Installing Shibboleth SP on RedHat based Linux

Introduction

There is already a significant amount of documentation on installing a Shibboleth SP notably:

- Understanding Shibboleth: how it all fits together: https://wiki.shibboleth.net/confluence/display/CONCEPT/FlowsAndConfig (useful for terminology and understanding how the Shibboleth SP uses session cookies)
- Installation: https://wiki.shibboleth.net/confluence/display/SP3/Installation (for installing on Linux, Mac, or Windows)
- SWITCH SP Installation manual for Debian and Ubuntu: https://www.switch.ch/aai/guides/sp/installation/
- Configuration reference: https://wiki.shibboleth.net/confluence/display/SP3/Configuration

This page draws on the above documents and gives the series of steps to install a Shibboleth SP and get it working in the Tuakiri federation.

This documentation now covers Shibboleth SP 3.x - though it does not significantly differ from 2.x for which this documentation was originally written. To upgrade from 2.x to 3.2, please see our Shibboleth SP 2.x to 3.x Upgrade Manual.

This documentation has been tested on RHEL/CentOS 6 and 7, but should work on other RedHat-based systems as well.

We recommend installing the most recent Shibboleth SP version. Version 3.3.0 is the latest version as of January 2022. We recommend updating existing deployments to the most recent version to get fixes for known vulnerabilities - please see the list of security advisories.

Prerequisites

Firewall settings

- inbound traffic:
  - webserver: port 80 and/or 443 are used by any browser-user
- outbound:
  - Shibboleth daemon (`shibd`): has to be able to connect to every remote IdP in the federation on port 8443 for back-channel communication.

Time synchronization

The host where Shibboleth SP is running must have time synchronized. We recommend using NTP for doing so - and synchronizing with your local NTP server. An example of configuring NTP can be found in the IdP Install Manual.

Dependencies

Before starting to build and configure the Shibboleth Sevice Provider, be sure that the dependent packages (Apache, and the `mod_ssl` module for Apache) are installed:

```
yum install httpd mod_ssl
```

SELinux

Configuring SELinux to permit httpd-shibd communication

These steps are required on an RHEL6/CentOS6 system with SELinux running in Enforcing mode - otherwise, `mod_shib` running inside Apache would not be able to communicate with `shibd`.

On RHEL7/CentOS7, the default SELinux policy already permits these actions and the following step is not required.

To configure SELinux to allow Apache (where `mod_shib` is loaded) to connect to `shibd`:

- Create a policy type enforcement file defining a policy module `mod_shib-to-shibd` - create `mod_shib-to-shibd.te` with the following contents:

```
module mod_shib-to-shibd 1.0;

require {
        type var_run_t;
        type httpd_t;
        type initrc_t;
        class sock_file write;
        class unix_stream_socket connectto;
}

#============= httpd_t ==============
allow httpd_t initrc_t:unix_stream_socket connectto;
allow httpd_t var_run_t:sock_file write;
```

- Compile, package and load the module with:

```
checkmodule -m -M -o mod_shib-to-shibd.mod mod_shib-to-shibd.te
semodule_package -o mod_shib-to-shibd.pp -m mod_shib-to-shibd.mod
semodule -i mod_shib-to-shibd.pp
```

Installation

Shibboleth SP is available for RedHat and derivative distributions via yum repositories maintained by the Shibboleth Project. The repository configuration files are generated by the shibbolet.net download site based on the target Linux distribution. You can either download the `.repo` file directly by passing in the distribution name as per the examples below, or you can download it via a browser from https://shibboleth.net/downloads/service-provider/latest/RPMS/ and then copy it to the target system.

- Add the `yum` repository for your distribution (example for CentOS 7):

```
wget -O /etc/yum.repos.d/shibboleth.repo https://shibboleth.net/cgi-bin/sp_repo.cgi?platform=CentOS_7
```

The table below includes links for additional supported distributions (taken from the download form linked above)

Platform (OS)	URL
CentOS 8 and RHEL 8	https://shibboleth.net/cgi-bin/sp_repo.cgi?platform=CentOS_8
CentOS 7 and RHEL 7	https://shibboleth.net/cgi-bin/sp_repo.cgi?platform=CentOS_7

Rocky Linux 8	https://shibboleth.net/cgi-bin/sp_repo.cgi?platform=rockylinux8
Amazon Linux 2	https://shibboleth.net/cgi-bin/sp_repo.cgi?platform=amazonlinux2

RHEL 7 RPMs

The Shibboleth Project provides binary packages for CentOS systems, but due to licensing restrictions, cannot build packages for RHEL 7 and above - full details are at https://wiki.shibboleth.net/confluence/display/SP3/RPMInstall.

For RHEL 7 (and above) systems, please use the binary-compatible CentOS repository.

And, for version 8, as CentOS 8 is not what it was expected to be, Rocky Linux 8 is a supported alternative...

- Install latest version via `yum`:

```
yum install shibboleth
```

Federation Membership

In order to register a Service Provider with the Federation Registry, it is **highly recommended** that you are able to log in with a user account authorised by an IdP or Virtual Home already registered with the federation.

It is possible to add an SP to the federation without an account but to become the administrator of that SP or later review the SP registration entry or to make any changes (in the textual description or the technical details - endpoint URLs, certificates, attributes required, etc), you **will** need an account.

Navigate to the Tuakiri federation management site https://registry.tuakiri.ac.nz/federationregistry (or, for Tuakiri-TEST federation, https://registry.test.tuakiri.ac.nz/federationregistry).

If you **do not** have an account with an IdP registered in the federation and you **do not** have an account with the Tuakiri VHO, start the SP registration (without logging in), by clicking the **Create Service Provider** link in the blue menu bar.

Otherwise, click **Login** and login using your IdP. Start the registration by clicking **Subscribers** > **Service Providers** > **Create**.

The registration form first displays a check-list of required information. Please check that you have all the information the check-list asks for readily available, otherwise the registration form may time out while you gather missing information.

Please note that on the registration form, you'll be asked to select the organization you are registering the SP under. If you have not registered your organization into Tuakiri (or Tuakiri-TEST) yet, please complete that process first, following our instructions for Creating an Organization in the Tuakiri Federation.

1. If you are filling this form without having logged in, you'll have to enter your details: **Given Name**, **Surname** and **Email** (otherwise, these are prefilled). (Do not use a shared mailbox, alias or mailing list when entering an email address because the confirmation email contains a single-use link and may cause some confusion should more than one person attempt to use it.)

2. Enter the information that describes the Service Provider being registered:
    - Select your **Organization**; if the organization you wish to host the Service Provider under does not exist, follow this procedure to create one. (*Required*)
    - Enter a **Display Name** and **Description** for your Service Provider. (*Required*)
    - Enter a **Service URL** for accessing your Service Provider in the form `http://sp.example.org`. The service URL is typically the base URL for accessing the Service Provider. (*Required*)
    - Optionally enter a URL in the **Service Logo URL** to a image or logo representing the service the Service Provider is authenticating. (*Optional*)
3. Enter the basic SAML configuration:
    - Choose the Shibboleth SP version that is installed on the Service Provider.
    - Enter the Service Provider's base URL for the **Host**, without a trailing slash, just as **`https://sp.example.org`**. The Federation Registry will automatically create all of the SAML2 endpoints from this base URL. You can skip over the **Advanced SAML2 Registration** section.
    - Alternatively, if your deployment does not fit the basic SAML configurator (either your implementation is not among the pre-configured ones, or you use custom endpoint URLs), please fill in the entityID and the endpoint URLs in the **Advanced SAML2 Registration** section.
4. Copy and paste in the back-channel certificate(s) generated when installing the Shibboleth SP software.

    For Shibboleth SP 2.x, the certificate is usually located in `/etc/shibboleth/sp-cert.pem`.

Starting with Shibboleth 3.0, the two separate certificates are generated for encryption and signing. These certificates are stored as `/etc/shibboleth/sp-signing-cert.pem` and `/etc/shibboleth/sp-encrypt-cert.pem`

As of October 2018, Federation Registry supports including separate signing and encryption certificates on the registration form. If you are registering an SP that has a single certificate used for both signing and encryption, copy the same certificate into both fields. If your SP does not support encryption, leave the encryption certificate field blank (but a signing certificate is required).

When pasting the certificate into the form, please take care that no line breaks, spaces or other characters are introduced during the cut-paste process.

- Please note that it is highly recommended that the `CN` in the certificate matches the hostname the service provider is being registered under. If this is an alias and your system thinks of itself with a different hostname, we recommend you instead generate a new certificate with the correct hostname: run the following, substituting the externally visible hostname for `sp.example.org`:

```
cd /etc/shibboleth
# for a Shibboleth 2.x system:
./keygen.sh -f -u shibd -g shibd -y 20 -h sp.example.org -e https://sp.example.org/shibboleth
# or for a Shibboleth 3.x system:
./keygen.sh -f -n sp-signing -u shibd -g shibd -y 20 -h sp.example.org -e https://sp.example.org
/shibboleth
./keygen.sh -f -n sp-encrypt -u shibd -g shibd -y 20 -h sp.example.org -e https://sp.example.org
/shibboleth
```

5. Select the attributes **Requested** and mark which are **Required**. For each attribute requested give a good explanation for why the attribute is requested. This information will later be displayed to users as justification for why the information is being released.

Persistent NameID

Please note that with the IdPv3 upgrade, Tuakiri is moving from passing Persistent NameIDs in the eduPersonTargetedID attribute to passing them as a Persistent SAML2 NameID. When registering a new SP requesting a persistent NameID, please request both the eduPersonTargetedID attribute (for interoperability with existing V2 IdPs), as well as NameID of Persistent format. You will be able to add the SAML 2.0 Persistent NameIDFormat after your SP registration is approved - or please get in touch with the Tuakiri Support.

schac attributes

Please note that as of 2.6.0, Shibboleth SP includes attributes from the schac schema in the default configuration. The names used for the attributes there are slightly different from what has been used in the attribute-map.xml file provided by Tuakiri for use with earlier versions of Shibboleth SP. For compatibility with 2.6.0, we have adjusted the names in attribute-map.xml to match the names used by the 2.6.0 default configuration.

homeOrganization is becoming **schacHomeOrganization**
homeOrganizationType is becoming **schacHomeOrganizationType**

eduPersonEntitlement attribute

Please note: if intending to request the eduPersonEntitlement attribute, you cannot add the attribute to the list of requested attributes on the registration page; you'll have to add it separately later.

Also, because of the nature of this attribute, you also have to include a specific requested value (or a regular expression matching a set of values), but an eduPersonEntitlement attribute request without specific values is not considered complete and will be ignored by the Federation Registry.

6. Click **Submit** and wait for a confirmation email.

Please note: Once the registration is approved, the Federation Registry will send an email with an invite code to claim administrative rights over the SP being registered.

It is important to follow the instructions in the email to get the administrative privileges over the SP. These privileges are required for making any subsequent changes to the SP registration.

Note that the invite code can only be used once - but once the original recipient has administrative privileges, these can be used to grant the same administrative privileges to additional users as required.

ECP support

If your SP should support ECP (access via non-browser clients), then also register support for ECP:

- After your SP registration is complete, log into the Federation Registry again (in the same way as above)
- Open the entry for your SP (under Subscriprs -> Service Providers or directly from the Dashboard)
- Under EndPoints -> Assertion Consumer Service, add a new Endpoint:
  - Select Binding: `urn:oasis:names:tc:SAML:2.0:bindings:PAOS`
  - Enter Location: `https://sp.example.org/Shibboleth.sso/SAML2/ECP` (substituting `sp.example.org` with your SP hostname)
  - Enter Index: 4 (value 4 matches the value in the Shibboleth SP internal metadata in the default configuration)
- Remember to also configure support for ECP in your `/etc/shibboleth/shibboleth2.xml` file.

Configuration

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

- Replace all instances of `sp.example.org` with your hostname.
- In the <Sessions>element:
  - Make session handler use SSL: set `handlerSSL="true"`
    Recommended: go even further and in the '"Sessions"' `Sessions` element, change the '"handlerURL"' `handlerURL` from a relative one (`"/Shibboleth.sso"` to an absolute one - `handlerURL="https://sp.example.org/Shibboleth.sso"`. In the URL, use the hostname used in the endpoint URLs registered in the Federation Registry. This makes sure the server is always issuing correct endpoint URLs in outgoing requests, even when users refer to the server with alternative names. This is in particular important when there are multiple hostnames resolving to your server (such as one prefixed with "www." and one without).
  - We also strongly recommend to configure the SP to use **secure** cookies that would only be sent over an encrypted (https) connection. Unless you are also using plain HTTP to access your application in authenticated mode (which is dangerous - risk of cookie theft / session hijacking), change the `cookieProps` setting to use secure cookies:

    ```
    cookieProps="https"
    ```

  - Configure Session Initiator: locate the <SSO>element and:
    - Remove reference to default `idp.example.org` - delete the `entityID` attribute
    - Configure the Discovery Service URL in the `discoveryURL` attribute:

      ```
      discoveryURL="https://directory.tuakiri.ac.nz/ds/DS"
      ```

    - or, alternatively, if connecting to the Tuakiri TEST federation (Staging Environment), use:

      ```
      discoveryURL="https://directory.test.tuakiri.ac.nz/ds/DS"
      ```

- In `AttributeExtractor`, set `reloadChanges="true"`
- Shibboleth 2.x only: restrict cipherSuites:

  In earlier versions (Shibboleth SP 2.x), we were recommending to configure the TLS protocols and cipher-suites acceptable on the back-channel - the default settings were overly permissive and insecure.

  Shibboleth 3.x now sets a new default, identical to our recommendation in terms of actual ciphers permitted. So, this step is no longer needed on Shibboleth SP 3.x

  On Shibboleth SP 2.x, add the following XML attribute to the <`ApplicationDefaults`> element:

  ```
  cipherSuites="DEFAULT:!EXP:!SSLv2:!DES:!IDEA:!SEED:!RC4:!3DES:!kRSA:!SSLv3:!TLSv1:!TLSv1.1"
  ```

  This sets the protocols to TLSv1.2 only (banning SSLv2, SSLv3, TLSv1.0, TLSv1.1) and blocks all ciphers deemed insecure (as of October 2017).
- Optionally, customize settings in the <`Errors`> element. These settings configure the error handling pages that would be rendered to the users should an error occur. At the very least, we recommend changing the `supportContact` attribute from `root@localhost` to your support service email address. Documentation for advanced configuration of error handling is available at the Shibboleth SP Errors documentation page.

- Download the metadata signing certificate for the federation metadata into `/etc/shibboleth`:
  - For Tuakiri, run:

    ```
    wget https://directory.tuakiri.ac.nz/metadata/tuakiri-metadata-cert.pem -O /etc/shibboleth/tuakiri-metadata-cert.pem
    ```

  - or for Tuakiri-TEST, run:

```
wget https://directory.test.tuakiri.ac.nz/metadata/tuakiri-test-metadata-cert.pem -O /etc
/shibboleth/tuakiri-test-metadata-cert.pem
```

- Load the federation metadata: add the following (or equivalent) section into `/etc/shibboleth/shibboleth2.xml` just above the sample (commented-out) `MetadataProvider`element.
  - For Tuakiri add:

```
        <MetadataProvider type="XML" url="https://directory.tuakiri.ac.nz/metadata/tuakiri-
metadata-signed.xml"
                backingFilePath="metadata.tuakiri.xml" reloadInterval="7200" validate="true">
            <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
            <MetadataFilter type="Signature" certificate="tuakiri-metadata-cert.pem" verifyBackup="
false"/>
        </MetadataProvider>
```

  - For Tuakiri-TEST, add instead:

```
        <MetadataProvider type="XML" url="https://directory.test.tuakiri.ac.nz/metadata/tuakiri-
test-metadata-signed.xml"
                backingFilePath="metadata.tuakiri-test.xml" reloadInterval="7200" validate="true">
            <MetadataFilter type="RequireValidUntil" maxValidityInterval="2419200"/>
            <MetadataFilter type="Signature" certificate="tuakiri-test-metadata-cert.pem"
verifyBackup="false"/>
        </MetadataProvider>
```

- The Shibboleth SP installation needs to be configured to map attributes received from the IdP - in `/etc/shibboleth/attribute-map.xml`. Change the attribute mapping definition by either editing the file and uncommenting attributes to be accepted, or replace the file with the recommended **Tuakiri attribute-map.xml file mapping *all* Tuakiri attributes** (and optionally comment out those attributes not used by your SP). This can be conveniently done with

```
wget -O /etc/shibboleth/attribute-map.xml https://github.com/REANNZ/Tuakiri-public/raw/master/shibboleth-
sp/attribute-map.xml
```


In addition to mapping received attributes to local names (and thus accepting them), it is also possible to configure filtering rules in `attribute-policy.xml`.

In most cases, this can be left as-is (the default rules do the filtering applicable to Tuakiri attributes), but additional rules can be added here.

For further information, please see https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPAttributeFilter

- With earlier versions of Shibboleth SP (2.x), it was necessary to work around issues with rotation of logs generated by the `mod_shib` module running inside Apache. In Shibboleth SP 3.x, this module logs via syslog and this is no longer an issue. If deploying a 2.x installation or explicitly logging to file, expand this section (otherwise archived for historical purposes only).

  Workaround : move the log rotation from the module to logrotate.

  - Otherwise, SELinux rules would not permit the log rotation (by design Apache is allowed to only *append* to logs, but cannot remove them – incl. renaming).
  - And there is also a race condition in the log rotation. This has been reported upstream as SSPCPP-757 - and we recommend to move log rotation out of `mod_shib` to `logrotate`.
  - Edit `/etc/shibboleth/native.logger` and:
    - replace `RollingFileAppender` with `FileAppender`
    - comment out log rotation-specific options: `maxFileSize` and `maxBackupIndex`
    - or just replace the file with our copy with exactly these customizations: native.logger
  - Install a new file into `/etc/logrotate.d/shibboleth-www` to rotate these files via `logrotate` (and reload Apache post-rotate): shibboleth-www containing:

```
/var/log/shibboleth-www/*.log {
    missingok
    daily
    rotate 10
    nodateext
    size 1000000
    sharedscripts
    postrotate
        /sbin/service httpd reload > /dev/null 2>/dev/null || true
    endscript
}
```

- These can be both installed with:

```
wget -O /etc/shibboleth/native.logger https://github.com/REANNZ/Tuakiri-public/raw/master
/shibboleth-sp/native.logger
wget -O /etc/logrotate.d/shibboleth-www https://github.com/REANNZ/Tuakiri-public/raw/master
/shibboleth-sp/logrotate-redhat/shibboleth-www
```

Special considerations

HTTP/HTTPS access

Normally, the Shibboleth endpoints are accessible only via HTTPS (also configured by the `handlerSSL="true"` setting above). Applications that make use of (plain) http for access to content using Shibboleth protection can run into issues if the client is using inconsistent proxy connection settings for http and https.

By default Shibboleth SP checks that the IP address stays the same - but in this case, the IP address for the http and https traffic appears to be different. The safety mechanisms then suspect the session has been hijacked and terminate the session. This can lead to the SP keeping the user in an infinite loop.

For such applications we recommend setting `consistentAddress="false"` on the <Sessions> element:

```
consistentAddress="false"
```

RedHat Enterprise Linux 6 and 7

Please note that RedHat Enterprise Linux 6 and 7 (and so also CentOS 6 and 7) come with CURL built against NSS, not OpenSSL. Using this version of the CURL libraries would break the SOAP calls Shibboleth SP is making to the IdP port 8443 (back-channel communication) for artifact resolution and attribute queries. While initial approach taken by the Shibboleth SP project was to provide CURL version linked against OpenSSL that would "upgrade" (replace) the one that comes with the OS, this was later seen as having undesired consequences and the new approach is to instead provide "look-aside" version of the library that installs into `/opt`.

These libraries install automatically as dependencies of the main shibboleth package and no action is needed by the deployer.

Further information is available in the upstream documentation at https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPLinuxRH6

ECP

If your SP should support ECP (access via non-browser clients), then also:

1. Edit the <SSO> element in `/etc/shibboleth/shibboleth2.xml` and add an `ECP="true"`attribute:

```
<SSO ECP="true" ....>
```

2. Add support for ECP in the metadata registered in the federation (as instructed above).

Logging

Shibboleth SP has two separate components (the `shibd` daemon and the `mod_shib` module running inside Apache), and they also have separate logging configuration and destinations.

- The `shibd` daemon logs primarily into `/var/log/shibboleth/shibd.log` (with transaction details in `/var/log/shibboleth/transaction.log`)
  - Logging configuration is in `/etc/shibboleth/shibd.logger`
  - Log files should be owned by `shibd` (the user account `shibd` daemon runs under)
- The `mod_shib` Apache module logs into syslog (as facility `LOCAL0`).
  - Logging configuration in `/etc/shibboleth/native.logger`
  - In Shibboleth SP 2.x, `mid_shib` was logging into `/var/log/shibboleth-www/native.log` and `/var/log/shibboleth-www/native-warn.log`Log (and these files were owned by `apache`, the user account Apache httpd runs under)

Protecting a resource

You can protect a resource with Shibboleth SP by adding the following directives into your Apache configuration. By default, a sample configuration snippet protecting the `/secure` URL on the server is included in `/etc/httpd/conf.d/shib.conf`:

```
<Location /secure>
  AuthType shibboleth
  ShibRequestSetting requireSession 1
  require shib-session
</Location>
```

You can add additional access control directives either to this file or anywhere else in the Apache configuration, as it fits with your application.

Another frequently used technique is *lazy sessions* - access is granted also for unauthenticated users, but if a session exists, the attributes in the session are passed through to the application - and the application can then make access control decision (and initiate a login where needed).

Applying lazy sessions (making the Shibboleth sessions visible) to the whole application can be achieved e.g. with:

```
<Location />
  AuthType shibboleth
  ShibRequestSetting requireSession 0
  require shibboleth
</Location>
```


Apache 2.2 deployments

Because the way authentication modules (like `mod_shib`) link into Apache has changed substantially between Apache 2.2 and 2.4, the directives to protect a resource with mod_shib has changed as well.

The module provides the `ShibCompatWith24` directive to emulate the Apache 2.4 behavior on Apache 2.2 and we recommend using this directive on new deployments (if they are with Apache 2.2) - the configuration will otherwise be ready for Apache 2.4.

However, this directive is **only** available with Apache 2.2 and is **not** available on Apache 2.4, so only use it on actual Apache 2.2 deployments.

Protecting a resource with eager protection in Apache 2.2:

```
<Location /secure>
  AuthType shibboleth
  ShibCompatWith24 On
  ShibRequestSetting requireSession 1
  require shib-session
</Location>
```

Protecting a resource with lazy sessions in Apache 2.2:

```
<Location />
  AuthType shibboleth
  ShibCompatWith24 On
  ShibRequestSetting requireSession 0
  require shibboleth
</Location>
```

Note that in this case, to actually trigger a login, the application would have to redirect the user to a Session Initiator - a default one is located at `/Shibboleth.sso/Login` (see the links below for more details).
You are welcome to use the Tuakiri logo with the Login link - please visit our Logos page to get a suitably sized Tuakiri logo.

For further information, please see the following pages in the Shibboleth SP documentation:

- Protecting a resource: basic concepts: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPProtectContent
- Protecting a resource with Apache directives: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPhtaccess
- Shibboleth SP Apache module configuration reference: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPApacheConfig
- Shibboleth SP configuration reference: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPConfiguration
- Integrating Shibboleth SP with your application: https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPEnableApplication
- Shibboleth configuration How-Tos: https://wiki.shibboleth.net/confluence/display/SHIB/ConfigurationHowTos
- Session creation parameters (when using a session initiator): https://wiki.shibboleth.net/confluence/display/SHIB2/NativeSPSessionCreationParameters

Finishing up

- Start up Apache and shibd:

```
service httpd start
service shibd start
chkconfig httpd on
chkconfig shibd on
```


On RHEL7/CentOS7 using systemd, the commands should properly be:

```
systemctl enable httpd shibd
systemctl start httpd shibd
```

(but the legacy syntax invoking `service` and `chkconfig` still works and is rerouted to systemctl)

Testing

1. Place a script inside the protected directory. PHP example script such as the following is good enough:

```
<?php print_r($_SERVER) ?>
```

2. Access the protected directory/script (http://your.server/secure) from your browser, this should trigger a complete SSO cycle where you can authenticate on your IdP
3. Upon successful authentication, the page should display all received attributes. Make sure you have non empty **Shib-Application-ID** amongst other attributes (if your IdP release them).
4. Check your **shibd.log** to see if there are attributes received or errors encountered.