Lecture no. 1

Shibboleth SP: installation and basic configuration for Single Sign On (SSO) first part

Andrea Biancini – Consortium GARR
Outline

• VM checking – operational instructions
• Introduction on Service Providers
• Installation
• Configuration
• Protection of a web service
Hands-on

- Open a web page
  - Use your own or use the one on the VM provided
- Configure Shibboleth/Apache/etc
  - Copy files using the terminal (\texttt{sudo -i})
  - Edit files with vi/gedit/emacs/atom/etc
- Material for the configuration is in
  - \texttt{/home/testuser/SPCourse/01_SESSION}
Introduction to Shibboleth Service Provider

- Flow of information
- Trust relationships
- SAML Framework
Flow of information

1. The SP detects the user attempting to **access restricted** content within the resource.

2. The SP generates an **authentication request**, then sends the request, and the user, to the user’s IdP.

3. The IdP **authenticates** the user, then sends the **authentication response**, and the user, back to the SP.

4. The **SP verifies the IdP's response** and sends the request through to the resource.

https://wiki.shibboleth.net/confluence/display/SHIB2/NewUnderstandingShibboleth
Trust Relationship

- In legacy applications, the trust relationship is implicit
  - All functions are internal to the application itself
- In the Federated paradigm with SP and IdP, there has to be an explicit trust relationship between IdP and SP
- This mutual trust is created by exchanging metadata and digital certificates
<EntitiesDescriptor ID="IDEM">
    .......
    <ds:X509Certificate>
        MIIDnzCCAoeg......
    </ds:X509Certificate>
</EntitiesDescriptor>

<EntityDescriptor entityID="https://IDP.example.it">
    .......
    <ds:X509Certificate>
        MIIDEzCCAfugAwIBAg......
    </ds:X509Certificate>
</EntityDescriptor>

<EntityDescriptor entityID="https://SP.example.it">
    .......
    <ds:X509Certificate>
        MQswCQY......
    </ds:X509Certificate>
</EntityDescriptor>
Trust Relationship - Assertions

<saml2p:Response Destination="https://filesender.garr.it" .....>
<saml2:Issuer>https://idp.uniparthenope.it</saml2:Issuer>
<ds:Signature...>
....
<ds:X509Certificate>
  MIIDOzCCAiO...
</ds:X509Certificate>

<saml2:EncryptedAssertion>
......
<ds:X509Certificate>
  MIEZTCCA02gAw....
</ds:X509Certificate>

<xenc:CipherValue>
UKuw3E7+YXdLrVUFjy+eFFf5aQRDjdAvdiYNZd51823d
YDo734pJiaKSWQgfZLlxevHVC5SwwLK2axe6rZFZpm/a
........
SAML Framework

- **Shibboleth Service Provider** - An Apache/IIS module which can be used in any language supported by the webserver

- **SimpleSAMLphp** - A native PHP implementation which can be directly integrated in our application

- Full list of software:
SAML Extensions

- Many widely-known applications have SAML extensions available

- https://wiki.shibboleth.net/confluence/display/SHIB2/ShibEnabled
- https://rnd.feide.no/fed_software/

Andrea Biancini – Consortium GARR
Shibboleth SP

- Shibboleth Service Provider is composed of
  - mod_shib (Apache/IIS)
  - Shibd daemon

- Characteristics:
  - Protect access using “require”
  - User attributes are accessible in the web server environment, which can then be used by applications (php, perl, .NET, ASP, CGI) e.g. $_SERVER['mail']
  - Servlet containers, e.g. tomcat, must use Apache or IIS as front-end.
**SP Installation**

- Debian/Ubuntu packages:
  ```
  sudo apt-get install apache2 libapache2-mod-shib2 openssl php5 ntp
  ```

- RPM-based distros:
  ```
  http://download.opensuse.org/repositories/security:/shibboleth/
  ```
Directory structure

- **Apache module:**
  /etc/apache2/mod-available/shib2.load
  /usr/lib/apache2/modules/mod_shib2.so

- **Daemon:**
  /usr/bin/
  /usr/lib/x86_64-linux-gnu/shibboleth/*
  /usr/sbin/shib-keygen
  /usr/sbin/shibd

- **Configuration:**
  /etc/shibboleth/*

- **Installed files:**
  - **Deb – Use** `dpkg -L libapache2-mod-shib2`
  - **RPM – See** [http://packages.ubuntu.com/trusty/amd64/libapache2-mod-shib2/filelist](http://packages.ubuntu.com/trusty/amd64/libapache2-mod-shib2/filelist)
SP configuration

- Certificate configuration
  - Download the certificate to check metadata
  - Generate our SP certificate

- Note: the certificate use by the web server for https traffic is completely independent this is only for the shibboleth web service
Certificate configuration

- Save the certificates **to check** the metadata file
- IDEM federation:

- For this course:
  - `wget https://sp.lab.unimo.it/metadata-signer.crt -O /etc/shibboleth/metadata-signer.crt`
Create the signature and encryption assertions

- Generate the certificates:
  - `/usr/sbin/shib-keygen`
  - `/etc/shibboleth/sp-cert.pem`
  - `/etc/shibboleth/sp-key.pem`

- Just for this course:
  - `cp` `/home/testuser/SPCourse/01_SESSION/shibboleth/sp-* .pem /etc/shibboleth`
Shibd configuration

- Set `entityID`
- Set `SSO`
- Set `metadata provider`
Configuring shibboleth2.xml (1/3)

- Edit `/etc/shibboleth/shibboleth2.xml`.

  **entityID**

- Replace
  ```xml
  <ApplicationDefaults
  entityID="https://sp.example.org/shibboleth"
  REMOTE_USER="eppn persistent-id targeted-id"/>
  ```

- With
  ```xml
  <ApplicationDefaults entityID="https://sp1.local/shibboleth"
  REMOTE_USER="eppn persistent-id targeted-id"/>
  ```
Configuring Shibboleth2.xml (2/3)

- Edit /etc/shibboleth/shibboleth2.xml

**SSO (case for a single IdP)**

- Replace
  ```xml
  <SSO entityID="https://idp.example.org/shibboleth"
  discoveryProtocol="SAMLDS"
  discoveryURL="https://ds.example.org/DS/WAYF">
  SAML2 SAML1
  </SSO>
  ```

- With
  ```xml
  <SSO entityID="https://idp-corso.irccs.garr.it/idp/shibboleth"
  discoveryProtocol="SAMLDS"
  discoveryURL="https://ds.example.org/DS/WAYF"> SAML2 SAML1
  </SSO>
  ```
Configuring Shibboleth2.xml (3/3)

- Edit `/etc/shibboleth/shibboleth2.xml`

  ```xml
  <MetadataProvider type="XML" uri="https://sp.lab.unimo.it/rr3/signedmetadata/federation/fed-corso/metadata.xml"
  backingFilePath="/etc/shibboleth/signed-test-metadata.xml" reloadInterval="7200">
  <MetadataFilter type="Signature" certificate="metadata-signer.crt"/>
  </MetadataProvider>
  ```
Configuration - check and start

- Check the configuration:
  
  ```
  shibd -t
  ```

- Restart the service:
  
  ```
  service shibd restart
  ```

- Activate the shib2 module
  
  - Deb-based: `a2enmod shib2`
  
  - RPM-based: `service apache2 restart`
Our Metadata

● Let's take a look at our metadata:
  --https://sp1.local/Shibboleth.sso/Metatada

● Check the various tags:
  <Request Initiator>
  <Assertion consumer service>
Protecting a web resource
Protection of a local resource

- The page to protect:
  - https://sp1.local/intranet/intranet.html

Open access
Apache Configuration

- Configuration of the location to protect:
  - **Modify** /etc/apache2/sites-available/service_provider.conf

```xml
<Location /intranet>
  AuthType shibboleth
  ShibRequestSetting requireSession true
  Require shib-session
</Location>
```

- **Activate shib2 module:**
  `service apache2 restart`
Protecting a local resource

- Check:
  - [https://sp1.local/intranet/intranet.html](https://sp1.local/intranet/intranet.html)

**Access with authentication**
Prepare for the next phase

- Run the script to synchronise all the VMs

```bash
cd /home/testuser
./SPCourse/update1.sh
```
Any questions?