Automated Testing of SAML 2.0 Service Providers

Andreas Åkre Solberg
UNINETT
andreas@uninett.no
http://rnd.feide.no
Background

› 0% of SAML 2.0 implementations do SAML 100% correct.
› SAML includes a lot of options.
› There is special case flows and messages, where SAML does not provide a well-defined correct behaviour.
› SAML includes extensions points.
› Limited negotiation of what an entity supports (SAML Metadata)
Heterogenous federations

› In Shibboleth 1.3-federations most of entities were running Shibboleth software.

Why?
– Shibboleth defined its own protocol. (SAML 1.1 extension)

In Feide we have these SP softwares:
› SimpleSAMLphp, Shibboleth, mod_mellon, Sun OpenSSO, Sun FM, Sun AM, Microsoft ADFS, Novell Access Manager, SAML2API (NTNU), ComponentSpace (.net), OIOSAML, several (5+) 'home made' solutions. (+ multiple versions)

› Other federations will experience an explosion of SP software soon
Confederations

SP <-> IdP connections becomes very large when federations are interconnected.

25 SP/IdP implementions, each in 5 versions, becomes:
› **15,625** different combinations of software.
That's only software / versions, deployments also is configured differently.

No good test framework!

**End-users will be the ones to encounter interoperability problems.**
More and more Service Providers would embed SAML support into their product, and rely on SAML libraries rather than full software or alternatively implement from scratch.

These integrations is likely to be less interoperable.

Focus for a SAML product is often full SAML compliance.

Focus for an web application (with SAML support) is often compatibility with a specific IdP software, rather than compliance to SAML spec.

*Example:* Some org. pays some other org. to SAML-enable a product in order to connect it to Feide.
More advanced use

The more advanced usage of SAML, the more likely to encounter interoperability problems.

Encourage Simple use of SAML

• saml2int encourage choosing the options that is most likely to interoperate better. It also discourage the use of advanced stuff.

• Metadata Interoperable Profile.
Kantara
SAML 2.0 Full Matrix Test Event

SimpleSAMLphp have been preparing for participation at this test event for this autumn.

• Idea was to learn more about interoperability testing (from Drummond Group) and feed this back to the automated test tool.

• Kantara Full Matrix test event is probably far more friendlier than the automated test tool.
Automated Testing Tool

› Acts as an Identity Provider.

In order to test an SP:

1) SP provides:
   – SP Metadata
   – URL to initiate login
   – URL to show attributes
   – URL to initiate Single Logout
2) SP loads the IdP Metadata
3) Click the 'start test' button.
Automated Testing Tool

Register Federation Lab Metadata at your Service Provider

Configure your Service Provider to trust the IdP metadata below:

```xml
<?xml version="1.0"?>
  <md:KeyDescriptor use="signing">
    <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
      <ds:KeyInfo>
        <ds:KeyValue>
          <ds:X509Data>
            <ds:X509Certificate longName="MIIDUjCCARngAwIBAgIjJAKUThxcg0XMA0GCSqGSGjB3QDEBQUAUMH0xGzAJejBhNVAYTAk5PMK1wEAYDVQQIEwULc2Vzc2VzLmlud3MjMwNQYJK4MA8GS0wLDEYDAHMA0GCSQGSIb3QDEBQUAUMH0xGzAJejBhNVAYTAk5PMK1wEAYDVQQIEwULc2Vzc2VzLmlud3MjMwNQYJK4MA8GS0wLDEYDAHMA0GCSQGSIb3QDEBQUAUMH0xGzAJejBhNVAYTAk5PMK1wEAYDVQQIEwULc2Vzc2VzLmlud3MjMwNQYJK4MA8GS0wLDEYDAHMA0GCSqGSIb3QDEBQUAUMH0xGzAJejBhNVAYTAk5PMK1wEAYDVQQIEwULc2Vzc2VzLmlud3MjMwNQYJK4MA8GS0wLDEY" />
        </ds:X509Data>
      </ds:KeyInfo>
    </ds:KeyInfo>
  </md:KeyDescriptor>
</md:entityDescriptor>
</md:entityDescriptor>
```

If your service provider prefers to load the metadata from an URL instead, use this url:

- `http://fedlab.bridge.feide.no simplesxml/module.php/fedlab/?output=xml

I'm done. Let me register my SP »
Automated Testing Tool

Post metadata for your Service Provider

Paste in SAML 2.0 XML Metadata for the entity that you would like to add.

```xml
<EntityDescriptor
    xmlns="urn:oasis:names:tc:SAML:2.0:metadata"
    entityID="https://skjak2.uninett.no:443/openam">
    <SPSSODescriptor
        AuthnRequestsSigned="false"
        WantAssertionsSigned="false"
        protocolSupportEnumeration=
            "urn:oasis:names:tc:SAML:2.0:protocol">
        <SingleLogoutService
            Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
            Location="https://skjak2.uninett.no:443/openam/SPSIdRedirect/metaAlias/sp"/>
    </SPSSODescriptor>
</EntityDescriptor>
```

Enter an URL on the Service Provider that will initiate authentication with this IdP (without interaction):

https://skjak2.uninett.no/openam/olavtest/fedlab.jsp

Enter an URL on the Service Provider that will show attributes:

https://skjak2.uninett.no/openam/olavtest/fedlab.jsp

Enter an URL on the Service Provider that will initiate single logout (SP-initiated SAML 2.0 logout):


» Verify SP connectivity »
Automated Testing Tool

The testing tool will typically:

› Do a HTTP request to the URL initiating login
  – follow redirects, until it detects the hostname of the IdP
  – then parse the SAML Request from the Location: header.

› Then create a Response message, and sends directly to the SP over HTTP Post.
First experience with the tool

I've connected a few SP software, in order to implement the tool.

I'll present some of the results I've seen so far.

Disclaimer

There is a strong possibility that many of the tests is actually wrong.

Part of further work is to implement more tests, and perform quality control of the existing tests.
Sun OpenSSO Service Provider

› Not proper handling of AudienceRestrictions with multiple values
› SP is accepting SessionNotOnOrAfter set in the past!
› SP is not validating the Condition NotBefore and NotAfter attributes.
› SP is ignoring unknown Conditions - it should not.
› SP is ignoring client Address attributes.
› SP is not accepting persistent or e-mail NameID formats
› SP is ignoring the DestinationURL attribute
› SP is not requiring signature on LogoutRequest
› SP ignores the Destination of LogoutRequest
› SP does not handle multiple SubjectConfirmation recipients
› SP ignores the NameID format and SPnamequalifier in a LogoutRequest
› SP does not handle LogoutRequests send before Assertion.
Ping Federate

› Insecure handling of multiple AudienceRestrictions
› Should not have accepted an empty Audience
› SP ignores the SubjectConfirmationData @ NotOnOrAfter
› SP ignores Condition NotBefore and NotAfter
› SP ignores DestinationURL in the assertion.
› SP does not properly handle invalid InResponseTo in Assertion

Very few tests were run against Ping Federate, as we lost our test environments.
mod_mellon (Lasso)

› Insecure handling of multiple AudienceRestrictions
› Accepts Response without AuthenticationStatement!
› Ignores SubjectConfirmationData @ NotOnOrAfter
› Ignores Condition @ NotBefore and NotAfter
› Insecure handling of unknown Condition
› Ignoring client IP Address Condition if provided
› Ignoring DestinationURL in Response
› Not proper handling of invalid InResponseTo values
› Not protection against Response replay
› Does not handle multiple Assertions or AttributeStatements
› Accepts unsigned LogoutRequests
› Ignores alot of stuff in the LogoutRequest!
› Does not cache LogoutRequest sent before Assertion
Shibboleth 2.X

› All Condition checks fails. Seems to completely ignore Condition.
  › Ignores Audience
  › Ignores NotBefore and NotOnOrAfter
  › Insecure handling of unknown Conditions
  › Ignoring client IP address.
› Not proper handling of invalid InResponseTo values.
› SP does not handle LogoutRequest with multiple SessionIndexes
› SP does not accept LogoutRequest without SessionIndex
› SP does not accept LogoutRequest sent in separate session.
› SP Does not handle LogoutRequest sent before Assertion.
Siemens kindly offered to connect their product to the test environments.

During the test period, they have significantly improved their product.

*Test results not offered public.*
# SimpleSAMLphp

Before

<table>
<thead>
<tr>
<th>Condition</th>
<th>Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session fixation check</td>
<td></td>
</tr>
<tr>
<td>SP MUST accept LogoutRequest with sessionindex in a separate session, not relying on the session-cookie.</td>
<td></td>
</tr>
<tr>
<td>SP MUST accept an LogoutRequest with no sessionindex (sent in separate session, no session-cookies).</td>
<td></td>
</tr>
<tr>
<td>SP MUST accept an LogoutRequest with two sessionindices (first valid) (sent in separate session, no session-cookies).</td>
<td></td>
</tr>
<tr>
<td>SP MUST accept an LogoutRequest with two sessionindices (second valid) (sent in separate session, no session-cookies).</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT accept LogoutRequest when NameID content is wrong</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT accept LogoutRequest when NameID@Format is wrong</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT accept LogoutRequest when NameID@SPNNameQualifier is wrong</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT logout user when Invalid SessionIndex is sent</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT accept LogoutRequest when Destination is wrong</td>
<td></td>
</tr>
<tr>
<td>SP MUST NOT accept unsigned LogoutRequest</td>
<td></td>
</tr>
<tr>
<td>SP SHOULD find attributes in a second AssertioN/AttributeStatement, not only in one of them (last 1 of 2 - attributes in first).</td>
<td></td>
</tr>
<tr>
<td>SP SHOULD find attributes in a second AssertioN/AttributeStatement, not only in one of them (last 2 of 2 - attributes in last).</td>
<td></td>
</tr>
<tr>
<td>SP should accept a Response with two SubjectConfirmation elements representing two recipients (test 1 of 2, correct one last).</td>
<td></td>
</tr>
<tr>
<td>SP should accept a Response with two SubjectConfirmation elements representing two recipients (test 1 of 2, correct one first).</td>
<td></td>
</tr>
<tr>
<td>SP should not accept a Response with a SubjectConfirmationData elements with a incorrect @Address attribute</td>
<td></td>
</tr>
<tr>
<td>SP should accept a Response with multiple SubjectConfirmation elements with /SubjectConfirmationData/@Address-es, where one is correct (last 1 of 2, correct one last).</td>
<td></td>
</tr>
<tr>
<td>SP should accept a Response with multiple SubjectConfirmation elements with /SubjectConfirmationData/@Address-es, where one is correct (last 1 of 2, correct one first).</td>
<td></td>
</tr>
<tr>
<td>SP should not accept a Response with a AuthnStatement missing</td>
<td></td>
</tr>
<tr>
<td>SP should not accept an IssueInstant for (24 hours) into the future</td>
<td></td>
</tr>
<tr>
<td>SP should not accept an IssueInstant for (24 hours) into the past</td>
<td></td>
</tr>
</tbody>
</table>
SimpleSAMLphp

Now

Remaining:
  › LogoutRequest send before Assertion

This test sends IdP initiated LogoutRequest to SP before the authn Response.

SP should not accept an IssueInstant far (24 hours) into the future

SP should not accept an IssueInstant far (24 hours) into the past
Additional security features

Not only testing for interoperability problems, but also security issues:

› Testing for session fixation (in work)
  › Do a login for discovering SSO cookie
  › Wipe cookies
  › Set a custom fixed SSO cookie
  › Do a login
  › Wipe cookies
  › Re-introduce fixed cookie
  › Check if having access to service!
› Check for secure cookies (in work)
› Security test: validating of SP follows processing rules.
Signing and encryption

I got very few tests on various ways of sign and encrypt.

I'm pretty sure the results would have been very interesting.

› X.509
› XMLdsig
Demo time
What's next?

› Connect more SP products.
  › I want ADFS, pySAML, Ping, and others...
› Complete code re-write.
› A good name?
› Made publicly available on GÉANT Federation Lab
› Quality control of tests
› Categorization of tests + custom priorities

› IdP testing tool!