Higgins SAML2 IdP Tutorial

Version 1.1, Oct 18th 2007, msabadello@parityinc.net

The Higgins SAML2 IdP supports the SP-initiated SSO profile defined by SAML2 specifications. Two parties are involved in this profile: A service provider (relying party, SP), and an identity provider (IdP).

The SP offers protected services and relies on the IdP to authenticate users.

Contents

Higgins components

This document illustrates a sample setup of the following Higgins components:

```
org.eclipse.higgins.saml2idp.server
org.eclipse.higgins.saml2idp.saml2
org.eclipse.higgins.saml2idp.test
```

Other Higgins components required by the setup:

```
org.eclipse.higgins.configuration.api
org.eclipse.higgins.configuration.common
org.eclipse.higgins.configuration.xml
org.eclipse.higgins.configuration.xrds
org.eclipse.higgins.idas.api
org.eclipse.higgins.idas.common
org.eclipse.higgins.idas.cp.jndi
org.eclipse.higgins.idas.registry
org.eclipse.higgins.idas.spi
org.eclipse.higgins.util.idas.cp
org.eclipse.higgins.util.jscript
org.eclipse.higgins.util.socket
```

Third party dependencies:

```
axiom-api-1.2.jar
axiom-impl-1.2.jar
bandit-jndi-0.2.489.jar
bandit-misc-0.2.489.jar
commons-codec-1.3.jar
commons-logging.jar
dom4j-1.6.1.jar
js.jar
ldap.jar
log4j-1.2.13.jar
openxdas-0.3.192.jar
openxri-client.jar
openxri-syntax.jar
stax-api-1.0.1.jar
wstx-asl-3.0.1.jar
xalan-2.6.0.jar
xercesImpl.jar
xml-apis.jar
xmlsec-1.4.0.jar
```

Setup

In this document, the base URI of the SP is:

```
http://localhost/org.eclipse.higgins.saml2idp.test/
```

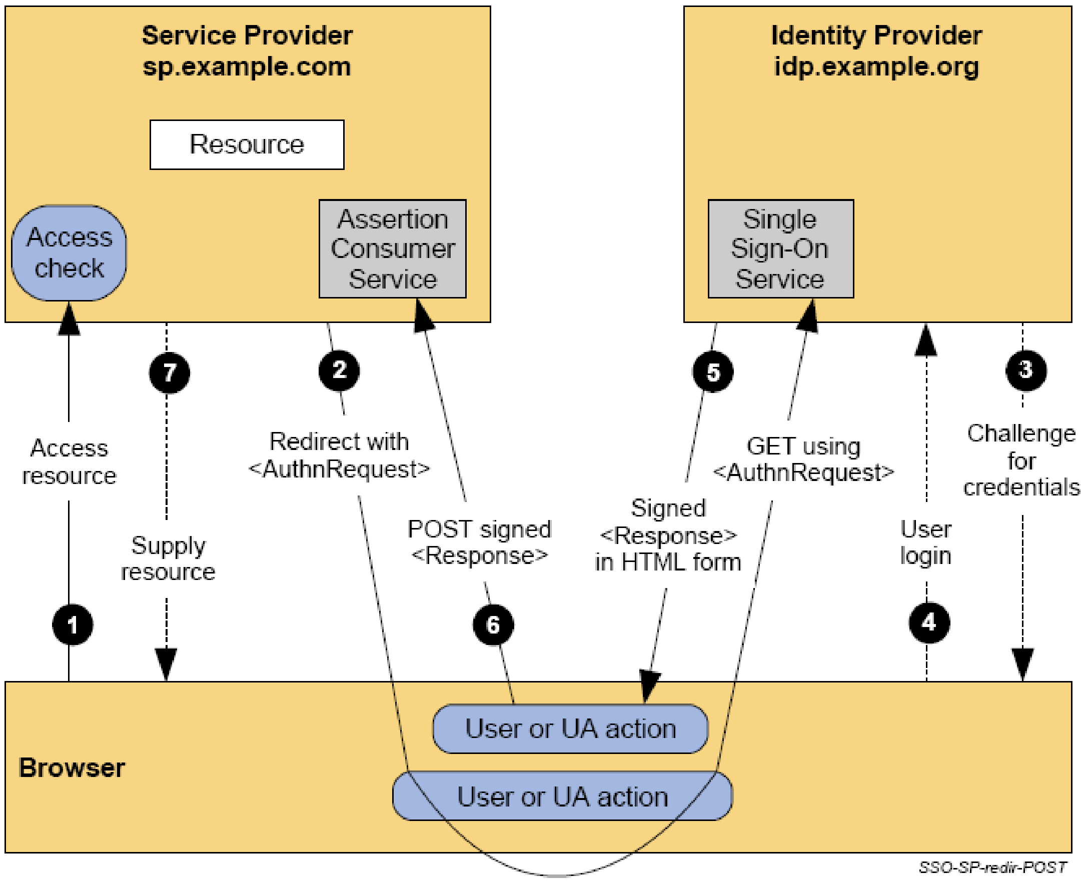
The base URI of the IdP is:

```
http://localhost/org.eclipse.higgins.saml2idp.server/
```

In the screenshots provided in this document, the relying party is indicated by a light blue screen border and the identity provider by an orange screen border.

Protocol Flow

The following diagram is taken from the SAML Technical Overview and outlines the protocol view in the SP-initiated SSO profile (see http://www.oasis-open.org/committees/security/).



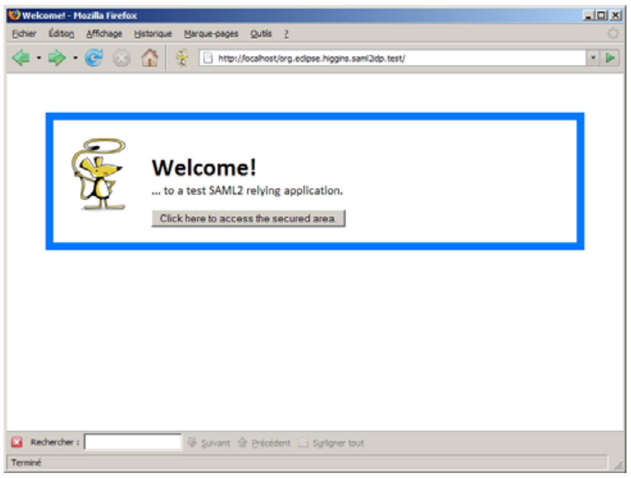
The remaining part of this document describes the protocol flow in greater detail and illustrates the roles played by Higgins components.

If the user is not logged in at the IdP yet, all 7 steps are executed (this is referred to as Flow #1 in the rest of the document).

If the user is already logged in at the IdP, steps 3 and 4 are omitted (this is referred to as Flow #2 in the rest of the document).

Flow #1: First time log in

In this situation, the user tries to access the SP for the first time. The following welcome screen appears:



Welcome!

... to a test SAML2 relying application.

Click here to access the secured area.

Figure 1: Start screen of the service provider.

In order to access the secured applications or pages at the SP, the user initiates some kind of action; in this case, a button is clicked. The SP then sends a SAML2 AuthnRequest message to the IdP using the SAML2 HTTP Redirect binding, i.e. by including the message in a deflated, base64-encoded form as a URI query string parameter.

The SP also optionally includes a SAML2 RelayState, i.e. string data it needs to use after the SAML2 protocol flow is complete (e.g. the original URL the user was trying to access).

The SAML2 endpoint URI of the IdP is a configurable setting at the SP.

Example SAML2 AuthnRequest message sent from the SP to the IdP:

```
<?xml version="1.0" encoding="UTF-8"?>
<AuthnRequest
  AssertionConsumerServiceURL="http://localhost/org.eclipse.higgins.saml2idp.test/SAMLEndpoint"
  Destination="http://localhost/org.eclipse.higgins.saml2idp.server/SAMLEndpoint"
  IssueInstant="2007-10-17T11:23:56.609Z" Version="2.0"
  ProtocolBinding="urn:oasis:names.tc:SAML:2.0:bindings:HTTP-Redirect"
  ProviderName="Test SAML2 SP"
  xmlns="urn:oasis:names:tc:SAML:2.0:protocol" />
```

This AuthRequest message is encoded and sent to the following URL at the IdP via HTTP Redirect:

```
http://localhost/org.eclipse.higgins.saml2idp.server/SAMLEndpoint?SAMLRequest=eJyVUEtLw0AQ%2FivL3
rN5iK0uTUt9YaFKSVIP3uJmSAeS2bCzCf58t62C1JPHGb73YvXZd2ICx2gpl6lKpAAytkFqc7mvnqIbuVouCuDBEoNYM4PzAX
sfzrEHV4Kb0MC%2B2Oby4P2g47izpu4Oln1sXavAdDgwqAO2LRIrrvsuw2ZQHgKiXL9sH6kZLJKX4iG8kGp%2FyvI%2FtRArt
LjQ2zCPsCH2NflcZkkyj9IkSudVmursSl%2FP1Cy5fZdi56y3xnZ3SOfioyNta0bWVPfAyht9VNaZSvTHGcT6uap2UQENOjD%
2BJDJhA%2B41MHJZhSriyMlEuZPi7Wfh7Lhw2Jz4j4v%2B7TJ8R5Lx8gvNJJBB&RelayState=Test+relay+state%21%21
```

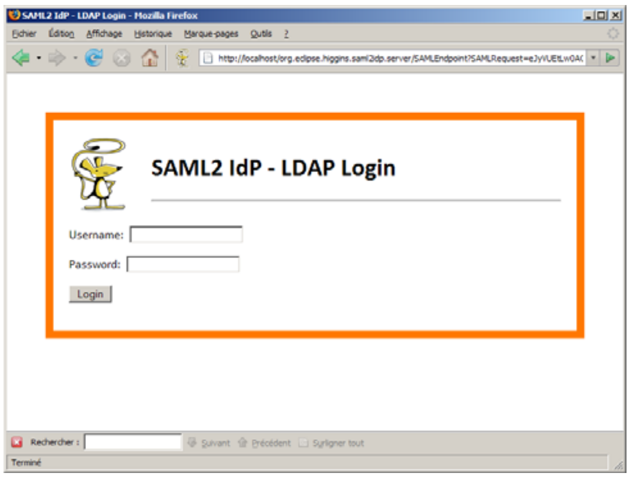
The AuthnRequest message is read by the IdP. In particular, the `AssertionConsumerServiceURL` attribute is needed to pass the SAML2 protocol response back to the SP.

The IdP keeps the SAML2 RelayState string to pass it back to the SP later.

The IdP checks if the user is already logged in by examining the session state. In this case, the user is not logged in, so the IdP needs to ask the user to provide authentication materials (credentials).

The kind of credentials required and the backend store they are validated against are configurable at the Higgins SAML2 IdP. This authentication mechanism is handled by the Higgins IdAS (Identity Attribute Service) component, which can authenticate users against a wide variety of underlying technologies.

In this case, an LDAP directory is used, therefore the IdP asks the user to provide a username and a password.



http://localhost/org.eclipse.higgins.saml2idp.server/SAMLEndpoint?SAMLRequest=eJyVUEtLw0A(

SAML2 IdP - LDAP Login

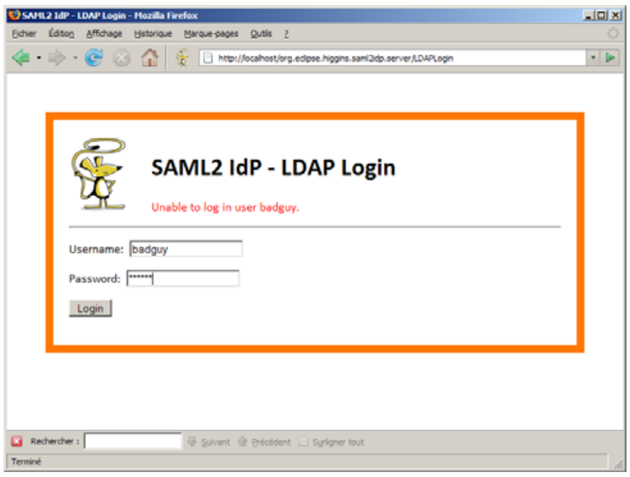
Username:

Password:

Login

Figure 2: The Higgins SAML2 IdP asks the user to provide credentials.

If the user enters invalid credentials, an error message is displayed.



http://localhost/org.eclipse.higgins.saml2idp.server/LDAPLogin

SAML2 IdP - LDAP Login

Unable to log in user badguy.

Username: badguy

Password: ******

Login

Figure 3: Invalid credentials entered at the IdP.

Once the user enters valid credentials, the IdP assembles a SAML2 Response message and sends it to the SP using the SAML2 HTTP POST binding, i.e. by including the message in a base64-encoded hidden input field in a HTML form. This form is auto-submitted using simple JavaScript. The original RelayState provided by the SP is also included as a form field.

In addition, the IdP notes in the user's session object that the user is logged in now, so that during future requests the user does not have to enter credentials again.

The SAML2 Response message is signed with a DSA private key using XML Signature. The SP is in possession of the corresponding public key to verify the signature. These keys are previously agreed on by the SP and the IdP.

The SP's SAML2 endpoint URI (the target of the HTML Form POST) was provided by the SP in the `AssertionConsumerServiceURL` of the original SAML2 AuthnRequest message.

Example signed SAML2 Response message sent from the IdP to the SP:

```
<?xml version="1.0" encoding="UTF-8"?>
<SAMLResponse xmlns="urn:oasis:names:tc:SAML:2.0:protocol"
  Destination="http://localhost/org.eclipse.higgins.saml2idp.test/SAMLEndpoint"
  IssueInstant="2007-10-17T11:42:41.828Z" Version="2.0">

  <Status><StatusCode Value="urn:oasis:names:tc:SAML:2.0:status:Success" /></Status>

  <Issuer xmlns="urn:oasis:names:tc:SAML:2.0:assertion">Test SAML2 IdP</Issuer>

  <Assertion xmlns="urn:oasis:names:tc:SAML:2.0:assertion"
    ID="1192632279953" IssueInstant="2007-10-17T14:44:39.953Z" Version="2.0">
    <Issuer>Test SAML2 IdP</Issuer>
    <Subject>
      <NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:entity">admin</NameID>
      <SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer" />
     </Subject>
    <AuthnStatement AuthnInstant="2007-10-17T14:44:39.953Z">
      <AuthnContext>
        <AuthnContextClassRef>
          urn:oasis:names:tc:SAML:2.0:ac:classes:Password
        </AuthnContextClassRef>
      </AuthnContext>
    </AuthnStatement>
  </Assertion>

  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    <SignedInfo>
      <CanonicalizationMethod Algorithm="http://www.w3.org/2001/10/xml-exc-c14n#" />
      <SignatureMethod Algorithm="http://www.w3.org/2000/09/xmldsig#dsa-sha1" />
      <Reference URI="">
        <Transforms>
          <Transform Algorithm="http://www.w3.org/2000/09/xmldsig#enveloped-signature" />
        </Transforms>
        <DigestMethod Algorithm="http://www.w3.org/2000/09/xmldsig#sha1" />
        <DigestValue>BR95pMxuDchpwQdSZkcHSfxi4CE=</DigestValue>
      </Reference>
    </SignedInfo>
    <SignatureValue>B7P2mOLYn1/JkWQtc54qpnrgwusXF2OwnDXtavYDbsybf5d8vVne0Q==</SignatureValue>
  </Signature>

</SAMLResponse>
```

This Response message is base64-encoded and sent to the SP’s SAML2 endpoint via a HTML form:

```
<form name="form-redirection"
action="http://localhost/org.eclipse.higgins.saml2idp.test/SAMLEndpoint" method="post">

  <input type="hidden" name="RelayState" value="Test relay state!!">
  <input type="hidden" name="SAMLResponse" value="
PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGluZz0iVVRGLTgiPz48U0FNTFJlc3BvbnNlIHhtbG5zPSJ1cm46b2FzaXM6bmFtZ
XM6dGM6U0FNTDoyLjA6cHJvdG9jb2wiIERlc3RpbmF0aW9uPSJodHRwOi8vbG9jYWxob3N0L29yZy5lY2xpcHNlLmhpZ2dpbn
Muc2FtbDJpZHAudGVzdC9TQU1MRW5kcG9pbnQiIElEPSIxMTkyNjMyMjc5OTUzIiBJc3N1ZUluc3RhbnQ9IjIwMDctMTAtMTd
UMTQ6NDQ6MzkuOTUzWiIgVmVyc2lvbj0iMi4wIj48U3RhdHVzPjxTdGF0dXNDb2RlIFZhbHVlPSJ1cm46b2FzaXM6bmFtZXM6
dGM6U0FNTDoyLjA6c3RhdHVzOlN1Y2Nlc3MiLz48L1N0YXR1cz48SXNzdWVyIHhtbG5zPSJ1cm46b2FzaXM6bmFtZXM6dGM6U
0FNTDoyLjA6YXNzZXJ0aW9uIj5UZXN0IFNBTUwyIElkUDwvSXNzdWVyPjxBc3NlcnRpb24geG1sbnM9InVybjpvYXNpczpuYW
1lczp0YzpTQU1MOjIuMDphc3NlcnRpb24iIElEPSIxMTkyNjMyMjc5OTUzIiBJc3N1ZUluc3RhbnQ9IjIwMDctMTAtMTdUMTQ
6NDQ6MzkuOTUzWiIgVmVyc2lvbj0iMi4wIj48SXNzdWVyPlRlc3QgU0FNTDIgSWRQPC9Jc3N1ZXI+PFN1YmplY3Q+PE5hbWVJ
RCBGb3JtYXQ9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjEuMTpuYW1laWQtZm9ybWF0OmVudGl0eSI+YWRtaW48L05hbWVJR
D48U3ViamVjdENvbmZpcm1hdGlvbiBNZXRob2Q9InVybjpvYXNpczpuYW1lczp0YzpTQU1MOjIuMDpjbTpiZWFyZXIiLz48L1
N1YmplY3Q+PEF1dGhuU3RhdGVtZW50IEF1dGhuSW5zdGFudD0iMjAwNy0xMC0xN1QxNDo0NDozOS45NTNaIj48QXV0aG5Db25
0ZXh0PjxBdXRobkNvbnRleHRDbGFzc1JlZj51cm46b2FzaXM6bmFtZXM6dGM6U0FNTDoyLjA6YWM6Y2xhc3NlczpQYXNzd29y
ZDwvQXV0aG5Db250ZXh0Q2xhc3NSZWY+PC9BdXRobkNvbnRleHQ+PC9BdXRoblN0YXRlbWVudD48L0Fzc2VydGlvbj48U2lnb
mF0dXJlIHhtbG5zPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjIj48U2lnbmVkSW5mbz48Q2Fub25pY2FsaX
phdGlvbk1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly93d3cudzMub3JnLzIwMDEvMTAveG1sLWV4Yy1jMTRuIyIvPjxTaWduYXR
1cmVNZXRob2QgQWxnb3JpdGhtPSJodHRwOi8vd3d3LnczLm9yZy8yMDAwLzA5L3htbGRzaWcjZHNhLXNoYTEiLz48UmVmZXJl
bmNlIFVSST0iIj48VHJhbnNmb3Jtcz48VHJhbnNmb3JtIEFsZ29yaXRobT0iaHR0cDovL3d3dy53My5vcmcvMjAwMC8wOS94b
Wxkc2lnI2VudmVsb3BlZC1zaWduYXR1cmUiLz48L1RyYW5zZm9ybXM+PERpZ2VzdE1ldGhvZCBBbGdvcml0aG09Imh0dHA6Ly
93d3cudzMub3JnLzIwMDAvMDkveG1sZHNpZyNzaGExIi8+PERpZ2VzdFZhbHVlPllac1VuVXFOVm94V3RjYjhhSjVjUTBQYU1
PVT08L0RpZ2VzdFZhbHVlPjwvUmVmZXJlbmNlPjwvU2lnbmVkSW5mbz48U2lnbmF0dXJlVmFsdWU+Sjd4UVJxWkhJTXQ2MERV
WTdqM3ZMNSt5cXVwZDRDb1loQkJ0a0hzTE8zelBTcUwweTZPM0xBPT08L1NpZ25hdHVyZVZhbHVlPjwvU2lnbmF0dXJlPjwvU
0FNTFJlc3BvbnNlPg==">

  <input type="submit" value="Continue...">
</form>
```

This form appears on a simple screen in the user's browser and is auto-submitted via JavaScript.

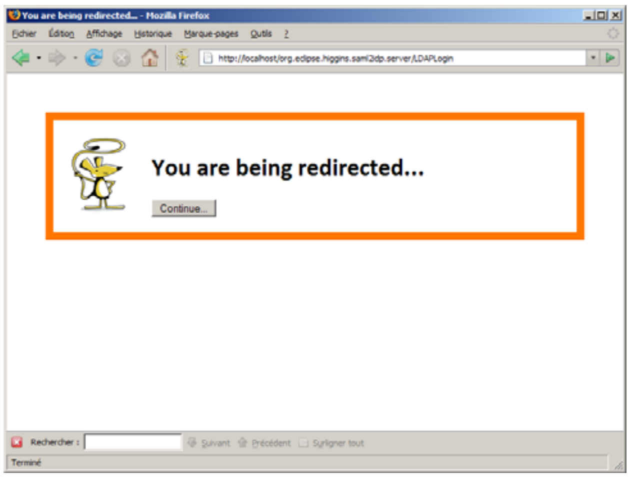
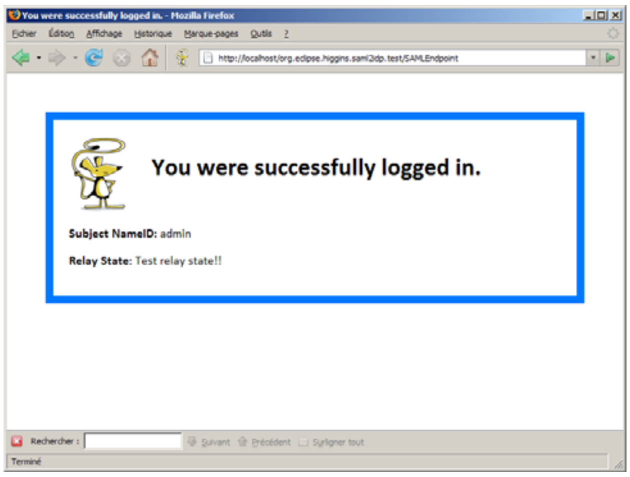

You are being redirected... - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://localhost/org.eclipse.higgins.saml2idp.server/LDAPLogin
You are being redirected...
Continue...
Rechercher : Suivant Précédent Surligner tout
Terminé

Figure 4: The IdP's SAML2 Response is transmitted to the SP via the SAML2 HTTP POST binding.

The SP receives and checks the SAML2 Response message. In particular, the XML Signature is verified using the preconfigured DSA public key.

If the signature is valid and the SAML2 status code provided by the IdP indicates successful authentication, the SP allows the user access to secured resources. The SAML2 Assertion contained in the response contains various useful information such a NameID of the logged in user.

The original RelayState provided by the SP is preserved; e.g. it may be used to redirect the user to the originally request URL.


You were successfully logged in. - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://localhost/org.eclipse.higgins.saml2idp.test/SAMLEndpoint
Rechercher : Suivant Précédent Surligner tout
Terminé

You were successfully logged in.

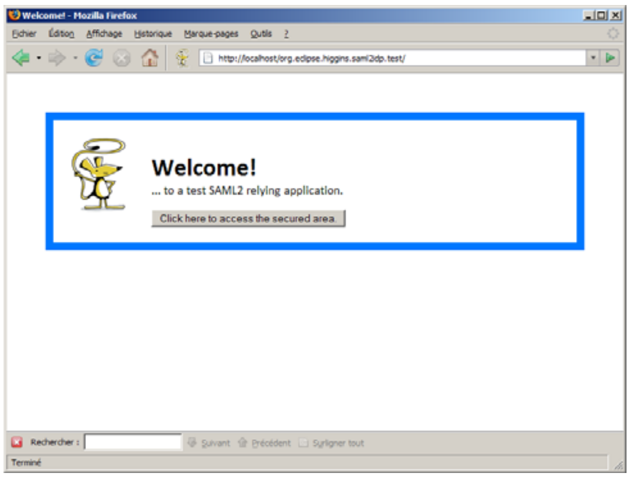
Subject NameID: admin

Relay State: Test relay state!!

Figure 5: The user now has access to the SP's restricted resources.

Flow #2: Subsequent logins

After the user is authenticated at the IdP, the IdP stores this information in the user's session state and will no longer require the user to provide credentials when the SP issues a SAML2 AuthnRequest message.

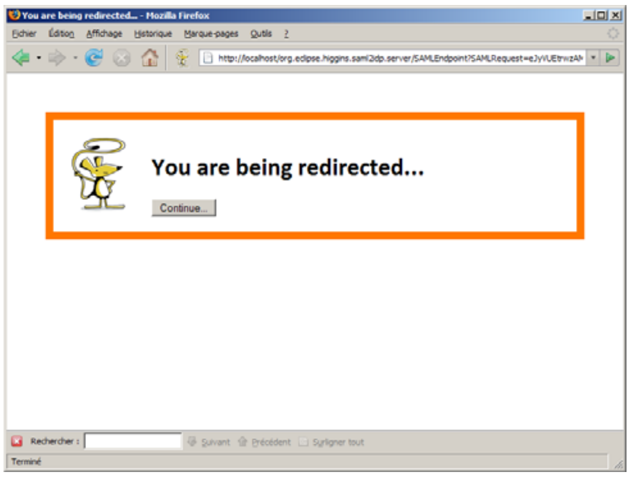

Welcome! - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
http://localhost/org.eclipse.higgins.saml2idp.test/
Click here to access the secured area.
Rechercher : Suivant Précédent Surligner tout
Terminé

Welcome!

... to a test SAML2 relying application.

Figure 6: The user wants to access the SP again.

As before, the SP issues a SAML2 AuthnRequest message to the IdP (via the HTTP Redirect binding), only this time no credentials are required, and the SAML2 Response is sent back to the SP immediately (via the HTTP POST binding).

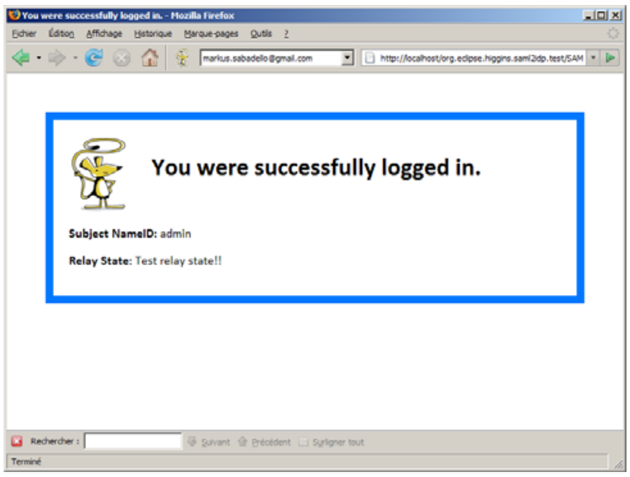

You are being redirected... - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
Continue...
Rechercher : Suivant Précédent Surligner tout
Terminé

http://localhost/org.eclipse.higgins.saml2idp.server/SAMLEndpoint?SAMLRequest=eJyVUEtrwzAM

You are being redirected...

Figure 7: The IdP receives the SP's SAML2 AuthnRequest and immediately answers with a SAML2 Response.

As before, XML Signatures and other aspects of the SAML2 messages are verified, and the user can access the SP’s restricted resources once again.


You were successfully logged in. - Mozilla Firefox
Fichier Édition Affichage Historique Marque-pages Outils ?
Rechercher : Suivant Précédent Surligner tout
Terminé

markus.sabadello@gmail.com

http://localhost/org.eclipse.higgins.saml2idp.test/SAM

You were successfully logged in.

Subject NameID: admin

Relay State: Test relay state!!

Figure 8: The user is logged in again at the SP.

Configuration

Higgins components are configured using the Higgins Configuration API. Configuration settings on the IdP’s side include the following:

- A human-friendly name for the IdP that is used in the SAML2 <Issuer> element.
- The DSA private key used to sign SAML2 messages.
- IdAS-related settings needed to access the Higgins context against which users are validate.

Logging

The Apache Commons Logging API is used for logging purposes. All activity at both the Higgins SAML2 IdP and the Higgins Test SAML2 SP is logged. The following text displays logging messages in a typical protocol flow as outlined in this document:

```
[SP]  INFO: Sending SAML2 AuthnRequest to IdP.
[IdP] INFO: User is not logged in. Displaying credentials form for context type $context+ldap.
[IdP] ATTENTION: Cannot login user: badguy
[IdP] INFO: User admin logged in. Sending SAML2 Response to SP.
[SP]  INFO: SAML2 Response XML Signature verified.
[SP]  INFO: SAML2 Response StatusCode: urn:oasis:names:tc:SAML:2.0:status:Success
[SP]  INFO: User successfully logged in.
[SP]  INFO: Sending SAML2 AuthnRequest to IdP.
[IdP] INFO: User is logged in already. Sending SAML2 Response to SP.
[SP]  INFO: SAML2 Response XML Signature verified.
[SP]  INFO: SAML2 Response StatusCode: urn:oasis:names:tc:SAML:2.0:status:Success
[SP]  INFO: User successfully logged in.
```

Future Work

The following issues need to be addressed in future versions:

- Create more Higgins context providers to be able to authenticate against Kerberos, RADIUS, etc.
- Improve security features (e.g. prevent password guessing attacks, run more checks on the SAML2 messages).
- Improve the Higgins Configuration API so that existing configuration can be modified programmatically.
- SAML2 support is very limited. Support for more SAML2 features (assertions, authentication contexts etc.) is highly desirable.