

Configurazione di Shibboleth IdP v.3.4

Abilitiamo l'accesso federato

Corso Shibboleth IdP - 22-23/01/2020

Giuseppe De Marco - giuseppe.demarco@unical.it

Differenze di configurazione

```
cd /opt/shibboleth-idp
```

```
git status
```

```
git stash
```

```
git log
```

```
# chown -R jetty: metadata conf credentials tmp logs
```

```
git checkout [master, configuration, federation]
```

```
# differenze con un'installazione di base
```

```
git diff master conf/*
```

```
# differenze delle modifiche per singoli files
```

```
git diff master conf/services.xml
```

Configurazione dei file **.properties**

1. `git diff master conf/idp.properties`
 - **idp.cookie.secure**, invio di cookie esclusivamente via TLS
 - **idp.storage**.[htmlLocalStorage](#)
 - **idp.session.trackSPSessions**, supporto SingleLogout (SLO)
 - **idp.session.secondaryServiceIndex**, supporto SLO
2. `git diff master conf/ldap.properties`
 - parametri di connessione al server LDAP
certificato self-signed: **certificateTrust** altrimenti **keyStoreTrust**
3. `git diff master conf/saml-nameid.properties`
 - **idp.persistentId.sourceAttribute** = uid
 - **idp.persistentId.salt**, caratteri come sale per produrre NameID

Configurazione dei file **.xml**

1. `git diff master conf/global.xml`
Configurazione della connessione ad un RDBMS (opzionale)
Quanto qui definito ha scopo globale (è richiamabile ovunque)
2. `git diff master conf/services.xml`
Aggiunta dei files personalizzati:
 - a. `attribute-resolver-*.xml`
 - b. `attribute-filter-*.xml`
 - c. `metadata-providers-*.xml`
3. `git diff master conf/metadata-providers.xml`
Abbiamo incluso i metadati dell'SP di test

Attribute-resolver

1. Definisce gli attributi supportati (AttributeDefinition) le fonti di questi (InputDataConnector, eg. *ref="myLDAP"*) la loro rappresentazione in SAML2 (AttributeEncoder)
2. Gli attributi **R&S**: le [specifiche](#) ci indicano quanti sono e come usarli
3. Abbiamo configurato in **services.xml**:

```
<value>{%idp.home}/conf/attribute-resolver-v3_4-idem-custom.xml</value>
```

```
<value>{%idp.home}/conf/attribute-resolver-dynamic.xml</value>
```

```
<value>{%idp.home}/conf/attribute-resolver-dbsql.xml</value>
```

Attribute-filter

1. Controlla il rilascio degli attributi supportati secondo specifiche regole
2. Consigliamo l'adozione di `FileBackedHTTPResource` per i filtri IDEM

```
<bean id="IDEM-Default-Filter"  
class="net.shibboleth.ext.spring.resource.FileBackedHTTPResource"
```

3. La configurazione ideale prevede singole definizioni per ogni scopo

```
<value>{%idp.home}/conf/attribute-filter-local.xml</value>  
<value>{%idp.home}/conf/attribute-filter-v3-RS-CoCo.xml</value>  
<ref bean="IDEM-Production-Filter"/>
```

Metadata Providers

1. Aggiunto SP di test in **metadata-providers.xml**

2. Abbiamo aggiunto inoltre in **services.xml**

```
<value>{%idp.home}/conf/metadata-providers-eduGAIN.xml</value>
```

Prendiamo visione di **metadata-providers-eduGAIN.xml**

```
less conf/metadata-providers-eduGAIN.xml
```

Services

1. Definisce i file di configurazione da caricare all'avvio di ShibIdP
2. I servizi possono essere riavviati a "caldo" (-u <http://localhost:8080/idp>)

```
bin/reload-service.sh -id shibboleth.AttributeResolverService
```

```
bin/reload-service.sh -id shibboleth.AttributeFilterService
```

```
bin/reload-service.sh -id shibboleth.MetadataResolverService
```

```
bin/reload-service.sh -id shibboleth.LoggingService
```

... Non stupiamoci se un riavvio del Servlet Container fosse necessario!

Test sul rilascio degli attributi

```
/opt/shibboleth-idp/bin/aaccli.sh -n luigi \  
-r https://shib-sp.aai-test.garr.it/shibboleth \  
--saml2 \  
-u http://localhost:8080/idp
```

```
tail -f /opt/shibboleth-idp/logs/idp-process.log
```

ATTENZIONE

In *services.xml* l'ordinamento all'interno delle liste è fondamentale.

Esempio:

```
path: "{ { idp_path } }/conf/services.xml"
regexp: '<value>{%idp.home}/conf/metadata-providers.xml</value>'
replace: '<value>{%idp.home}/conf/metadata-providers.xml</value>\n
         \t<value>{%idp.home}/conf/metadata-providers-eduGAIN.xml</value>'
replace: '<value>{%idp.home}/conf/metadata-providers-eduGAIN.xml</value>\n
         \t<value>{%idp.home}/conf/metadata-providers.xml</value>'
backup: yes
```

La modifica da **rosso** a **verde** ha sbloccato un apparente incomprensibile problema, vale il pattern: *"Gli ultimi saranno i primi"*.

ATTENZIONE agli identificatori sovrapposti!

Se i bean hanno id uguali, sebbene siano configurati in file diversi, l'ultimo file ad essere caricato sovrascrive i primi. Bisogna sempre verificare che non esistano ID uguali. Esempio per i filtri sugli attributi:

```
grep "id=" /opt/shibboleth-idp/conf/attribute-filter*
```

ATTENZIONE all'uso di RAM

- Java beve ... eseguire ShibIdP senza alcuna configurazione particolare richiede ~1,5Gb
- Caricare i Metadata di federazione IDEM/eduGAIN in ShibIDP richiede almeno 2GB di RAM (ad oggi)
- Altri GB sono necessari in base al volume di traffico del servizio
- In futuro sicuramente questa soglia relativa ai Metadata aumenterà di dimensione ... (la soluzione sarà MDQ)
- i servlet container pongono un limite all'uso di RAM, tararli in `/opt/jetty/start.ini` oppure `/etc/default/jetty`

Domande?

marco.malavolti@garr.it
giuseppe.demarco@unical.it
maurizio.festi@unitn.it