Shibboleth SP con Debian

Francesco Malvezzi

Università di Modena e Reggio nell'Emilia

30 marzo 2009

Introduzione

Lo scopo di questo tutorial è:

- installare un Shibboleth Service Provider 2.0 su Debian Lenny;
- provarlo.

Prerequisiti

- web server Apache2.2
- ntp

Installazione del servizio.

Download del modulo:

```
apt-get install libapache2_mod_shib2
```

Attenzione al “2” di shib2 (esiste anche libapache2_mod_shib)
Creazione di una cartella protetta: aggiungere in /etc/apache2/sites-available/default[1]

```
<Location /secure>
 AuthType shibboleth
 ShibRequireSession On
 require valid-user
</Location>
```

Abilitazione del modulo:

```
a2enmod shib2
/etc/init.d/apache2 force-reload
```

[1] Ricordarsi di creare la cartella “secure”.

File di configurazione - panoramica

Si trovano in `/etc/shibboleth/`

- shibboleth2.xml – impostazioni generali del servizio;
- attribute-map.xml – definisce la conversione tra gli attributi ricevuti dallo IdP e le variabili server;
- attribute-policy.xml – definisce l'accettabilità degli attributi a partire dal loro formato.

Test della correttezza sintattica del servizio:

```
shibd -t /etc/shibboleth/shibboleth2.xml
```

shibboleth2.xml

Minimo indispensabile da variare:

- EntityID in ApplicationDefaults;
- SessionInitiator, con l'url dell'IdP da contattare;
- MetadataProvider;
- CredentialResolver, con il path per chiave e certificato;
- Errors, con i propri riferimenti;

shibboleth2.xml - segue

Esempio di SessionInitiator:

```
 <SessionInitiator type="Chaining" Location="/IDEM"
id="WAYF" isDefault="true" relayState="cookie">
                <SessionInitiator type="SAML2" defaultACSIndex="1"
              template="bindingTemplate.html"/>
                <SessionInitiator type="Shib1" defaultACSIndex="5"/>
                <SessionInitiator type="WAYF" defaultACSIndex="5"
        URL="https://wayf.idem.garr.it/WAYF"/>
            </SessionInitiator>
<SessionInitiator type="Chaining" Location="/Login"  id="Intranet"
                    relayState="cookie"
                    entityID="https://idem-idp.dmz-ext.unimo.it/idp/shibboleth">
                <SessionInitiator type="SAML2" defaultACSIndex="1"
           template="bindingTemplate.html"/>
                <SessionInitiator type="Shib1" defaultACSIndex="5"/>
            </SessionInitiator>
```

Si usa lo Id nella sezione RequestMap per richiedere uno specifico IdP:

```
<RequestMapper type="Native">
 <RequestMap applicationId="default">
  <Host name="test-sp.dmz-ext.unimo.it">
   <Path name="idem" authType="shibboleth" requireSession="true" requireSessionWith="WAYF"/>
   <Path name="test-idp" authType="shibboleth" requireSession="true"
              requireSessionWith="Intranet"/>
  </Host>
 </RequestMap>
</RequestMapper>
```

shibboleth2.xml - segue

Esempio di MetadataProvider:

```
<MetadataProvider type="Chaining">
          <!-- Esempio di metadata firmato su url remoto. -->
          <MetadataProvider type="XML"
               uri="https://www.idem.garr.it/docs/conf/signed-metadata.xml"
               backingFilePath="idem-metadata.xml" reloadInterval="7200">
             <SignatureMetadataFilter certificate="signer_bundle.pem"/>
          </MetadataProvider>

          <!-- Metadata locale -->
          <MetadataProvider type="XML" file="metadata2.xml"/>
   </MetadataProvider>
```

attribute-map.xml

Esempio del mappaggio di un attributo:

```
<Attribute name="urn:mace:dir:attribute-def:eduPersonPrincipalName"
    id="eppn">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
</Attribute>
<Attribute name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" id="eppn">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder"/>
</Attribute>
```

Nell'esempio si esporta nella variabile d'ambiente `eppn` l'attributo in formato SAML1 "urn:mace:dir:attribute-def:eduPersonPrincipalName" e quello in formato SAML2 "urn:oid:1.3.6.1.4.1.5923.1.1.1.6".

In questo modo lo SP può ricevere asserzioni SAML da IdP Shibboleth2.* o Shibboleth1.*

Generare il frammento di metadata

- http://localhost/Shibboleth.sso/Status
- http(s)?://FQDN/Shibboleth.sso/Metadata

I log

I log di Shibboleth sono:

/var/log/shibboleth/shibd.log funzionamento del demone, dialogo con lo IdP, scambio degli attributi;

/var/log/shibboleth/transaction.log log forensico: tiene traccia delle transazioni effettuate con lo IdP; il valore del NameIdentifier permette di collegare le entry di questo log con quelle dello IdP (shib-access.log);

/var/log/apache2/native.log autorizzazione;

Come opzione predefinita del .deb l'ultimo file è sostituito dalla ridirezione in /var/log/messages tramite syslog.
Per creare il native.log, decommentarne la sezione in /etc/shibboleth/native.logger ed eseguire:

```
touch /var/log/apache2/native.log
chown www-data /var/log/apache2/native.log
```

Controllo accesso con HtAccessControl

In una sezione `<Location>` dopo i necessari:

```
AuthType shibboleth
ShibRequireSession On
```

è possibile aggiungere una o più direttive tipo:

```
require affiliation staff@uniprova.it member@uniprova.it
```

che sono in `OR` logico tra di loro a meno che non sia definito:

```
ShibRequireAll on
```

È possibile usare le espressioni regolari con una tilde dopo l'alias:

```
require affiliation ~ ^staff@uni.*\.it$
```

Lazy sessions

Talvolta può essere necessario non obbligare l'utente all'autenticazione Shibboleth. Ad esempio possono coesistere due autenticazioni (Shibboleth ed una locale) tra cui l'utente può scegliere.
Sono necessarie due modifiche:

sessioni pigre	sessioni strette
<Location /secure> AuthType shibboleth Require shibboleth </Location>	<Location /secure> AuthType shibboleth ShibRequireSession On require valid-user </Location>

- in `shibboleth.xml` settare a `false` il requireSession della directory da proteggere:

```
<Path name="secure" authType="shibboleth" \
 requireSession="false"/>
```

Lazy sessions – 2

E per permettere l'autenticazione Shibboleth in una lazy session?
Ridirigere l'utente al concatenamento:

- hostname dello SP;
- indirizzo del WAYF (come specificato nel SessionInitiator);
- stringa fissa: "?target="
- indirizzo completo cui ridirigere l'utente dopo l'autenticazione (url-encoded)

Esempio:

```
http://idem-sp.dmz-ext.unimo.it/Shibboleth.sso/Intranet? \
 target=https%3A%2F%2Fidem-sp.dmz-ext.unimo.it%2Fsecure
```

webisoget

È una utility per lo scripting di interazioni http che includono autenticazioni shibboleth (può gestire il dialogo attraverso un WAYF).
Download:
`http://staff.washington.edu/fox/webisoget/`

Installazione con

```
./configure --prefix=/opt/webisoget
make
make install
```

Creare un file con le credenziali:

```
j_username=user; j_password=password
```

Creare un script per automatizzare il lancio di webisoget

```
#!/bin/bash

export LD_LIBRARY_PATH=/opt/webisoget/lib/

/opt/webisoget/bin/webisoget -verbose -text -out text.txt \
        -formfile form.login \
        -url http://idem-sp.dmz-ext.unimo.it/secure/test.php
```

Riferimenti

- Documentazione ufficiale di Internet2 (`https://spaces.internet2.edu/display/SHIB2/Home`)
- Bellissimo howto di SWITCH per l'installazione del pacchetto da sorgente (`https://www.switch.ch/aai/docs/shibboleth/SWITCH/2.1/sp/deployment/debian-lenny-source.html`)
- tutorial di Giacomo Tenaglia, 2 aprile 2007 (`http://www.garr.it/meeting_aai/slide_sem/3sp.pdf`)