Kerberos Login Handler

The Kerberos Idp Handler uses the kerberos protocol to implement an SSO (Single Sing On) authentication mechanism.

A presentation with some general information about the project can be found on the SWITCH website

Requirements

Kerberos Infrastructure

To use the "Kerberos Login Handler" in the Identity Provider, first is necessary to have the kerberos environment configured and working properly.

Some interesting tutorials are:

http://www.grolmsnet.de/kerbtut/

HTTP-Based Cross-Platform Authentication by Using the Negotiate Protocol.

Before start please check if:

- The service-principal (usually "HTTP/principal@your_realm.com") was configured at the KDC
- The keytab file that holds the key of the service-principal was generated (recommended)
- In the IDP: Check if is possible to get the service tickets with kinit from the KDC.

There is also a step by step guide to setting up kerberos at http://research.ncl.ac.uk/gfivo/documents/UsingKerberosticketsfortrueSingleSignOn.pdf The hardest area in getting shibboleth setup with kerberos "true single sign on" is the kerberos setup. It is therefore recommended that you get kerberos to work on it's own on the IdP server before you try integrating with shibboleth. This guide walks you through these steps, once you have kerberos setup and working on it's own integration with shibboleth is relatively straight forward. The guide details setup of mod_auth_kerb which is the simplest route to proving you have SPNEGO based "true single sign on" working . Once that is setup the Kerberos login handler detailed in this wiki page is a better solution for providing a user facing service as it's behavior when dealing with clients that don't have a kerberos ticket (mobile phones etc) is superior in that it presents a "web form" based login rather than the grey pop up "basic auth" login box.

Installation and configuration

- **Download and build the source (replace X.0 with the last stable version, e.g.: "tags/1.0"):**

```
svn export https://subversion.switch.ch/svn/general/aai/java-idp-kerberos-login-handler/tags/X.0 java-idp-kerberos-login-handler
cd java-idp-kerberos-login-handler
mvn package
```

IdP - Configuration

Copy the example pages from the source (/examples) to the installation folder at: $IDP_INSTALL_DIR/src/main/webapp/

- Basic installation:
  - kerberos-default.jsp.dist (rename to kerberos-default.jsp) - Main configuration file
  - kerberos-login.inc.jsp - The source code for the kerberos login form.
  - kerberos.css.dist - (rename to kerberos.css) Some styles to the kerberos login page.
- Optional:
  - kerberos-report.jsp (necessary if showTestPage=true at kerberos-default.jsp) - Test page to verify if the browser supports kerberos and links to documentation.
  - kerberos-config.jsp (necessary if option2Hide=true at kerberos-default.jsp) - Change the kerberos options (enabled, visible and auto-login) at the client/browser.
- Examples:
  - login_example.jsp - Example of Kerberos integrated with "UsernamePassword" Idp login page.
  - kerberos-login.jsp - Example of "Kerberos-only" login page.
  - unauthorized.html - Custom message for "401 - unauthorized" error (necessary if customUnauthorized configured at handler.xml).

Rename the distribution files:

```
mv kerberos-default.jsp.dist kerberos-default.jsp
mv kerberos.css.dist kerberos.css
```

And check the configurations at kerberos-default.jsp. In this file you can change the text messages, enable or disable some functionality (kerberos enabled, visible and auto-login) and customize functions.

Copy the .jar file to the installation folder:

```
cp target/kerberos-login-handler-X.0.jar $IDP_INSTALL_DIR/lib
```

To expose the Kerberos Login Handler to the user you have two options. Either you include a snippet into your existing UsernamePassword Login Handler or you can use just the Kerberos Login Handler.

1 - Kerberos embedded with UsernamePassword login page

- Rename "login_example.jsp" to "login.jsp" if there is not yet a "login page" installed.
- Edit the "login.jsp" (UsernamePassword login page) and include the following code snipped where you want that kerberos (button) shows up.

```
(...)
</form>

<jsp:include page="kerberos-login.inc.jsp" />

(...)
```

2 - "Kerberos only" login page

- Verify if "kerberos-login.inc.jsp" is in the "$IDP_INSTALL_DIR/src/main/webapp/" folder
- Use the "kerberos-login.jsp" page as an example. Rename it to login.jsp

Configuring the server (tomcat)

In the web-application you have to enable the Kerberos login servlets. You do that in *$IDP_INSTALL_DIR/src/main/webapp/WEB-INF/web.xml*:

```
 <webapp>

(...)
    <!-- Kerberos Login Handler -->
    <servlet>
      <servlet-name>KrbLoginServlet</servlet-name>
      <servlet-class>ch.SWITCH.aai.idp.kerberos.KrbLoginServlet</servlet-class>
    </servlet>

    <servlet-mapping>
      <servlet-name>KrbLoginServlet</servlet-name>
      <url-pattern>/Authn/Kerberos/Login</url-pattern>
    </servlet-mapping>

(...)

</webapp>
```

Attention!

- If you change the standard "KrbLoginServlet" url-pattern ("/Authn/Kerberos/Login"), you need also to set "krbServletPattern" in the kerberos-login-handler.xsd file.
- If you change the standard "UsernamePasswordAuthHandler" url-pattern ("/Authn/UserPassword"), you need also to set "loginPagePattern" in the kerberos-login-handler.xsd file.
- Pay attention that you still need the "UsernamePassword" Idp active and configured, in order to use both authentication methods.

Kerberos - Test Page

There is a Servlet to test if the browser supports kerberos. This configuration is necessary if showTestPage=true at kerberos-default.jsp (default=false).
- The "kerberos-report.jsp" file must to be in the "$IDP_INSTALL_DIR/src/main/webapp/" folder
- Configure the servlet at *$IDP_INSTALL_DIR/src/main/webapp/WEB-INF/web.xml*:

```
(...)
    <!-- Test Kerberos -->
    <servlet>
      <servlet-name>TestKrbServlet</servlet-name>
      <servlet-class>ch.SWITCH.aai.idp.kerberos.test.TestKrbServlet</servlet-class>
    </servlet>

    <servlet-mapping>
      <servlet-name>TestKrbServlet</servlet-name>
      <url-pattern>/Authn/Kerberos/test-kerberos</url-pattern>
    </servlet-mapping>
(...)
```

handler.xml configuration

Configure the handler.xml at:

- new install: $IDP_INSTALL_DIR/src/installer/resources/conf-tmpl/handler.xml
- reinstall: $IDP_DIR/conf/handler.xml

```
<ProfileHandlerGroup xmlns=
    "urn:mace:shibboleth:2.0:idp:profile-handler"
(...)
    xmlns:krb="http://www.switch.ch/aai/idp/kerberos"
    xsi:schemaLocation="
        urn:mace:shibboleth:2.0:idp:profile-handler classpath:/schema/shibboleth-2.0-idp-profile-handler.xsd
(...)
        http://www.switch.ch/aai/idp/kerberos classpath:/schema/kerberos-login-handler.xsd
        ">
(...)
    <!-- Kerberos Idp -->
    <ph:LoginHandler xsi:type="krb:KERBEROS"
                kerberosCfg="/opt/kerberos/krb5.conf"
                customUnauthorized="/opt/shibboleth-identityprovider-2.2.0/src/main/webapp/unauthorized.html"
    >
<!-- LoginHandler optional attributes:
                kerberosCfg - kerberos configuration file (e.g.: /etc/krb.conf)
                customUnauthorized - custom html page for error 401 - Unauthorized. (e.g.:  /opt/shibboleth-
identityprovider-2.2.1/src/main/webapp/error-404.jsp)
                auto_login_durantion - auto login duration (seconds)
                loginPagePattern - (default: "/login.jsp") - path for login page
                krbServletPattern - (default: "/Authn/Kerberos") - path for kerberos login page
-->
        <ph:AuthenticationMethod>urn:oasis:names:tc:SAML:2.0:ac:classes:Kerberos</ph:AuthenticationMethod>
        <krb:Realm domain="DOMAIN_A.COM">
            <krb:principal>HTTP/aai-logon.domain_a.com@DOMAIN_A.COM</krb:principal>
            <krb:keytab>/opt/kerberos/http_domainA.keytab</krb:keytab>
        </krb:Realm>

        <krb:Realm domain="DOMAIN_B.COM">
            <krb:principal>HTTP/aai-logon.domain_b.com@DOMAIN_B.COM</krb:principal>
            <krb:keytab>/opt/kerberos/http_domainB.keytab</krb:keytab>
<!-- Realm optional elements:
            <krb:password>password (if no keytab available)</krb:password>
-->
        </krb:Realm>
    </ph:LoginHandler>

(...)
</ProfileHandlerGroup>
```

Attribute resolver configuration

Configure the attribute-resolver.xml at:

- new install: $IDP_INSTALL_DIR/src/installer/resources/conf-tmpl/attribute-resolver.xm

- reinstall: $IDP_DIR/conf/attribute-resolver.xml

Check if your configuration will accept the "principal name" provided by Kerberos (format: Principal@REALM.COM).

Here you can find an example.

Log configuration

The logging for the Handler is configured in the logging.xml file. It can be found at:

- new install: $IDP_INSTALL_DIR/src/installer/resources/conf-tmpl/logging.xml
- reinstall: $IDP_DIR/conf/logging.xml

```
(...)

<!-- Log level for Kerberos - LoginHandler -->
<logger name="ch.SWITCH.aai.idp.kerberos">
<level value="DEBUG"/>
</logger>

(...)
```

Deployment

Backup the IdP configuration before re-deploying the application:

```
$IDP_INSTALL_DIR/install.sh
```

Configuration - Client Side

The client's browser must to be configured before using Kerberos - Single Sign on. See the Documentation

Troubleshooting

- Project page (bugs, features): https://forge.switch.ch/redmine/projects/java-idp-kerberos-login-handler
- Some FAQ

Contact for comments/questions:

Rodrigo Ristow - rodrigo.ristow@fhnw.ch

University of Applied Sciences and Arts Northwestern Switzerland (FHNW)