

# 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 are special case flows and messages, where SAML does not provide a well-defined correct behaviour.
- › SAML includes extension 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 implementations, 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.**

# Home-made

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

Configure your SP Register your SP Prepare for testing Running tests

## Register Federation Lab Metadata at your Service Provider

Configure your Service Provider to trust the IdP metadata below:

```
<?xml version="1.0"?>
<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metadata" xmlns:ds="http://www.w3.org/2000/09/xmldsig#" entityID="ht
  <md:IDPSSODescriptor protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol">
    <md:KeyDescriptor use="signing">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>MIIDUjCCArugAwIBAgIJAKUTNwcdpKB/MA0GCSqGSIB3DQEBBQUAMHoxCzAJBgNVBAYTAk5PMRIwEAYDVQQIEwluUcm9uZGhlaW
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:KeyDescriptor use="encryption">
      <ds:KeyInfo xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
        <ds:X509Data>
          <ds:X509Certificate>MIIDUjCCArugAwIBAgIJAKUTNwcdpKB/MA0GCSqGSIB3DQEBBQUAMHoxCzAJBgNVBAYTAk5PMRIwEAYDVQQIEwluUcm9uZGhlaW
        </ds:X509Data>
      </ds:KeyInfo>
    </md:KeyDescriptor>
    <md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="http://fedlab.bridge.feide.no
    <md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect" Location="http://fedlab.bridge.feide.no
  </md:IDPSSODescriptor>
  <md:ContactPerson contactType="technical">
    <md:SurName>Administrator</md:SurName>
    <md:EmailAddress>na@example.org</md:EmailAddress>
  </md:ContactPerson>
</md:EntityDescriptor>
```

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

- <http://fedlab.bridge.feide.no/simplesaml/module.php/fedlab/?output=xml>

I'm done. Let me register my SP »

# Automated Testing Tool

Configure your SP Register your SP Prepare for testing Running tests

## Post metadata for your Service Provider

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

```
<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/SPSloRedirect/metaAlias/sp">
```

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

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

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

# 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 proper handle invalid InResponseTo in Assertion

Very few tests was 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 IdM product

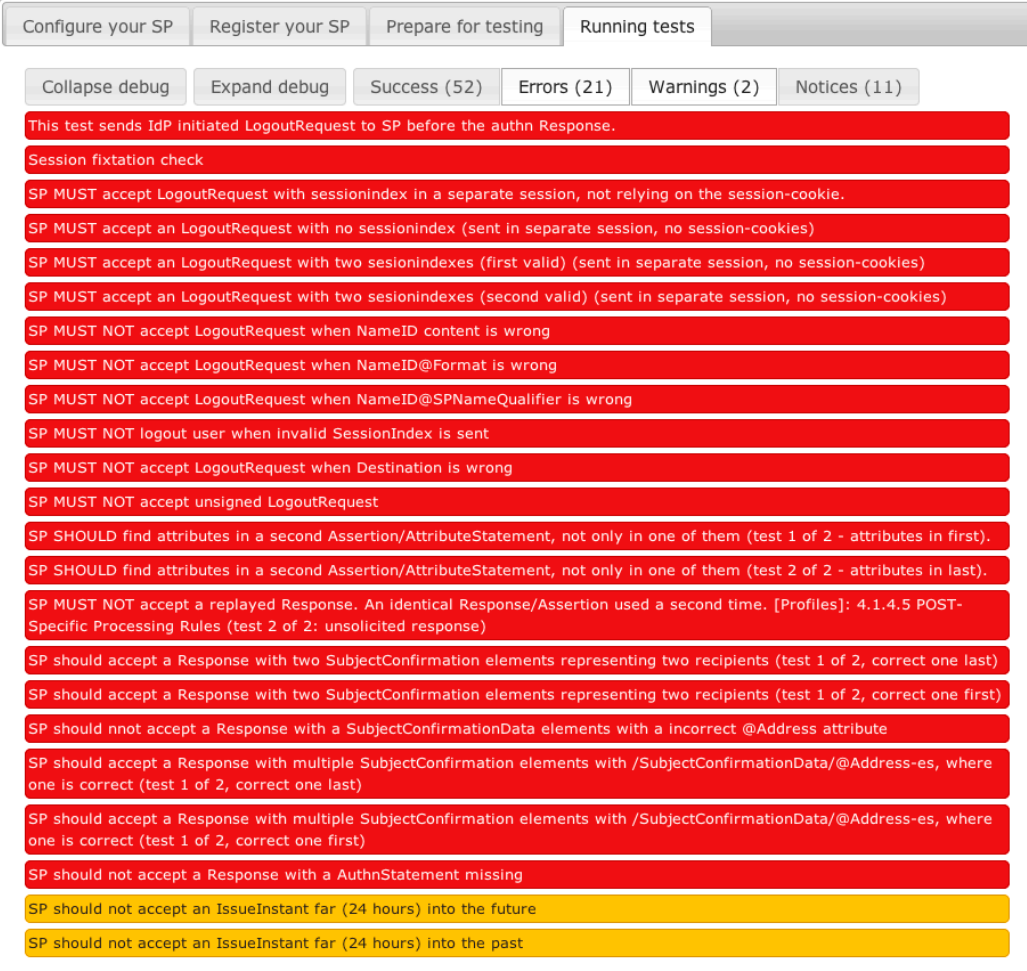
Siemens kindly offered to connect their product to the test environments.

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

*Test results not offered public.*

# SimpleSAMLphp

Before



The screenshot displays the 'Running tests' tab of the SimpleSAMLphp test runner. It features a navigation bar with tabs for 'Configure your SP', 'Register your SP', 'Prepare for testing', and 'Running tests'. Below the navigation bar, there are buttons for 'Collapse debug', 'Expand debug', and summary counts for 'Success (52)', 'Errors (21)', 'Warnings (2)', and 'Notices (11)'. The main area contains a list of 23 test cases, each with a red background and white text. The test cases describe various scenarios for IdP initiated LogoutRequests, including session fixation checks, sessionindex handling, NameID content and format validation, NameID@SPNameQualifier validation, invalid SessionIndex, Destination validation, unsigned LogoutRequests, attribute finding in Assertion/AttributeStatement, replayed Responses, SubjectConfirmation elements, SubjectConfirmationData elements, AuthnStatement missing, and IssueInstant far into the future or past.

Configure your SP | Register your SP | Prepare for testing | Running tests

Collapse debug | Expand debug | Success (52) | Errors (21) | Warnings (2) | Notices (11)

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

Session fixation check

SP MUST accept LogoutRequest with sessionindex in a separate session, not relying on the session-cookie.

SP MUST accept an LogoutRequest with no sessionindex (sent in separate session, no session-cookies)

SP MUST accept an LogoutRequest with two sessionindexes (first valid) (sent in separate session, no session-cookies)

SP MUST accept an LogoutRequest with two sessionindexes (second valid) (sent in separate session, no session-cookies)

SP MUST NOT accept LogoutRequest when NameID content is wrong

SP MUST NOT accept LogoutRequest when NameID@Format is wrong

SP MUST NOT accept LogoutRequest when NameID@SPNameQualifier is wrong

SP MUST NOT logout user when invalid SessionIndex is sent

SP MUST NOT accept LogoutRequest when Destination is wrong

SP MUST NOT accept unsigned LogoutRequest

SP SHOULD find attributes in a second Assertion/AttributeStatement, not only in one of them (test 1 of 2 - attributes in first).

SP SHOULD find attributes in a second Assertion/AttributeStatement, not only in one of them (test 2 of 2 - attributes in last).

SP MUST NOT accept a replayed Response. An identical Response/Assertion used a second time. [Profiles]: 4.1.4.5 POST-Specific Processing Rules (test 2 of 2: unsolicited response)

SP should accept a Response with two SubjectConfirmation elements representing two recipients (test 1 of 2, correct one last)

SP should accept a Response with two SubjectConfirmation elements representing two recipients (test 1 of 2, correct one first)

SP should not accept a Response with a SubjectConfirmationData elements with a incorrect @Address attribute

SP should accept a Response with multiple SubjectConfirmation elements with /SubjectConfirmationData/@Address-es, where one is correct (test 1 of 2, correct one last)

SP should accept a Response with multiple SubjectConfirmation elements with /SubjectConfirmationData/@Address-es, where one is correct (test 1 of 2, correct one first)

SP should not accept a Response with a AuthnStatement missing

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

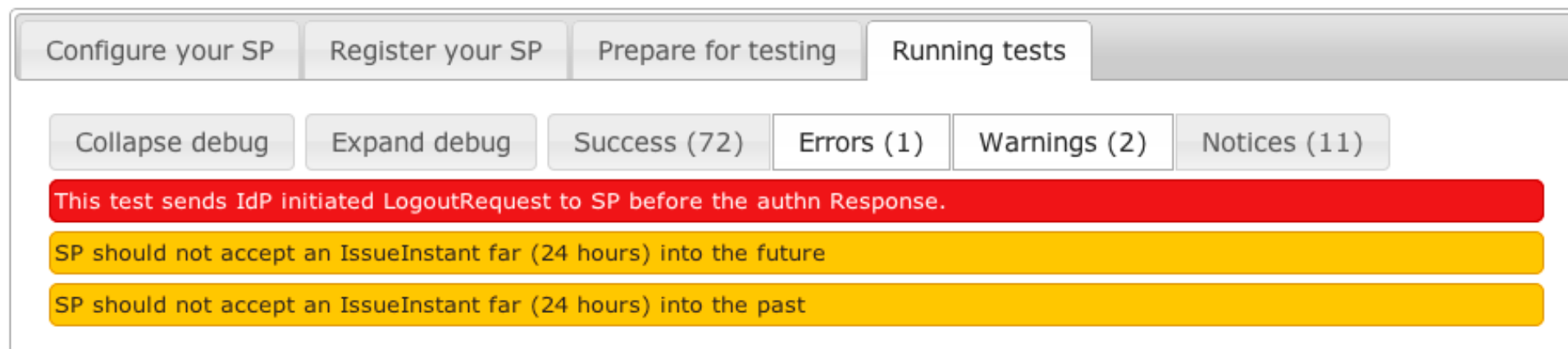
SP should not accept an IssueInstant far (24 hours) into the past

# SimpleSAMLphp

Now

Remaining:

› LogoutRequest send before Assertion



The screenshot shows the SimpleSAMLphp test interface. At the top, there is a navigation bar with five tabs: "Configure your SP", "Register your SP", "Prepare for testing", "Running tests", and a greyed-out "Running tests" tab. Below the navigation bar, there are six buttons: "Collapse debug", "Expand debug", "Success (72)", "Errors (1)", "Warnings (2)", and "Notices (11)". The "Errors (1)" button is highlighted, and a red bar displays the error message: "This test sends IdP initiated LogoutRequest to SP before the authn Response." Below this, two yellow bars display warnings: "SP should not accept an IssueInstant far (24 hours) into the future" and "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!



<http://rnd.feide.no>