyNameSpaceMapping : syntaxe pour attribut-filter
https://www.switch.ch/aai/support/presentations/shibboleth-training-2015/T3P09-User_Consent.pdf
https://wiki.shibboleth.net/confluence/display/IDP30/UpgradingFromV2
https://www.switch.ch/aai/guides/idp/installation/#servletcontainer
https://spaces.internet2.edu/pages/viewpage.action?pageId=49841792#LinuxIdentityProviderIdPv3(RHEL7)-5.ConfigureUserAttributes/AttributeResolver : pas mal
https://github.com/malavolti/HOWTO-Install-and-Configure-Shibboleth-Identity-Provider/blob/master/HOWTO%20Install%20and%20Configure%20a%20Shibboleth%20v3.2.1%20on%20Ubuntu%20Linux%20LTS%2014.04%20with%20Tomcat%208%20only.md
http://www.testshib.org/test.html : pour tester son shibboleth

https://github.com/REANNZ/arcs-shibext/blob/master/INSTALL-SharedToken.md