Liberty ID-WSF Multi-Device SSO Deployment Guide

Version: 1.0-02

Editors:
Paul Madsen, NTT

Contributors:
Hiroki Itoh, NTT
Kiyohiko Ishikawa, NHK
Fujii Arisa, NHK

Abstract:

This document profiles how to use the Liberty Identity Web Services Framework (ID-WSF) to support single sign-on for users that crosses devices, i.e. the session is initiated from one device or user-agent, and subsequently transferred to a second, as might be desirable in the enjoyment of long running media, e.g. streaming video.

Filename: draft-liberty-idwsf-mdsso-deployguide-v1.0-02.pdf

Notice

This document has been prepared by Sponsors of the Liberty Alliance. Permission is hereby granted to use the document solely for the purpose of implementing the Specification. No rights are granted to prepare derivative works of this Specification. Entities seeking permission to reproduce portions of this document for other uses must contact the Liberty Alliance to determine whether an appropriate license for such use is available.

Implementation or use of certain elements of this document may require licenses under third party intellectual property rights, including without limitation, patent rights. The Sponsors of and any other contributors to the Specification are not and shall not be held responsible in any manner for identifying or failing to identify any or all such third party intellectual property rights. **This Specification is provided "AS IS," and no participant in the Liberty Alliance makes any warranty of any kind, express or implied, including any implied warranties of merchantability, non-infringement of third party intellectual property rights, and fitness for a particular purpose.** Implementers of this Specification are advised to review the Liberty Alliance Project's website (http://www.projectliberty.org/) for information concerning any Necessary Claims Disclosure Notices that have been received by the Liberty Alliance Management Board.

Liberty Alliance Project
Licensing Administrator
c/o IEEE-ISTO
445 Hoes Lane
Piscataway, NJ 08855-1331, USA
info@projectliberty.org

Contents

1. Introduction

Multi-Device SSO (MD SSO) refers to users enjoying single sign-on across multiple devices and/or user agents - thereby able to enjoy uninterrupted delivery from service provides to those devices.

The following is a representative scenario for the MD SSO use case:

1. While commuting home from work, Alice uses her mobile to browse free media content.
2. Alice begins watching a movie on mobile while riding the bus.
3. As she nears her bus stop, Alice decides to watch the rest of the movie on her home HD.
   She stops the movie and purchases HD version.
4. IdP transmits Alice's identity to SP.
5. On arriving home, Alice swipes her phone by her set top box.
   Alice watches the rest of movie from where she had previously stopped watching.

More generally, the user would be able to access different SPs from the second device, but with the same degree of cross-device convenience.

1.1. Namespaces

This document uses the following namespaces:

- The prefix `xs:` stands for the W3C XML schema namespace (`http://www.w3.org/2001/XMLSchema`) [Schema1-2].
- The prefix `xml:` stands for the W3C XML namespace (`http://www.w3.org/XML/1998/namespace`) [XML].
- The prefix `saml:` stands for the OASIS SSTC SAML2.0 Assertion namespace (`urn:oasis:names:tc:SAML:2.0:assertion`) [SAMLCore2].
- The prefix `samlp:` stands for the OASIS SSTC SAML2.0 Protocol namespace (`urn:oasis:names:tc:SAML:2.0:protocol`) [SAMLCore2].
- The prefix `ims:` stands for the Liberty ID-WSF Authentication Service Identity Mapping Service namespace (`urn:liberty:ims:2006-08`) [LibertySOAPAuthn].
- The prefix `sec:` stands for the Liberty ID-WSF Security Mechanisms Core namespace (`urn:liberty:sec:2005-11`) [LibertySecMech].
- The prefix `as:` stands for the Liberty ID-WSF Authentication Service namespace (`urn:liberty:as:2005-11`) [LibertySecMech].

2. Security Context

Enabling SSO across devices implies the transfer of a security context from the first device to the second. By using the passed security context, the second device will be able to re-establish a new security session on the behalf of the user.

An <Assertion> carried within the <SecurityContext> element of the <Metadata> element of an ID-WSF <EndPointReference> captures the invocation context necessary for interacting with the SSOS service instance represented by the containing ID-WSF EPR.

3. Application Context

For a user to be able to enjoy 'uninterrupted' service at some SP from one device to another implies that the application context (i.e. what they were doing) they ended at the first device can be reestablished at the second. If the server is unable to track this (and thereby free the device/clients from the burden), it will be necessary for such context to be transferred from the first device to the second.

Such application context MAY be optionally passed as an <ApplicationContext> element.

```
<xs:element name="ApplicationContext">
 <xs:complexType>
 <xs:sequence>
 <xs:any namespace="##other"
 processContents="lax"
 minOccurs="0"
 maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
```

The specifics of the context to capture and share between devices will depend on the application, e.g. the relevant data would be very different for a game than for a streaming video. The <ApplicationContext> schema is defined to allow such flexibility.

4. Transfer Context

The security and optional application context are packaged for transfer in a <TransferContext> element.

```
<xs:element name="TransferContext">
 <xs:complexType>
 <xs:sequence>
 <xs:element ref="wsa:EndpointReference"/>
 <xs:element ref="ApplicationContext" minOccurs="0"/>
 <xs:any namespace="##other"
 processContents="lax"
 minOccurs="0"
 maxOccurs="unbounded"/>
 </xs:sequence>
 </xs:complexType>
</xs:element>
```

It is the <TransferContext> element that is sent from the first device to the second.

5. Transfer Mechanism

The specifics of how the security and application context are passed from the first device to the second are not defined by this profile.

Different options (e.g. BlueTooth, Near Field Communication, etc) may have different security characteristics for interception of the SSOS EPR and embedded SAML Assertion.

6. Profile of ID-WSF for Multi-Device SSO

The following sequence profiles the use of Liberty ID-WSF Authentication and Single Sign-On Services in combination to support the Multi-Device SSO Use Case:

1. User authenticates to IDP from Device A through AS (using a SASL negotiated mechanism)

```
<soap:Envelope>
 <soap:Body>
 <SASLRequest mechanism="GSSAPI">
  <Data>
  Q29ub3IgQ2FoaWxsIGNhc3VhbGx5IG1hbm dsZXMgcGFzc3dvcmRzCg==
  </Data>
 </SASLRequest>
 </soap:Body>
</soap:Envelope>
```

2. IDP returns to Device A the SSOS EPR. The IDP MAY also return the DS EPR.

   The IDP SHOULD ensure that the lifetimes of the EPR and embedded SAML Assertion are sufficiently long to allow them to be transferred to the second device before their expiration.

   The IDP SHOULD set the ConfirmationMethod of the SubjectConfirmation of the embedded SAML Assertion as 'urn:oasis:names:tc:SAML:1.0:cm:bearer'.

```
<soap:Envelope>
<soap:Body>
<sa:SASLResponse xmlns:sa="urn:liberty:sa:2004-04"
xmlns:disco="urn:liberty:disco:2003-08">
<Status code="sa:OK"/>

<wsa:EndpointReference>
 <wsa:Address>
 http://tg2.example.com:8080/tfs-soap/IdPSSOService
 </wsa:Address>
 <wsa:Metadata>
  <disco:ServiceType>urn:liberty:sso s:2003-08</disco:ServiceTyp e>
  <disco:ProviderID>http://ssos.example.com:8080</di sco:ProviderID>
  <ds:SecurityContext>
   <disco:SecurityMechID>
   urn:liberty:security:2005-02:null:Be arer
   </disco:SecurityMechID>
   <sec:Token>
    <saml2:Assertion
    ID="i1b42508103cab657f34e5ef189 f28ea10dd86926 "
    Version="2.0"
    IssueInstant="2004-02-03T22:12:33Z">
     <Issuer>http://idp.example.com:8080</Issuer>
    </saml2:Assertion>
   </sec:Token>
  </ds:SecurityContext>
 </wsa:Metadata>
</wsa:EndpointReference>

</sa:SASLResponse>
</soap:Body>
</soap:Envelope>
```

3. Device A uses the SSOS EPR to request and obtain SAML Assertions for presentation to SP1 (and other SPs).

   Device A sends an <saml:AuthnRequest> to the endpoint within the SSOS EPR, using as a security token the SAML Assertion from the <ds:SecurityContext> element in the EPR.

```
<soap:Envelope>
 <soap:Header>
  <wse:Security>
   <saml2:Assertion
     ID="i1b42508103cab657f34e5ef189f28ea10dd86926 "
     Version="2.0"
     IssueInstant="2004-02-03T22:12:33Z">
      <Issuer>http://idp.example.com:8080</Issuer>
   </saml2:Assertion>
  </wse:Security>
 </soap:Header>
 <soap:Body>
  <samlp:AuthnRequest>
   <saml2:AudienceRestriction>
    <saml2:Audience>http://sp1.example.com</saml2:Audience>
   </saml2:AudienceRestriction>
  </samlp:AuthnRequest>
 </soap:Body>
</soap:Envelope>
```

4. The SSOS returns to Device A a SAML Assertion targeted at SP1.

5. After presenting SSO Assertion to SP1 (specifics will depend on the binding), user enjoys service at SP1.

6. Some time later (e.g when initiated by user selecting 'Move Session to Device B'), Device A prepares a package for delivery to Device B.

   If Device A is not adding application context, the SSOS EPR that Device A obtained, placed within the <TransferContext> element, constitutes the package.

   The SSOS EPR MUST be generated according to the rules of the ID-WSF 2.0 Discovery Service specification.

   Device A MAY supplement the SSOS EPR within the <TransferContext> with appropriate application context. If doing so, Device A MUST insert the appropriate <ApplicationContext> element following the <EndpointReference> element within the <TransferContext> element.

```
<mdsso:TransferContext>
 <wsa:EndpointReference>
  <wsa:Address>
  http://ssos.example.com:8080/IdPSSOService
  </wsa:Address>
  <wsa:Metadata>
   <disco:ServiceType>urn:liberty:ssos:2003-08</disco:ServiceType>
   <disco:ProviderID>http://ssos.example.com:8080</disco:ProviderID>
   <ds:SecurityContext>
    <disco:SecurityMechID>
    urn:liberty:security:2005-02:null:Bearer
    </disco:SecurityMechID>
    <sec:Token>
     <saml2:Assertion
     ID="i1b42508103cab657f34e5ef189f28ea10dd86926 "
     Version="2.0"
     IssueInstant="2004-02-03T22:12:33Z">
      <Issuer>http://idp.example.com:8080</Issuer>
     </saml2:Assertion>
    </sec:Token>
   </ds:SecurityContext>
  </wsa:Metadata>
```

```
 </wsa:EndpointReference>
 <mdsso:ApplicationContext>
  <vid:AppData>
    <vid:Name>King Kong</vid:Name>
    <vid:Time>1:42:7</vid:Time>
  </vid:AppData>
 </mdsso:ApplicationContext>
</mdsso:TransferContext>
```

7. Device A sends the <TransferContext> package to Device B.

8. Device B uses the SSOS EPR token received from Device A at the SSOS endpoint to obtain SSO Assertions for presentation to SPs.

```
<soap:Envelope>
 <soap:Header>
  <wse:Security>
   <saml2:Assertion
     ID="i1b42508103cab657f34e5ef189f28ea10dd86926 "
     Version="2.0"
     IssueInstant="2004-02-03T22:12:33Z">
      <Issuer>http://idp.example.com:8080</Issuer>
   </saml2:Assertion>
  </wse:Security>
 </soap:Header>
 <soap:Body>
  <samlp:AuthnRequest>
   <saml2:AudienceRestriction>
    <saml2:Audience>http://sp1.example.com</saml2:Audience>
   </saml2:AudienceRestriction>
  </samlp:AuthnRequest>
 </soap:Body>
</soap:Envelope>
```

9. SSOS returns a SAML Assertion to Device B targeted at relevant SPs.

10. With the SAML Assertion returned by the IDP, User enjoys access to SP1 resources from Device B (or other SPs).

11. Device B MAY use any application context delivered to it within the <TransferContext> package to reestablish any application context that was terminated at Device A.

```
<mdsso:ApplicationContext>
  <vid:AppData>
    <vid:Name>King Kong</vid:Name>
    <vid:Time>1:42:7</vid:Time>
  </vid:AppData>
</mdsso:ApplicationContext>
```

How Device B reestablishes application context is out of scope.

7. Security Considerations

Were the SAML assertion within the <EndpointReference> passed from Device A to Device B be intercepted by an attacker, it would serve as a bearer token for that attacker t0 impersonate the user at the SSOS and, as such, possibly at federated SPs. Depending on the security characteristics of the mechanism used for transferring the <TransferContext> package from one device to another, this risk may be significant.

To mitigate the risk of interception, the SAML assertion could be constrained such that it could be presented only be Device B (by, for instance, using a SAML holder of key confirmation method) or such that it could only be used at the SSOS to request a secondary assertion targeted at a particular SP. Support for such enhanced security will be explored in subsequent drafts of these guidelines.

References

Normative

[LibertySecMech] Hirsch, Frederick, eds. "Liberty ID-WSF Security Mechanisms Core," Version 2.0-errata-v1.0, Liberty Alliance Project (21 April, 2007). *http://www.projectliberty.org/specs*

[LibertySOAPAuthn] Hodges, Jeff, Aarts, Robert, Madsen, Paul, Cantor, Scott, eds. " Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification ," v2.0-errata-1.0-01, Liberty Alliance Project (28 November, 2006). *http://www.projectliberty.org/specs*

[SAMLCore2] Cantor, Scott, Kemp, John, Philpott, Rob, Maler, Eve, eds. (15 March 2005). "Assertions and Protocol for the OASIS Security Assertion Markup Language (SAML) V2.0," SAML V2.0, OASIS Standard, Organization for the Advancement of Structured Information Standards *http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf*

[Schema1-2] Thompson, Henry S., Beech, David, Maloney, Murray, Mendelsohn, Noah, eds. (28 October 2004). "XML Schema Part 1: Structures Second Edition," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/xmlschema-1/*

[XML] Bray, Tim, Paoli, Jean, Sperberg-McQueen, C. M., Maler, Eve, Yergeau, Francois, eds. (04 February 2004). "Extensible Markup Language (XML) 1.0 (Third Edition)," Recommendation, World Wide Web Consortium *http://www.w3.org/TR/2004/REC-xml-20040204*