IdPSPLocalTestInstall

Local Installation Recipe

The directions are for creating a sandbox'ed installation for testing purposes. They should never be used for a production installation.

We recommend all installers start with one provider and test it against a known relying party first because it's less confusing overall and simpler to troubleshoot one installation at a time. However, if you're in an environment where that's impossible, due to firewalls or policy rules, please follow the steps below closely.

1. Preparation
2. IdP Installation
3. SP Installation
4. Basic Configuration
5. Login Testing
6. Attribute Configuration
7. Attribute Testing

Preparation

- `sp.machine` will be the SP domain. You can replace it in all instructions, or map it to an address using `/etc/hosts`
- `idp.machine` will be the IdP domain. You can replace it in all instructions, or map it to an address using `/etc/hosts`. We'll be using an Apache-based installation to allow these to optionally co-exist on the same machine.
- Both machines can be hosted on the same machine, or they can be virtual hosts, or they can be on separate machines.
- Port 80 and 443 need to be open. Watch out for firewalls and permission problems.
- Using a RedHat derivative (e.g. CentOS 5) is simplest.
- Make sure your clocks are set properly.

IdP Installation

1. Download and install Java 1.5+, Tomcat 6.0 and Apache 2.2+. Install Tomcat at `/usr/local/tomcat`. Avoid other distributions of Tomcat, such as from `yum`. Make sure the `$JAVA_HOME` environment variable points at the root of your JDK.
2. Download the idp .bin binary installer from [http://shibboleth.net/downloads/identity-provider/latest/](http://shibboleth.net/downloads/identity-provider/latest/), unzip it, and move to that directory. The following is an example and may point to an older version.

```bash
unzip shibboleth-identityprovider-2.3.8-bin.zip
 cd identityprovider/
```
3. Run `./install.sh` (`./install.bat on Windows`). This is a new installation. Please use `/opt/shibboleth-idp` for your IdP directory.

```bash
$ ./install.sh
```
4. Configure Apache by adding the following line to `httpd.conf` or `ssl.conf` in the SSL virtual host to pass requests for the IdP into Tomcat:

```bash
ProxyPass /idp/ ajp://localhost:8009/idp/
```
5. Enable Tomcat to run the IdP by endorsing additional libraries for XML processing. You will need to make the endorsed directory.

```bash
mkdir /usr/local/tomcat/endorsed/
cp endorsed/*.jar /usr/local/tomcat/endorsed/
```
6. Change Tomcat's `/usr/local/tomcat/conf/server.xml` port 8009 AJP13 connector. We'll add `request.tomcatAuthentication="false"` and `Address="127.0.0.1"` to allow Apache to send usernames to the IdP.

```xml
<Connector port="8009"
   enableLookups="false" redirectPort="8443" protocol="AJP/1.3"
   request.tomcatAuthentication="false" address="127.0.0.1" />
```
7. Define the following in httpd.conf or ssl.conf to force users to authenticate when accessing the RemoteUser handler.

```xml
<Location /idp/Authn/RemoteUser>
  AuthType Basic
  AuthName "My Identity Provider"
  AuthUserFile /usr/local/idp/credentials/user.db
  require valid-user
</Location>
```

8. Create a test user or two using the `htpasswd` command.

```
htpasswd -c /usr/local/idp/credentials/user.db myself
```

9. Deploy the WAR to Tomcat.

```
cp /opt/shibboleth-idp/war/idp.war /usr/local/tomcat/webapps/
```

### SP Installation

The installation of the SP is different depending on your platform. Please follow the right instructions for your machine. Linux RPM installation for Apache is simplest. IIS is most complex due to host mapping. Return here when you're done.

### Basic Configuration

We need to configure the providers to talk to each other. Please make these precise changes.

The SP configuration files are in `/etc/shibboleth/` for an RPM-based installation.

**SP: shibboleth2.xml:**

1. Change the entityID on the `<ApplicationDefaults>` element to `https://sp.machine/shibboleth`. This is your SP's name.
2. Change the entityID on the `<SSO>` element to `https://idp.machine/idp/shibboleth`. This is your IdP's name, and will send users directly to your IdP's login service (via the location specified in its metadata).
3. Uncomment the remotely maintained metadata example `<MetadataProvider>`. This will describe the IdP to your SP. Change the `uri` to point to `http://idp.machine/idp/profile/Metadata/SAML`. Delete or comment out both of the `<MetadataFilter>` elements as they will not allow the metadata being loaded from the IdP host to work.

**SP: shibd.logger:** Set DEBUG logging using log4j. `rootCategory=DEBUG, shibd_log`.

The IdP configuration files are in `/opt/shibboleth-idp/conf/`.

**IdP: relying-party.xml:**

1. Uncomment the `MetadataProvider` reading metadata from a URL. Change the `metadataURL` to `http://sp.machine/Shibboleth.sso/Metadata`. This will describe a basic SP to your IdP; for more advanced deployments, you'll need to hand-edit that metadata to reflect more advanced use cases.
2. Comment out the `MetadataFilter` element contained in the `MetadataProvider` element. The metadata is unsigned, so this filter would cause the metadata to fail to load.

Restart Tomcat, Apache, and `shibd`.

### Login Testing

Access `https://sp-machine/secure` using a web browser. Don't forget the `https`.

This is the fake location we protect for testing by default. It would be the URL of the application you want to protect. You should be redirected to your IdP, asked to login, and then see a 404 or `This page could not be displayed` error. That's because there is no real webpage at `/var/www/html/secure/index.html`, but you can create one. Go to `https://sp-machine/Shibboleth.sso/Session` with the same web browser. You should see basic login information. You now have a full login system, but attributes aren't enabled yet. If you have problems, [here's the troubleshooting page](#)

### Attribute Configuration

The easiest way to add attributes is to set up an LDAP directory. However, we can fake a lot using the built-in staticAttributes. The configuration directories are the same.
SP: attribute-map.xml: Uncomment all the sets of attribute mappings you can find. This tells the SP to listen for many common attributes.

IdP: attribute-resolver.xml: The IdP acquires attributes to send to the SP using this file.

1. Uncomment the Example Static Connector.
2. Uncomment the eduPerson attribute definitions except for eduPersonTargetedID and eduPersonTargetedID.old, and all the others you'd like. The eduPersonTargetedID definitions rely on a computed attribute definition that we won't have available.
3. Add a <resolver:Dependency ref="staticAttributes" /> to the eduPersonEntitlement and eduPersonAffiliation attribute definitions, just like the myLDAP examples.
4. Comment out the existing dependencies for <resolver:Dependency ref="myLDAP" /> in the attribute definitions you've uncommented, unless you're also also using the example <resolver:DataConnector id="myLDAP"/> from the distributed resolver configuration.

IdP: attribute-filter.xml: This tells the IdP which attributes to send to which SP. We're going to send two more attributes to your SP.

1. Uncomment the last <AttributeFilterPolicy>, with the portal example.
3. Change the <AttributeRule> attributeID to eduPersonAffiliation.
4. Create a second, identical <AttributeRule> with an attributeID of eduPersonEntitlement.

It should look like this:

```xml
<AttributeFilterPolicy>
  <AttributeRule attributeID="eduPersonAffiliation">
    <PermitValueRule xsi:type="basic:ANY" />
  </AttributeRule>
  <AttributeRule attributeID="eduPersonEntitlement">
    <PermitValueRule xsi:type="basic:ANY" />
  </AttributeRule>
</AttributeFilterPolicy>
```

Restart Tomcat, Apache, and shibd.

Attribute Testing

Access https://sp-machine/secure using a web browser. Don't forget the https. Go to https://sp-machine/Shibboleth.sso/Session with the same web browser. You should see a couple attributes. You can find more information in /var/log/shibboleth/shibd.log. If you have problems, here's the troubleshooting page.