Specifiche SSOReggio

Specifiche per l'utilizzo, da parte delle procedure web, del Single Sign-On del Comune di Reggio Emilia

Data	Autore	Note
07/04/2016	Giovanni Spigoni	Configurazione generale e di Apache httpd
19/05/2016	Giovanni Spigoni	Aggiornato con configurazione shibboleth2.xml

Indice generale

Scopo del documento

Questo documento ha lo scopo di dare delle specifiche ai fornitori per l'integrazione di procedure web con il Single Sign-On Shibboleth del Comune di Reggio Emilia (SSOReggio). Il Single Sign-On in questione NON è il sistema di autenticazione interno della rete Microsoft basato su tecnologia NTLM anche se utilizza lo stesso repository utenti (Active Directory).

Configurazione del sistema di autenticazione

Il sistema di autenticazione attuale utilizza l'infrastruttura Shibboleth che a sua volta è basata, nel nostro caso, su tecnologia SAML2. Shibboleth viene utilizzato sia per l'Identity Provider (IdP), che effettua la vera e propria autenticazione, sia per il Service Provider (SP) che controlla se l'utente è autenticato in fase di accesso ad una procedura interna ed eventualmente reindirizza all'IdP.

Tuttavia qualsiasi SP può essere utilizzato per proteggere una procedura, basta che supporti la tecnologia SAML2. Quindi le procedure esterne che vogliono essere protette da SSOReggio non devono necessariamente essere protette da Shibboleth SP.

Il sistema di autenticazione Shibboleth utilizza il formato SAML2 per lo scambio di richieste e dei dati di un utente tra IdP e SP e utilizza i cookies del browser per identificare la sessione di autenticazione dell'utente.

Configurazione dell'Identity Provider

Per integrare un SP con l'IdP è necessario che l'IdP abbia accesso ai metadata dell'SP e che l'entity ID del service provider sia elencato tra i Relying Party dell'IdP.

I metadata dell'SP possono essere ottenuti o tramite un URL http e quindi scaricati dinamicamente dall'IdP, oppure possono essere un file statico salvato in locale sul server dell'IdP.

I gestori del service provider devono quindi fornire:

1. un entity ID, ovvero una stringa che identifichi il service provider in modo univoco dal punto di vista dell'IdP
2. un URL da cui scaricare i metadata o un file in formato xml contenente i metadata del service provider.

Nel caso di Shibboleth SP l'entity ID ha tipicamente il seguente formato:

https://<nome.host>/shibboleth-sp

inoltre l'URL di default per ottenere i metadata in Shibboleth SP è:

https://<nome.host>/Shibboleth.sso/Metadata.

É comunque possibile ottenere l'entity ID dai metadata dell'SP.

Configurazione del Service Provider

La configurazione dell'SP deve essere eseguita dai gestori della procedura, poiché la parte di software dell'SP è in esecuzione sullo stesso host, o comunque nello stesso dominio, della procedura.

Il service provider deve essere configurato in modo da reindirizzare gli utenti non autenticati all'IdP del Comune di Reggio Emilia. L'SP necessita quindi dell'entity ID dell'IdP che è:

https://ssoshib.comune.re.it/idp/shibboleth

l'entity ID dell'IdP rappresenta anche l'URL da cui è possibile scaricare i metadata dell'IdP.

Il service provider deve inoltre essere configurato in modo da mappare gli attributi dell'utente ricevuti dall'IdP con i suoi attributi interni in modo che questi vengano riconosciuti correttamente dalle procedure.

Lista attributi inviati dall'IdP

La lista degli attributi inviati dall'IdP è la seguente:

Nome attributo	Descrizione
trustLevel	Livello di registrazione dell'utente: Alto – utente registrato “de visu” tramite documento di identità Basso – utente registrato solo online senza documenti
policyLevel	Password policy: Alto - lunghezza minima password 8 caratteri di cui 4 alfabetici, 2 numerici, 1 extra alfanumerico; scadenza password 3 mesi Basso - lunghezza minima password 6 caratteri
userid	Identificativo dell'utente, in questo caso uguale al codice fiscale
emailAddressPersonale	Indirizzo email personale, NO PEC
CodiceFiscale	Codice fiscale
cognome	Cognome
nome	Nome
emailAddress	Indirizzo email PEC
indirizzoResidenza	Via e numero civico di residenza
capResidenza	CAP del comune di residenza
cittaResidenza	Città del comune di residenza
provinciaResidenza	Provincia di residenza
statoResidenza	Stato di residenza
lavoro	Professione dell'utente

indirizzoDomicilio	Via e numero civico del domicilio
capDomicilio	CAP del comune di domicilio
cittaDomicilio	Città di domicilio
provinciaDomicilio	Provincia di domicilio
statoDomicilio	Stato del domicilio
dataNascita	Data di nascita in formato LDAP: AAAAMMGG000000Z
luogoNascita	Comune di nascita
provinciaNascita	Provincia di nascita
sesso	Sesso dell'utente
telefono	Telefono 1
cellulare	Telefono 2
federaUserDomain	Dominio dell'utente utilizzato dal sistema di autenticazione federato FedERa

Service Provider più comuni

- Shibboleth SP https://shibboleth.net/products/service-provider.html
- SimpleSAMLphp https://simplesamlphp.org/
- OpenAM https://www.forgerock.com/platform/access-management/
- JOSSO (Community Ed.) http://www.josso.org/
- Authentic2 http://authentic2.readthedocs.org/en/stable/

Configurazione Shibboleth SP con server HTTP Apache

Sebbene sia possibile utilizzare qualsiasi service provider che supporti il protocollo SAML2, in questo documento viene descritta nel dettaglio la configurazione per il service provider Shibboleth SP.

Per una documentazione più generale, ma comunque esaustiva, di Shibboleth SP fare riferimento alla documentazione ufficiale:
https://wiki.shibboleth.net/confluence/display/SHIB2/Installation

Shibboleth SP presuppone che le chiamate alla procedura protetta vengano intercettate da un server HTTP Apache (httpd) o Microsoft Internet Information Services (IIS) e poi, una volta effettuata l'autenticazione dell'utente, queste vengano inoltrate all'application server tramite il modulo proxy. In questo documento viene descritta solo la configurazione di Shibboleth SP in presenza di un server Apache.

Per l'installazione di Shibboleth SP fare riferimento alla documentazione ufficiale. Per l'installazione su Windows è possibile scegliere la versione per Apache o per IIS. Per Linux

esiste solo la versione per Apache, le distribuzioni Linux ufficialmente supportate sono:

- Red Hat Enterprise and CentOS 6, 7
- SUSE Linux Enterprise Server 10, 11, 11-SP1, 11-SP2, 11-SP3, 11SP4, 12, 12SP1, 12SP2, 12SP3
- OpenSUSE Linux 13.1, 13.2

Shibboleth SP è composto da due parti:

- la prima parte è un modulo del server web Apache (httpd) e in quanto tale viene configurata attraverso i file .conf del server httpd,
- la seconda parte è un servizio (shibd), indipendente da httpd, i cui file di configurazione si trovano nella cartella <PATH_INSTALLAZIONE>/shibboleth/

Configurazione di Apache HTTP Server

Le configurazioni del modulo shibboleth di Apache sono all'interno del file di configurazione httpd.conf nella cartella di installazione del server. Tipicamente il modulo shibboleth viene configurato nel file shib.conf. Questo file viene poi importato nel file httpd.conf tramite la direttiva `Include shib.conf`.

Prima di tutto è necessario caricare il modulo shibboleth tramite la direttiva `LoadModule`:

```
LoadModule mod_shib /usr/lib64/shibboleth/mod_shib_24.so
```

controllare che il path del file mod_shib_24.so sia corretto.

Le direttive necessarie per proteggere una risorsa sono le seguenti:

```
AuthType shibboleth
ShibRequestSetting requireSession 1
Require valid-user
```

Tipicamente le direttive per proteggere una procedura sono contenute in un tag <Location>, questo permette di proteggere un determinato percorso, ma di lasciare pubblico accesso alle altre risorse del server. Il tag <Location> può essere a sua volta contenuto in un virtual host per specificare ulteriormente l'ambito (il nome, l'ip o la porta dell'host) che deve essere protetto da Shibboleth SP. Se le direttive non sono contenute in un tag <Location> tutte le procedure servite dal server Apache, o da un determinato virtual host creato dal server Apache, richiederanno l'autenticazione.

Una tipica configurazione di Apache per proteggere una risorsa è la seguente:

```
<Location /protected>
      AuthType shibboleth
      ShibRequestSetting requireSession 1
      ShibUseHeaders On
      require valid-user
      ProxyPass ajp://nome_application_server:num_porta/protected
```

```
</Location>
```

In questo caso la risorsa /protected viene pubblicata da un application server per cui Apache inoltra la chiamata attraverso il modulo proxy utilizzando il protocollo AJP13. In questo caso è necessario caricare in Apache anche i moduli mod_proxy e mod_proxy_ajp. Ovviamente è possibile utilizzare anche il protocollo HTTP per il proxy.

La direttiva `ProxyPass` non è necessaria se la procedura da proteggere è servita direttamente dal server Apache.

Da notare inoltre la direttiva `ShibUseHeaders On`. Questa è necessaria se si vuola far passare gli attributi ricevuti dall'IdP alla procedura tramite gli header HTTP. Questa direttiva non è obbligatoria ed è ritenuto più sicuro (stando alla documentazione di Shibboleth SP) passare gli attributi come variabili d'ambiente. Nel caso del protocollo AJP però, per passare una variabile d'ambiente alla procedura, è necessario aggiungere al nome della variabile il prefisso “AJP_”. Questo si può fare, come vedremo in seguito, configurando correttamente il servizio shibd nel file shibboleth2.xml.

Per maggiori informazioni riguardo al passaggio degli attributi dal server HTTP alla procedura, fare riferimento alla documentazione di Shibboleth SP:

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeAccess

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPJavaInstall

Configurazione di Apache per protezione passiva (lazy login)

Talvolta è necessario pubblicare una pagina anche ad utenti non autenticati dando la possibilità, agli utenti già autenticati in un'altra procedura o in un'altra risorsa protetta, di visualizzare qualche contenuto aggiuntivo, come ad esempio semplicemente il nome dell'utente autenticato.

Nell'esempio successivo è riportata la configurazione per due location differenti. La location /protected è uguale a quella vista nella sezione precedente, la location /public invece è protetta in modo passivo: non è necessario essere autenticati per accedere alla location, però, se un utente è già autenticato, l'applicazione avrà la possibilità di accedere agli attributi dell'utente per poter fornire contenuti aggiuntivi.

```
<Location /protected>
      AuthType shibboleth
      ShibRequestSetting requireSession 1
      ShibUseHeaders On
      require valid-user
      ProxyPass ajp://nome_application_server:num_porta/protected
</Location>

<Location /public>
      AuthType shibboleth
      ShibUseHeaders On
```

```
      require shibboleth
      ProxyPass ajp://nome_application_server:num_porta/public
</Location>
```

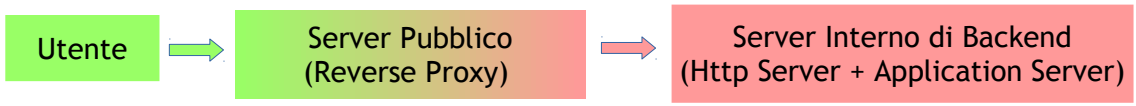
Nelle pagine sotto la location /public sarà possibile inserire dei link alla pagina di login. L'URL della pagina di login può avere il seguente formato:

https://<nome_host>/Shibboleth.sso/Login?target=/public

Se l'utente clicca su questo link gli verrà richiesto di autenticarsi, dopo di che l'utente verrà reindirizzato alla risorsa /public.

Configurazione di Apache dietro a Reverse Proxy

Una tipica configurazione di rete utilizzata per la pubblicazione di procedure o siti web è la seguente (al netto di firewall e altri nodi di rete):



dove la connessione tra l'utente finale e il *reverse proxy* è in HTTPS mentre la connessione tra *reverse proxy* e server interno è in HTTP.

In casi come questo è comunque necessario che il modulo Shibboleth di Apache riesca ad ottenere un url autoreferenziale per creare correttamente la richiesta SAML da inviare all'IdP. Per questo è necessario che in Apache venga aggiunta la direttiva `ServerName` e `UseCanonicalName`:

```
<VirtualHost *:80>

    ServerName https://nome_server_pubblico:443
    UseCanonicalName On

  <Location /protected>
      ...
  </Location>

</VirtualHost>
```

Lo schema https e la porta 443 nella direttiva ServerName sono necessari se si configura il servizio shibd in modo da forzare l'utilizzo dello schema https nelle chiamate al SP come vedremo di seguito.

Per maggiori informazioni fare riferimento alla documentazione:

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApacheConfig#NativeSPApacheConfig-PreppingApache

Configurazione del servizio shibd

Le principali configurazioni di Shibboleth SP sono nel file *shibboleth2.xml*. Per una configurazione classica del SP non è necessario apportare molte modifiche al file predefinito che viene fornito con l'installazione di Shibboleth SP. Di seguito è riportata una parte del file *shibboleth2.xml* in cui vengono evidenziate le parti che è necessario modificare per una corretta configurazione.

```
<ApplicationDefaults
      entityID="http://nome.host/shibboleth-sp"
      REMOTE_USER="eppn persistent-id targeted-id uid cf">

        <Sessions lifetime="28800"
            timeout="3600" relayState="ss:mem"
            checkAddress="false" consistentAddress="false"
            handlerSSL="true" cookieProps="https" >

            <SSO entityID="https://ssoshib.comune.re.it/idp/shibboleth">
                 SAML2 SAML1
            </SSO>

            <Logout asynchronous="false">SAML2 Local</Logout>
             ...

        </Sessions>

        <MetadataProvider type="XML" file="idp-metadata.xml"/>

            ...

        <!--
        <ApplicationOverride id="admin"
                  entityID="https://admin.example.org/shibboleth"/>
        -->

</ApplicationDefaults>
```

Innanzitutto è necessario impostare il nome, ovvero l'entity ID, del SP. Questo valore è definito dall'attributo `entityID` nell'elemento xml `ApplicationDefaults`.

L'attributo `REMOTE_USER` definisce come deve essere valorizzato l'identificativo dell'utente da passare all'applicazione. Il SP riceve dall'IdP un'asserzione SAML contenente una lista di attributi relativi all'utente autenticato. Il SP valorizzerà una variabile d'ambiente chiamata "REMOTE_USER" che potrà essere utilizzata dall'applicazione per identificare l'utente. Shibboleth SP prenderà il primo attributo non nullo fra quelli elencati nel valore dell'attributo

`REMOTE_USER`. Gli attributi elencati devono essere separati da uno spazio e sono identificati dall'id definito nel file *attribute-map.xml*.

Gli attributi handlerSSL e cookiesProps dell'elemento ApplicationDefaults impongono l'utilizzo dello schema https sia per le chiamate al SP sia per l'accesso al sito da proteggere. Notare che se il SP è dietro a un reverse proxy con interruzione della connessione sicura https è necessario configurare in Apache le direttive `ServerName` e `UseCanonicalName` come visto nella sezione precedente.

L'attributo `entityID` deve avere come valore l'entity ID dell'IdP del comune di Reggio Emilia:

https://ssoshib.comune.re.it/idp/shibboleth

In alternativa, in fase di test, è possibile utilizzare l'IdP di test:

https://ssoshibt.comune.re.it/idp/shibboleth

L'attributo `asynchronous="false"` serve per completare il logout su una pagina del SP: in modo predefinito il logout termina visualizzando una pagina dell'IdP che comunica il successo, o meno, del logout. Impostando l'attributo `asynchronous="false"` fa sì che, dopo aver eseguito il logout, l'IdP reindirizza il browser dell'utente al SP. In questo modo il SP può visualizzare una pagina predefinita dopo il logout. È possibile inoltre specificare al SP quale pagina visualizzare dopo il logout utilizzando il seguente link di logout:

https://nome_server_pubblico/Shibboleth.sso/Logout?return=<url pagina dopo il logout>

Infine è necessario configurare un elemento `MetadataProvider` per indicare la posizione dei metadata dell'IdP. É necessario specificare nell'attributo `file` il nome del file xml contenente i metadata. Il file dei metadata può essere posizionato, come in questo caso, nella stessa cartella del file *shibboleth2.xml*. È possibile scaricare i metadata dell'IdP dall'url:

https://ssoshib.comune.re.it/idp/shibboleth o

https://ssoshibt.comune.re.it/idp/shibboleth (IdP di test)

Configurazione del SP con passaggio attributi via AJP

Se la connessione tra Apache e l'application server è effettuata con protocollo AJP, ad esempio con la direttiva di Apache

```
ProxyPass ajp://application_server:8009/protected
```

per permettere il passaggio degli attributi dell'utente all'applicazione come variabili d'ambiente, è necessario aggiungere l'attributo `attributePrefix="AJP_"` nel file *shibboleth2.xml* come indicato di seguito:

```
<ApplicationDefaults
      entityID="http://nome.host/shibboleth-sp"
      REMOTE_USER="eppn persistent-id targeted-id cf"
      attributePrefix="AJP_" >

            ...
```

```
</ApplicationDefaults>
```

In questo caso però è necessario togliere la direttiva `ShibUseHeaders On` dalle configurazioni di Apache.

Configurare più di un Server Name (virtual host) con un solo SP

Poiché il sistema di autenticazione di Shibboleth si basa sui cookie del browser, è necessario che il nome pubblico del server dell'applicazione protetta sia lo stesso di quello che appare nei metadata del SP. In pratica, se su un server sono in esecuzione due applicazioni su due virtual host con nome del server differente, è necessario configurare il SP in modo da ottenere i metadata corretti per ogni virtual host.

In Apache è necessario aggiungere una direttiva `ShibRequestSetting applicationId` per ogni ulteriore virtual host oltre a quello predefinito:

```
<VirtualHost *:80>
    ServerName www.esempio.it
    <Location />
        AuthType shibboleth
        require valid-user
    </Location>
</VirtualHost>
<VirtualHost *:80>
    ServerName app1.esempio.it
    <Location />
        AuthType shibboleth
        require valid-user
        ShibRequestSetting applicationId app1
    </Location>
</VirtualHost>

<VirtualHost *:80>
    ServerName app2.esempio.it
    <Location />
        AuthType shibboleth
        require valid-user
        ShibRequestSetting applicationId app2
    </Location>
</VirtualHost>
```

Nel file shibboleth2.xml è necessario decommentare l'elemento ApplicationOverride nel seguente modo:

```
<ApplicationDefaults entityID="https://www.esempio.it/shibboleth-sp
            ...
        <ApplicationOverride id="app1"
                  entityID="https://app1.esempio.it/shibboleth-sp"/>

        <ApplicationOverride id="app2"
                  entityID="https://app2.esempio.it/shibboleth-sp"/>

</ApplicationDefaults>
```

Il valore dell'attributo id deve coincidere con il valore impostato nella direttiva `ShibRequestSetting applicationId` di Apache.

In casi come questo, al fine di proteggere tutte le applicazioni, è necessario fornire all'IdP i metadata di ogni virtual host. Nel caso in esempio è necessario inviare i metadata relativi a tre entity ID ottenibili con gli url:

http://www.esempio.it/Shibboleth.sso/Metadata

http://app1.esempio.it/Shibboleth.sso/Metadata

http://app2.esempio.it/Shibboleth.sso/Metadata

Mappatura degli attributi

Gli attributi inviati dall'IdP vengono mappati all'interno del SP tramite il file *attribute-map.xml*. Di seguito è riportata la parte di file che è necessario aggiungere per mappare correttamente gli attributi inviati dall'IdP.

```
<Attribute name="userid" id="uid">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="CodiceFiscale" id="cf">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="cognome" id="sn">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="nome" id="givenName">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="emailAddressPersonale" id="mail">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="trustLevel" id="trustLevel">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="policyLevel" id="policyLevel">
```

```
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="indirizzoResidenza" id="indirizzoResidenza">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="capResidenza" id="capResidenza">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="cittaResidenza" id="cittaResidenza">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="provinciaResidenza" id="provinciaResidenza">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="statoResidenza" id="statoResidenza">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="lavoro" id="lavoro">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="indirizzoDomicilio" id="indirizzoDomicilio">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="capDomicilio" id="capDomicilio">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="cittaDomicilio" id="cittaDomicilio">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="provinciaDomicilio" id="provinciaDomicilio">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="statoDomicilio" id="statoDomicilio">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="dataNascita" id="dataNascita">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="luogoNascita" id="luogoNascita">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="provinciaNascita" id="provinciaNascita">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
```

```
<Attribute name="sesso" id="sesso">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="telefono" id="telefono">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="cellulare" id="cellulare">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="emailAddress" id="emailAddress">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
<Attribute name="federaUserDomain" id="federaUserDomain">
      <AttributeDecoder xsi:type="StringAttributeDecoder"/>
</Attribute>
```

Note per lo sviluppo delle applicazioni

Recupero degli attributi dell'utente con Shibboleth SP

In un ambiente protetto da Shibboleth SP gli attributi dell'utente autenticato vengono passati dal server web all'applicazione o come header HTTP o come variabili d'ambiente. Il nome delle variabili d'ambiente è uguale all'id degli attributi definito nel file *attribute-map.xml* del SP.

Quando gli attributi vengono passati come header HTTP, al nome degli attributi viene anteposto il prefisso “HTTP_”. A seconda del linguaggio di programmazione utilizzato per le applicazioni web, talvolta è necessario richiamare gli attributi considerando il prefisso “HTTP_”.

Nel caso di applicazioni Java gli attributi possono essere recuperati nel seguente modo:

```
//passaggio attributi come variabili d'ambiente
request.getAttribute("cf")

//passaggio attributi come header HTTP
request.getHeader("cf")
```

quindi la servlet Java rimuove automaticamente il prefisso “HTTP_”.

Nel caso di PHP invece è necessario differenziare tra variabili d'ambiente e header HTTP:

```
$_SERVER["Shib-Identity-Provider"]
```

```
$_SERVER["HTTP_SHIB_IDENTITY_PROVIDER"]
```

Per maggiori informazioni riguardo al passaggio degli attributi:

https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeAccess#NativeSPAttributeAccess-Tool-SpecificExamples