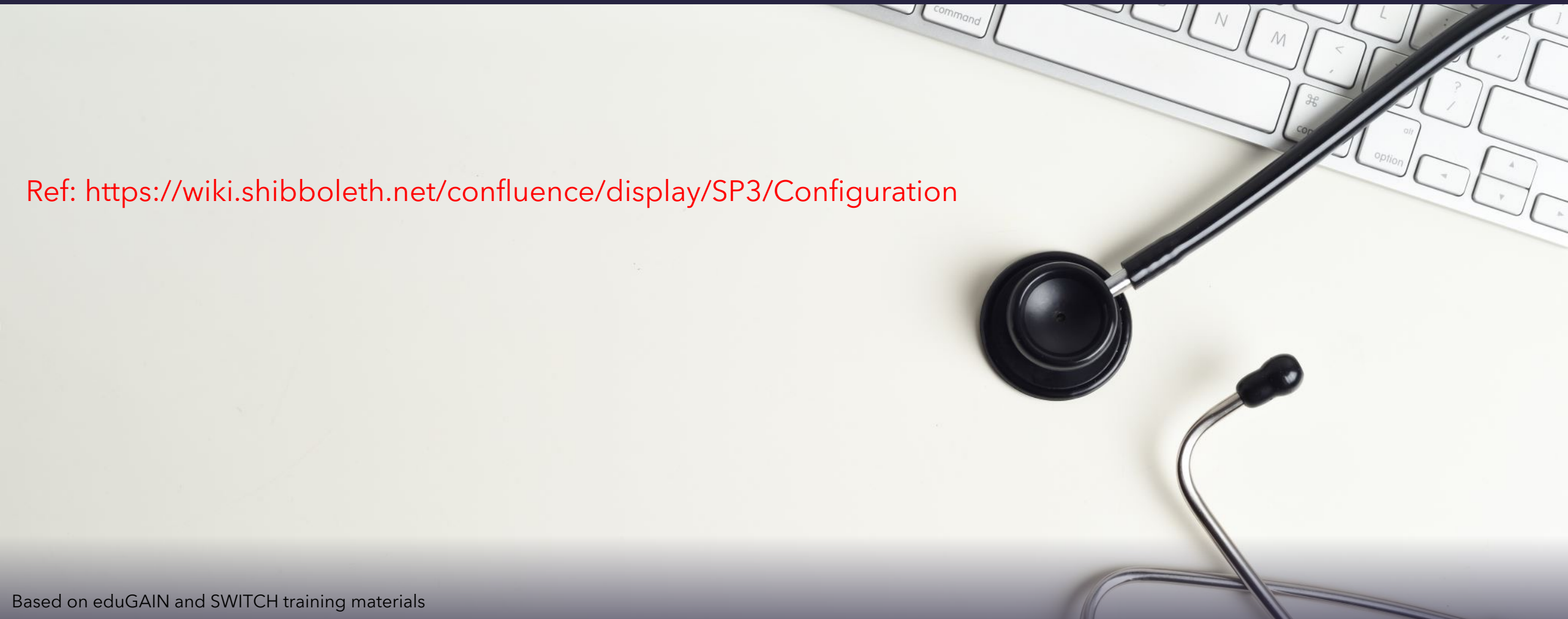
Shibboleth Service Provider

Configuration explanations

LEARN Identity Access Federation (LIAF)



Ref: https://wiki.shibboleth.net/confluence/display/SP3/Configuration

Based on eduGAIN and SWITCH training materials

Important locations

- **/etc/shibboleth/**
  Master and supporting configuration files
  Locally maintained metadata files
  HTML templates (to customize the look & feel of service)
  Logging configuration files (*.logger)
  Credentials (certificates and private keys)
- **/var/run/shibboleth/** *and* **/var/cache/shibboleth/**
  UNIX socket and remote metadata backups
- **/var/log/shibboleth/**
  shibd.log and transaction.log files
- **/var/log/apache2/** *or* **/var/log/shibboleth/apache2/**
  location for apache access and error logs

/etc/shibboleth/shibboleth2.xml

```
<ApplicationDefaults entityID="https://sp.YOUR-DOMAIN/shibboleth"
        REMOTE_USER="eppn subject-id pairwise-id persistent-id"
        cipherSuites="DEFAULT:!EXP:!LOW:!aNULL:!eNULL:!DES:!IDEA:!SEED:!RC4:!3DES:!kRSA:!SSLv2:!SSLv3:!TLSv1:!TLSv1.1">
```

Every SP needs an unique identifier: The entityID

Where is entityID used?
In transmitted messages, local configuration, metadata
IdP log files, configuration, filtering policies

EntityID should be: Unique, locally scoped, representative and unchanging

Convention: Include FQDN of your service: `https://sp.example.org/shibboleth`

In case your application needs to have a remote user for authentication, you just could make shibboleth put an attribute (e.g. ”eppn") as REMOTE_USER. In above example, if eppn attribute is available, it will be put into REMOTE_USER

Attribute eppn has precedence over persistent-id in this case

/etc/shibboleth/shibboleth2.xml

```
<SSO entityID="https://idp.example.org/idp/shibboleth">
  SAML2
</SSO>
```

Option 1

```
<SSO discoveryProtocol="SAMLDS" discoveryURL="https://fds.ac.lk">
   SAML2
</SSO>
```

Option 2

```
<SSO discoveryProtocol="SAMLDS" discoveryURL="https://service.seamlessaccess.org/ds/">
  SAML2
</SSO>
```

Option 3

Option 1: Using a single IDP as the login source
Option 2: Using LIAF Discovery Service to select the IDP
Option 3: Using seamlessaccess DS to open your service to eduGAIN

/etc/shibboleth/shibboleth2.xml

```
<MetadataProvider type="XML" url="https://fr.ac.lk/signedmetadata/metadata.xml"
          legacyOrgName="true" backingFilePath="liaf-metadata.xml" maxRefreshDelay="86400">
      <MetadataFilter type="Signature" certificate="federation-cert.pem" verifyBackup="false"/>
      <MetadataFilter type="RequireValidUntil" maxValidityInterval="864000" />
</MetadataProvider>
```

Option 1

```
<MetadataProvider type="XML" validate="true"
          url="https://fr.ac.lk/signedmetadata/LIAF-interfederation-idp-metadata.xml"
          backingFilePath="interfederation-metadata.xml" rmaxRefreshDelay="86400">
        <MetadataFilter type="RequireValidUntil" maxValidityInterval="864000"/>
        <MetadataFilter type="Signature" certificate="federation-cert.pem" verifyBackup="false"/>
</MetadataProvider>
```

Option 2

```
    <!-- Example of locally maintained metadata. -->
<MetadataProvider type="XML" validate="true" path="partner-metadata.xml"/>
```

Option 3

Option 1: Using LIAF Metadata (SP allowed only for LIAF)
Option 2: Using LIAF interfederation metadata (Opening SP for eduGAIN)
Option 3: Using a pre-downloaded metadata file. (If used with a single IDP)

/etc/shibboleth/shibboleth2.xml

```
<!-- Simple file-based resolvers for separate signing/encryption keys. -->
<CredentialResolver type="File" use="signing"
      key="sp-signing-key.pem" certificate="sp-signing-cert.pem"/>
<CredentialResolver type="File" use="encryption"
      key="sp-encrypt-key.pem" certificate="sp-encrypt-cert.pem"/>
```

Certificate files used for signing and encrypting

Certificates are created with,

/usr/sbin/shib-keygen -n sp-signing -e https://sp.YOUR-DOMAIN/shibboleth

/usr/sbin/shib-keygen -n sp-encrypt -e https://sp.YOUR-DOMAIN/shibboleth

/etc/shibboleth/shibboleth2.xml

```
<ApplicationOverride id="vhost2" entityID="https://vhost2.Your-Domain/shibboleth">
        <CredentialResolver type="File" use="signing"
             key="vhost2-signing-key.pem" certificate="vhost2-signing-cert.pem"/>
          <CredentialResolver type="File" use="encryption"
             key="vhost2-encrypt-key.pem" certificate="vhost2-encrypt-cert.pem"/>
</ApplicationOverride>
```

Optionally, One or More <ApplicationOverride> helps us to override the default shibboleth configurations.

Above defines a new entity ID for a 2nd web virtualhost on the same server and certificate pairs for that new domain.

Override is identified by the id value. (in this case vhost2)

When hosting multiple virtualhosts in the same server, it is recommended to use different entity ID's and Certificates pairs. Also these entities should be registered with LIAF as separate SP's.

/etc/shibboleth/attribute-map.xml

```
<Attribute name="urn:oasis:names:tc:SAML:attribute:subject-id" id="subject-id">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>

<Attribute name="urn:oasis:names:tc:SAML:attribute:pairwise-id" id="pairwise-id">
    <AttributeDecoder xsi:type="ScopedAttributeDecoder" caseSensitive="false"/>
</Attribute>

<Attribute name="urn:mace:dir:attribute-def:cn" id="cn"/>
<Attribute name="urn:mace:dir:attribute-def:sn" id="sn"/>
<Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
<Attribute name="urn:mace:dir:attribute-def:displayName" id="displayName"/>
<Attribute name="urn:mace:dir:attribute-def:uid" id="uid"/>
<Attribute name="urn:mace:dir:attribute-def:mail" id="mail"/>
```

Example

Enable required attributes that needs to be passed from the authenticated IDP. (by Uncomment or Comment)

Apache VirtualHost config

```
# Force user to authenticate on protected-directory
<Location /protected-directory>
  AuthType shibboleth
  ShibRequestSetting requireSession true
  Require shibboleth
</Location>

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

On your IDP/S

To enable a perticular SP to work, your IDP should be aware of the SP.

If you allowed your SP on LIAF, Federation will push these settings to every IDP.

As an IDP admin, if you need to allow a specific SP, follow these:

/opt/shibboleth-idp/conf/metadata-providers.xml

```
<MetadataProvider
      id="HTTPMD-LEARN-Federation"
      xsi:type="FileBackedHTTPMetadataProvider"
      backingFile="%{idp.home}/metadata/test-metadata.xml"
      metadataURL="https://fr.ac.lk/signedmetadata/metadata.xml">

      <MetadataFilter xsi:type="SignatureValidation" requireSignedRoot="true" certificateFile="${idp.home}/metadata/federation-cert.pem"/>
      <MetadataFilter xsi:type="RequiredValidUntil" maxValidityInterval="P10D"/>
      <MetadataFilter xsi:type="EntityRoleWhiteList">
        <RetainedRole>md:SPSSODescriptor</RetainedRole>
      </MetadataFilter>
</MetadataProvider>
<MetadataProvider id="HTTPMD-LEARN-interfederation"
                xsi:type="FileBackedHTTPMetadataProvider"
                backingFile="%{idp.home}/metadata/LEARNmetadata.xml"
                metadataURL="https://fr.ac.lk/signedmetadata/LIAF-interfederation-sp-metadata.xml">

  <MetadataFilter xsi:type="SignatureValidation" requireSignedRoot="true" certificateFile="${idp.home}/metadata/federation-cert.pem"/>
  <MetadataFilter xsi:type="RequiredValidUntil" maxValidityInterval="P11D"/>
  <MetadataFilter xsi:type="EntityRoleWhiteList">
      <RetainedRole>md:SPSSODescriptor</RetainedRole>
  </MetadataFilter>
</MetadataProvider>

<MetadataProvider id="LocalMetadata"  xsi:type="FilesystemMetadataProvider" metadataFile="PATH_TO_YOUR_METADATA"/>
```

LIAF Metadata

edugain Metadata

Local Metadata (OPTIONAL)

****Make sure to add before the very last </MetadataProvider> , and to keep xml integrity

/opt/shibboleth-idp/conf/attribute-filter.xml

```
<!-- Release some attributes to an SP. -->
    <AttributeFilterPolicy id="example1">
        <PolicyRequirementRule xsi:type="Requester" value="https://sp.example.org" />

        <AttributeRule attributeID="eduPersonPrincipalName">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>

        <AttributeRule attributeID="uid">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>

        <AttributeRule attributeID="mail">
            <PermitValueRule xsi:type="ANY" />
        </AttributeRule>
    </AttributeFilterPolicy>
```