Installare un Service Provider Shibboleth su Linux Debian Squeeze (OBSOLETO)

Le informazioni contenute in questa guida sono obsolete, supportano ancora il SAML v1 che è deprecato e si riferiscono ad una versione di Debian fuori supporto.

Per istruzioni aggiornate, fare riferimento a https://wiki.idem.garr.it/wiki/Guide tenendo in mente che la parte di metadati in caso di SP non federati ma collegati solo all'IdP di UniTS va scambiata con l'IdP di UniTS e non con la federazione IDEM

- sostituire ogni occorrenza di "*sp-test*.units.it" con il FQDN della vostra macchina o il nome del virtualhost.
- tutte le configurazioni sono per un **VirtualHost con ssl** , quindi la "s" di https e' sempre fondamentale in tutti i files di configurazione. Se non volete fornire il servizio https, ricordate che l'entity della macchina e' a tutti gli effetti l'unico modo che l'idp ha per sapere chi siete. Quindi sia nei certificati sia nei file di configurazione dovete configurare di conseguenza e comunicarlo.
- gli attributi rilasciati di default li trovate alla pagina dell'Identity Provider se avete bisogno di attributi particolari scrivete a idem@units.it per concordare.
- sincronizzate data e ora con il server ntp.units.it

Debian Squeeze da zero con pacchetti precompilati

Partiamo da una macchina Debian Squeeze appena installata da un netinst, se non e' la vostra condizione aggiustate di conseguenza il procedimento.

```
apt-get install openssl apache2 libshibsp4 libapache2-mod-shib2
```

Shibboleth Service Provider si aggancia ad apache "proteggendo" un URL con la direttiva <Location> o <Directory> o le altre per cui e' previsto il controllo di accesso. Per iniziare i test, prima di passare all'applicativo da shibbolettizzare, consiglio di proteggere una location /secure

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

Redirect seeother /shibboleth
```

```
https://sp-test.units.it/Shibboleth.sso/Metadata
```

che consiglio di popolare con il seguente file index.php, per avere un riscontro immediato degli attributi rilasciati dall'IdP:

index.php

```
<html><body><pre>
<?php print_r($_SERVER); ?>
</pre></body></html>
```

Suggerimento: In fase di aggancio con l'applicativo solitamente si proteggono gli script di login o la location generica /Shibboleth.sso, ma rimando alla documentazione del vostro applicativo per il dettaglio.

Abilitare il modulo ssl

```
a2enmod ssl
```

Attenzione: l'attivazione del modulo NON implica l'attivazione dell'SSL, che deve essere fatta manualmente realizzando un link simbolico:

```
ln -s /etc/apache2/sites-available/default-ssl /etc/apache2/sites-
enabled/000-default-ssl
```

NOTA Se si vogliono utilizzare diversi virtualhosts bisogna invece creare un file che inizi con

```
NameVirtualHost *:443
```

e racchiuda i vari virtualhost. Attenzione che alcuni browser (es. alcune versioni di Internet Explorer) non supportano il protocollo SNI e quindi danno errore di certificato. Si consiglia quindi di attribuire un diverso IP per ciascun virtual host.

Nel caso in cui si stia utilizzando un certificato rilasciato da Terena, ricordarsi di impartire ad apache la direttiva per erogare la CA-Chain che si può trovare su:

```
wget https://ca.garr.it/mgt/Terena-chain.pem
```

e quindi indicare nel file di configurazione (verificare il percorso in cui sta il file del certificato):

```
SSLCertificateChainFile /usr/local/apache/conf/ssl.crt/Terena-chain.pem
```

Abilitare il modulo shib2 per Apache2

```
a2enmod shib2
```

Riavviare Apache

```
/etc/init.d/apache2 restart
```

Ci spostiamo in ***/etc/shibboleth*** e generiamo i certificati auto firmati, che saranno usati per la firma delle comunicazioni tra idp e sp.
La “y” definisce la durata dei certificati in anni. La “e” sta' per entity deve essere la stessa che piu' sotto andrete ad inserire nel file shibboleth2.xml.

```
/usr/sbin/shib-keygen -h sp-test.units.it -y 3 -e
https://sp-test.units.it/shibboleth
```

Ora generiamo i metadati che contentono informazioni per l'idp.

```
/usr/bin/shib-metagen -c sp-cert.pem -h sp-test.units.it -e
https://sp-test.units.it/shibboleth > sp-test-metadata.xml
```

NOTA Se state usando dei vhost apache dovete specificarli con l'opzione -h ad esempio

```
/usr/bin/shib-metagen -c sp-cert.pem -h sp-test.units.it -h sp-
test1.units.it -e https://sp-test.units.it/shibboleth > sp-test-metadata.xml
```

Alla fine del file generato, tra i tag di chiusura md:SPSSODescriptor e md:EntityDescriptor inserire le informazioni relative all'organizzazione e al vostro contatto tecnico:

```
  </md:SPSSODescriptor>

  <Organization>
    <OrganizationName xml:lang="it">University of Trieste</OrganizationName>
    <OrganizationDisplayName xml:lang="it">Universita' degli Studi di
Trieste</OrganizationDisplayName>
    <OrganizationURL xml:lang="it">http://www.units.it/</OrganizationURL>
  </Organization>
  <ContactPerson contactType="technical">
    <GivenName>NOME</GivenName>
    <SurName>COGNOME</SurName>
    <EmailAddress>EMAIL@units.it</EmailAddress>
  </ContactPerson>

</md:EntityDescriptor>
```

spedite questi metadati a idem@units.it e riceverete istruzioni sulle specifiche dell'ambiente di test.

cosi' che i metadati generati conterranno le AssertionConsumerService per tutti i vhost che volete shibboletizzare. Contestualmente il file shibboleth2.xml dovra' essere un po' diverso da quello riportato, agguingendo la definizione dei diversi host dentro il requestmap

```
<RequestMapper type="Native">
        <RequestMap applicationId="default">
          <Host name="sp-test.units.it"></Host>
          <Host name="sp-test1.units.it"></Host>
        </RequestMap>
     </RequestMapper>
```

IMPORTANTE: SE usate una entity diversa da quella suggerita specificate nell'email sia l'entity sia l'URL dal quale si possano recuperare i metadata del vostro SP.

Scaricate i metadata dell'IdP

```
wget https://idemfero.units.it/idp/shibboleth -O
/var/run/shibboleth/idemfero-metadata.xml
```

Allego il file di configurazione di shibboleth Service Provider nel quale si danno quasi tutte le direttive per creare il canale di comunicazione con l'idp.

shibboleth2.xml

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    logger="syslog.logger" clockSkew="180">

    <OutOfProcess logger="shibd.logger">
    </OutOfProcess>
    <InProcess logger="native.logger">
    </InProcess>

    <UnixListener address="shibd.sock"/>
    <StorageService type="Memory" id="mem" cleanupInterval="900"/>
    <SessionCache type="StorageService" StorageService="mem"
cacheTimeout="3600" inprocTimeout="900" cleanupInterval="900"/>
    <ReplayCache StorageService="mem"/>
    <ArtifactMap artifactTTL="180"/>
    <RequestMapper type="Native">
        <RequestMap applicationId="default">
            <Host name="sp-test.units.it">
            </Host>
        </RequestMap>
    </RequestMapper>

    <ApplicationDefaults id="default" policyId="default"
        entityID="https://sp-test.units.it/shibboleth"
        REMOTE_USER="eppn persistent-id targeted-id"
        signing="false" encryption="false">

        <Sessions lifetime="28800" timeout="3600" checkAddress="false"
            handlerURL="/Shibboleth.sso" handlerSSL="false"
exportLocation="http://localhost/Shibboleth.sso/GetAssertion"
exportACL="127.0.0.1"
            idpHistory="false" idpHistoryDays="7">

            <SessionInitiator type="Chaining" Location="/Login"
isDefault="true" id="Intranet"
```

```
                    relayState="cookie"
entityID="https://idemfero.units.it/idp/shibboleth">
                <SessionInitiator type="SAML2" acsIndex="1"
template="bindingTemplate.html"/>
                <SessionInitiator type="Shib1" acsIndex="5"/>
            </SessionInitiator>

            <md:AssertionConsumerService Location="/SAML2/POST"
index="1"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"/>
            <md:AssertionConsumerService Location="/SAML2/POST-
SimpleSign" index="2"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST-SimpleSign"/>
            <md:AssertionConsumerService Location="/SAML2/Artifact"
index="3"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact"/>
            <md:AssertionConsumerService Location="/SAML2/ECP"
index="4"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:PAOS"/>
            <md:AssertionConsumerService Location="/SAML/POST"
index="5"
                Binding="urn:oasis:names:tc:SAML:1.0:profiles:browser-
post"/>
            <md:AssertionConsumerService Location="/SAML/Artifact"
index="6"
Binding="urn:oasis:names:tc:SAML:1.0:profiles:artifact-01"/>
                            <LogoutInitiator type="Chaining"
Location="/Logout" relayState="cookie">
                <LogoutInitiator type="SAML2"
template="bindingTemplate.html"/>
                <LogoutInitiator type="Local"/>
            </LogoutInitiator>

            <md:SingleLogoutService Location="/SLO/SOAP"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>
            <md:SingleLogoutService Location="/SLO/Redirect"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Redirect"/>
            <md:SingleLogoutService Location="/SLO/POST"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
POST"/>
            <md:SingleLogoutService Location="/SLO/Artifact"
conf:template="bindingTemplate.html"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-
Artifact"/>
```

```
            <md:ArtifactResolutionService Location="/Artifact/SOAP"
index="1"
                Binding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"/>

            <Handler type="MetadataGenerator" Location="/Metadata"
signing="false"/>
            <Handler type="Status" Location="/Status" acl="127.0.0.1"/>
            <Handler type="Session" Location="/Session"
showAttributeValues="false"/>

        </Sessions>

        <Errors supportContact="root@localhost"
            logoLocation="/shibboleth-sp/logo.jpg"
            styleSheet="/shibboleth-sp/main.css"/>

         <MetadataProvider type="XML"
uri="https://idemfero.units.it/idp/shibboleth"
              backingFilePath="idemfero-metadata.xml"
reloadInterval="7200">
         </MetadataProvider>

        <TrustEngine type="Chaining">
            <TrustEngine type="ExplicitKey"/>
            <TrustEngine type="PKIX"/>
        </TrustEngine>

        <AttributeExtractor type="XML" validate="true" path="attribute-
map.xml"/>
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path="attribute-
policy.xml"/>
        <CredentialResolver type="File" key="sp-key.pem"
certificate="sp-cert.pem"/>
    </ApplicationDefaults>

    <SecurityPolicies>
        <Policy id="default" validate="false">
            <PolicyRule type="MessageFlow" checkReplay="true"
expires="60"/>
            <PolicyRule type="Conditions">
                <PolicyRule type="Audience"/>
            </PolicyRule>
            <PolicyRule type="ClientCertAuth" errorFatal="true"/>
            <PolicyRule type="XMLSigning" errorFatal="true"/>
            <PolicyRule type="SimpleSigning" errorFatal="true"/>
        </Policy>
    </SecurityPolicies>

</SPConfig>
```

I cambiamenti rispetto al file dell'installazione originale dal pacchetto precompilato consistono in

- tolto i riferimenti alle ISAPI
- tolto il request mapper per la location secure
- tolto NIM
- rediretto il SessionInitiator all'entity del nostro idp entityID="https://idemfero.units.it/idp/shibboleth"
- aggiunto il metadataprovider statico verso l'idp idemfero.units.it

Riavviate il demone shibd

```
/etc/init.d/shibd restart
```

ATTENZIONE alla sincronizzazione dell'ora di sistema. Usare il server ntp.

E' possibile regolare la durata della sessione lato SP.

Apache e PHP con Suhosin patch

Installato di default su Debian, rifiuta i cookie troppo lunghi di shibboleth:

```
suhosin[1298]: ALERT - configured request variable name length limit
exceeded - dropped variable
'_shibsession_64656661756c7468747470733a2f2f7574736c6f676765722e756e6974732e
69742f73686962626f6c657468' (attacker '192.168.1.2', file '/var/www/....')
```

In tal caso di legga questo thread.

Aumentare quindi i valori in /etc/php5/conf.d/suhosin.ini a:

```
suhosin.cookie.max_name_length = 512
suhosin.request.max_varname_length = 512
```

Debian Squeeze da zero da sorgenti

Quello che segue e' uno script bash non molto performante che dopo una decina di minuti vi restituira' una macchina con un Service Provider 2.4.2 pronto per essere usato. Va sistemato. Le config alle quali si fa riferimento appena trovo dopo mettere le carico sul server.

install.sh

```
#!/bin/bash

if [ -z $1 ]
 then echo "ERROR: FQDN Service Provider Needed"
      echo "EXAMPLE: ./install.sh sptest.unitest.it"
```

```
      echo ""
      exit;
fi

if [ $0 = './install.sh' ]
 then   export HOME_INSTALL=`pwd`
 else export HOME_INSTALL=`dirname $0`
fi

#export $HOME_INSTALL
export SHIB_HOME=/opt/shibboleth-sp-2.4.3
export MYBUILD=/opt/shibsp2.4.3-build

mkdir $SHIB_HOME
mkdir $MYBUILD
mkdir /etc/shibboleth/

read -p "Press Enter to install ntp ..."
apt-get -y install ntp
read -p "Press Enter to install gcc g++ make ..."
apt-get -y install gcc g++ make
read -p "Press Enter to install apache2 openssl ..."
apt-get -y install apache2
apt-get -y install openssl
read -p "Press Enter to install libssl0.9.8 libssl-dev ..."
apt-get -y install libssl0.9.8 libssl-dev
read -p "Press Enter to install libcurl3 libcurl3-dev ..."
apt-get -y install libcurl3 libcurl3-dev
read -p "Press Enter to install libxerces-c3.1 libxerces-c-dev ..."
apt-get -y install libxerces-c3.1 libxerces-c-dev
read -p "Press Enter to install apache2-threaded-dev ..."
apt-get -y install apache2-threaded-dev


echo "Donwload Time"
echo "wget required"
read -p "Press Enter to download  ..."

wget
http://www.shibboleth.net/downloads/log4shib/1.0.4/log4shib-1.0.4.tar.g
z -P $MYBUILD
wget http://www.shibboleth.net/downloads/c++-
opensaml/2.4.3/xmltooling-1.4.2.tar.gz -P $MYBUILD
wget http://www.shibboleth.net/downloads/c++-
opensaml/2.4.3/opensaml-2.4.3.tar.gz -P $MYBUILD
wget http://www.shibboleth.net/downloads/service-
provider/latest/shibboleth-sp-2.4.3.tar.gz -P $MYBUILD
wget http://mirror.switch.ch/mirror/apache/dist/santuario/c-
library/xml-security-c-1.6.1.tar.gz -P $MYBUILD

for f in $MYBUILD/*.tar.gz; do tar -xzvf $f -C $MYBUILD; done
```

```
echo "Compiling Time"

read -p "Press Enter to install log4shib ..."
cd $MYBUILD/log4shib-1.0.4/
./configure --prefix=$SHIB_HOME --disable-static --disable-doxygen
make
make install

read -p "Press Enter to install xml-security-c-1.6.1 ..."
cd $MYBUILD/xml-security-c-1.6.1/
./configure --prefix=$SHIB_HOME
make
make install

read -p "Press Enter to install xmltooling-1.4.2 ..."
cd $MYBUILD/xmltooling-1.4.2/
./configure --prefix=$SHIB_HOME --with-log4shib=$SHIB_HOME --with-
xmlsec=$SHIB_HOME -C
make
make install

read -p "Press Enter to install opensaml-2.4.3 ..."
cd $MYBUILD/opensaml-2.4.3/
./configure --prefix=$SHIB_HOME --with-log4shib=$SHIB_HOME -C
make
make install

read -p "Press Enter to install shibboleth-2.4.3 ..."
cd $MYBUILD/shibboleth-2.4.3/
./configure --prefix=$SHIB_HOME --enable-apache-22 --with-
log4shib=$SHIB_HOME --with-xmltooling=$SHIB_HOME --with-saml=$SHIB_HOME
-C
make
make install

ln -s /opt/shibboleth-sp-2.4.3/ /opt/shibboleth-sp

cp /$HOME_INSTALL/configs/shibd /etc/init.d/

chmod +x /etc/init.d/shibd
update-rc.d shibd defaults

cp $SHIB_HOME/etc/shibboleth/protocols.xml /etc/shibboleth/
cp $SHIB_HOME/etc/shibboleth/security-policy.xml /etc/shibboleth/
cp $SHIB_HOME/etc/shibboleth/native.logger /etc/shibboleth/
cp $SHIB_HOME/etc/shibboleth/shibd.logger /etc/shibboleth/
cp $SHIB_HOME/etc/shibboleth/syslog.logger /etc/shibboleth/
mkdir -p /var/log/shibboleth/
touch /var/log/shibboleth/shibd.log
touch /var/log/shibboleth/native.log
```

```
chgrp www-data /var/log/shibboleth/native.log
chmod g+w /var/log/shibboleth/native.log

cd /etc/shibboleth/

echo "Certificate Creation Time"
read -p "Press Enter to continue ..."

sh $SHIB_HOME/etc/shibboleth/keygen.sh -h $1 -y 3 -e
https://$1/shibboleth

#cp $HOME_INSTALL/configs/native.logger /etc/shibboleth/
#cp $HOME_INSTALL/configs/shibd.logger /etc/shibboleth/


/etc/init.d/apache2 stop

echo "Copying Configs Time ...."
read -p "Press Enter to continue ...."

cp $HOME_INSTALL/configs/shib.load /etc/apache2/mods-available/
cp $HOME_INSTALL/configs/shib.conf /etc/apache2/mods-available/
cp $HOME_INSTALL/configs/envvars /etc/apache2/
a2enmod shib

mv /etc/apache2/sites-available/default-ssl /etc/apache2/sites-
available/default-ssl.BAK
cp $HOME_INSTALL/configs/default-ssl /etc/apache2/sites-available

a2ensite default-ssl
a2enmod ssl

echo "++++++++ ATTENTION +++++++++"
echo ""
echo ""
echo "Before restarting Apache make sure you have a shibboleth2.xml in
/etc/shibboleth"
echo ""
echo ""
echo ""
```

shib.conf

```
# Global Configuration
# This is the XML file that contains all the global, non-apache-
specific
# configuration.  Look at this file for most of your configuration
parameters.
ShibConfig /etc/shibboleth/shibboleth2.xml
```

```
# Used for example logo and style sheet in error templates.
<IfModule mod_alias.c>
  <Location /shibboleth-sp>
    Allow from all
  </Location>
  Alias /shibboleth-sp/main.css /opt/shibboleth-
sp/share/doc/shibboleth/main.css
  Alias /shibboleth-sp/logo.jpg /opt/shibboleth-
sp/share/doc/shibboleth/logo.jpg
</IfModule>
```

shib.load

```
LoadModule mod_shib /opt/shibboleth-sp/lib/shibboleth/mod_shib_22.so
```

Nel file shibboleth2.xml moltissimi elementi hanno valori standard e ammenoche' non si voglia una configurazione particolare, gli SP 2.4.2 hanno automatizzato molto riducendo il file di configurazione notevolmente. Ovviamente gli SP 2.4.2 mantengono anche tutte le specifiche dei precendeti.

shibboleth2.xml

```
<SPConfig xmlns="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:conf="urn:mace:shibboleth:2.0:native:sp:config"
    xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
    xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
    xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata"
    clockSkew="180">
    <ApplicationDefaults
entityID="<del>https://sp-test.units.it/shibboleth</del>"
                          REMOTE_USER="eppn persistent-id targeted-id">
        <Sessions lifetime="28800" timeout="3600" checkAddress="false"
relayState="ss:mem" handlerSSL="false">
            <SSO entityID="https://idemfero.units.it/idp/shibboleth" >
SAML1 SAML2 </SSO>
            <Logout>SAML2 Local</Logout>
            <Handler type="MetadataGenerator" Location="/Metadata"
signing="false">
                <Organization>
                <OrganizationName xml:lang="it">Universita' degli Studi
di Trieste - TEST</OrganizationName>
                <OrganizationDisplayName xml:lang="it">Universita'
degli Studi di Trieste</OrganizationDisplayName>
                <OrganizationURL
xml:lang="it">http://www.units.it/</OrganizationURL></Organization>
                <ContactPerson contactType="technical">
                <GivenName>Staff Tecnico</GivenName>
                <EmailAddress>idem@units.it</EmailAddress>
                </ContactPerson>
            </Handler>
```

```
            <Handler type="Status" Location="/Status" acl="127.0.0.1"/>
            <Handler type="Session" Location="/Session"
showAttributeValues="false"/>
            <Handler type="DiscoveryFeed" Location="/DiscoFeed"/>
        </Sessions>
        <Errors supportContact="idem@units.it"
            logoLocation="/shibboleth-sp/logo.jpg"
            styleSheet="/shibboleth-sp/main.css"/>
        <MetadataProvider type="XML"
uri="https://idemfero.units.it/idp/shibboleth"
              backingFilePath="idemfero-metadata.xml"
reloadInterval="7200">
        </MetadataProvider>
                <AttributeExtractor type="XML" validate="true"
path="attribute-map.xml"/>
        <AttributeResolver type="Query" subjectMatch="true"/>
        <AttributeFilter type="XML" validate="true" path="attribute-
policy.xml"/>
        <CredentialResolver type="File" key="sp-key.pem"
certificate="sp-cert.pem"/>
    </ApplicationDefaults>
    <SecurityPolicyProvider type="XML" validate="true" path="security-
policy.xml"/>
    <ProtocolProvider type="XML" validate="true" reloadChanges="false"
path="protocols.xml"/>
</SPConfig>
```

Configurazioni particolari per uso interno UniTs

Una volta verificato il funzionamento, è possibile limitare l'accesso a utenti e gruppi AD:

Modificare */etc/shibboleth/attribute-map.xml* aggiungendo gli attributi che si vogliono matchare e passare come variabili d'ambiente ad Apache:

```
<Attribute name="urn:mace:dir:attribute-def:accountName" id="accountName"/>
<Attribute name="urn:mace:dir:attribute-def:adGroup" id="adGroup"/>
```

Nella configurazione di Apache operare le modifiche necessarie:

```
<Directory /var/www>
    AuthType shibboleth
    ShibRequireSession On
    require user 5698 1235 s125896
    require accountName 8584
    require adGroup CN=C_451,OU=gruppi,OU=studenti,DC=ds,DC=units,DC=it
</Directory>
```

L'esempio qui sopra permette l'accesso ai dipendenti con matricola 5698 1235 8584, allo studente

con identificativo s125896 e a tutti gli appartenenti al corso di laurea C_451.
Notare l'uso della direttinva “require user”. Identica a quella usata con altri tipi di autorizzazione, è rimappata sull'attributo SAML accountName tramite la direttiva

```
    <ApplicationDefaults id="default" policyId="default"
        entityID="https://sp-test.units.it/shibboleth"
        REMOTE_USER="accountName"
        signing="false" encryption="false">
```

contenuta nel file /etc/shibboleth/shibboleth2.xml

Gli attributi SAML accountName e adGroup, per poter essere usati per l'autorizzazione e passati all'eventuale applicazione php o cgi, necessitano di essere definiti nel file /etc/shibboleth/attribute-map.xml

```
    <!-- Units attributes -->
    <Attribute name="urn:mace:dir:attribute-def:accountName"
id="accountName">
       <AttributeDecoder xsi:type="StringAttributeDecoder"/>
    </Attribute>
    <Attribute name="urn:mace:dir:attribute-def:adGroup" id="adGroup">
       <AttributeDecoder xsi:type="StringAttributeDecoder"/>
    </Attribute>
    <!-- / Units attributes -->
```

From:
https://docu.units.it/dokuwiki/ - **Area dei Servizi ICT - Documentation**

Permanent link:
https://docu.units.it/dokuwiki/gestione-server:idemsso:sp:shibsp

Last update: **2021/11/25 08:50 (5 anni fa)**