Ubilogin 4.0

Web Services Provider

1. Introduction

1.1. About this document

This document is intended for developers planning, developing or configuring Web Services Providers (WSP) that are relying on Ubilogin Web Services Identity Provider (WSIDP) as their Identity Provider. The communication between WSP, Web Services Client (WSC) and WSIDP is based on Oasis-Open's (http://www.oasis-open.org/) SAML 2.0 over SOAP protocol.

1.2. Ubilogin

The Ubisecure Ubilogin Single Sign-On is a solution that enables single sign-on user authentication using a selection of authentication methods: username and password, One-Time Passwords, smart card (or other client certificate), or GSM short messages (plain text or signed).

The key functionality of Ubilogin is to offer single sign-on to web applications with a selection of authentication methods to best serve the needs of the application or user level in question.

For more detailed introduction into Ubilogin please refer to [SolutionGuide].

1.3. Web services client, Web Services Consumer and Enhanced Client or Proxy

Liberty Alliance uses the term Web Services Consumer (WSC) for a web services client program with wider functionality than a normal web browser. The Security Services committee of Oasis Open uses the term Enhanced Client or Proxy (ECP) in SAML standards for SAML clients with SAML specific features. The term web services client (or simply client) is used in this document to refer to a client program that acts as WSC and/or ECP.

2. Ubilogin for Web Services Providers

Ubilogin WSIDP is based on SOAP 1.1, SAML 2.0 and Liberty ID-WSF 2.0 Authentication Service specifiations. Web services clients uses the Authentication Service to get SAML assertions from Ubilogin Web Services Identity Provider that acts as an Identity Provider (IDP) to a number of Web Services Providers (WSP). Web services client can then access services provided by Web Services Providers by getting an authorization from WSIDP.

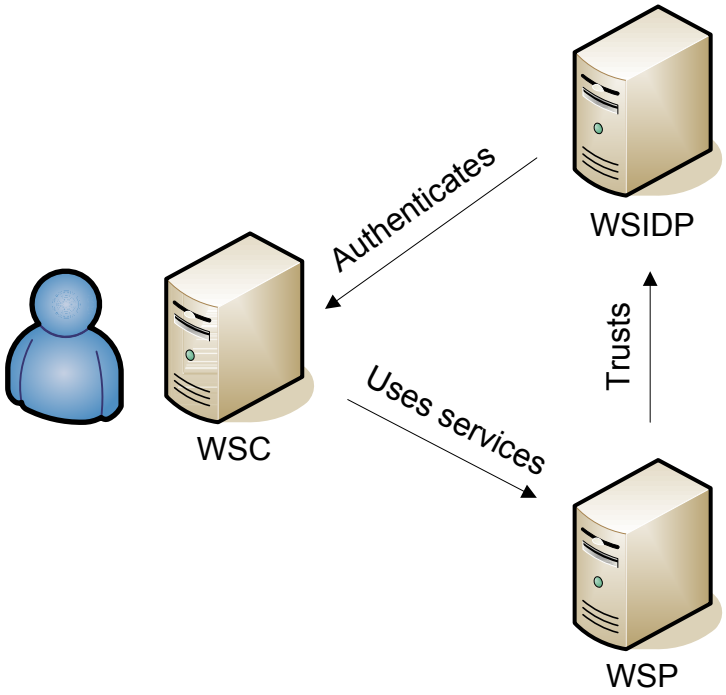

Figure 1. Client authenticates to the WSIDP and WSPs trust the assertions of WSIDP about Client's identity

2.1. The necessary software components

To implement a Web Services Provider the following components are needed:

1. **An XML parser**
2. **A SOAP and PAOS stack** [SOAP] [PAOS]
   SOAP and PAOS are used in communications between WSP and WSC. See chapter 2.4 for more details.
3. **An implementation of SAML SOAP-binding and SAML ECP profile** [SAML-Core] [SAML-Bindings] [SAML-Profiles]
   See chapter 2.4 for more details.
4. **Functionality to generate and read SAML metadata** [SAML-Meta]
   See chapters 2.2 and 2.3 for more details on the usage of SAML metadata on WSIDP and WSP.
5. **An XML Signature implementation** [XMLSig]
   The WSP must be able to buld XML Signatures in the AuthnRequest message and validate the XML Signatures made by the WSIDP in the SAML Response message

The following chapters introduce the Ubilogin specific elements and limitations to these techniques and protocols, but the primary source are the referenced standards and specifications. One should be familiar with these before continuing. See chapter 3 References for the sources of the normative specifications.

2.2. SAML operations supported by WSIDP

Latest WSIDP metadata can be downloaded directly from the server, using the metadata distribution URL of the server (https://example.com/wsidp/saml2/metadata.xml). The AuthnRequests have to be signed always (as described by WantAuthnRequestsSigned

attribute) and the metadata will always have one public RSAkey that WSIDP uses for signing the responses and assertions. Metadata will also list supported single sign-on bindings and locations. Web Services Provider will use only the SOAP-binding for the single sign-on. For a detailed description of the SAML 2.0 metadata please refer to [SAML-Meta].

Listing 1. WSIDP metadata example

```
<?xml version="1.0" encoding="UTF-8"?>

<md:EntityDescriptor xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
entityID='https://example.com/wsidp'
xmlns:md='urn:oasis:names:tc:SAML:2.0:metadata'
ID='_5b8e7a5f7ac51fa77e7a170a6f48dcce6004df7d'>

  <ds:Signature xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>

    …signature removed…

  </ds:Signature>

  <md:IDPSSODescriptor
protocolSupportEnumeration='urn:oasis:names:tc:SAML:2.0:protocol'
WantAuthnRequestsSigned='true'>

    <md:KeyDescriptor use='signing'>

      <ds:KeyInfo xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>

        <ds:KeyValue>

          <ds:RSAKeyValue>

            <ds:Modulus>

wvjRdAKcZDxHsnB6nKyRedAgBCk5nACsDUEw3PSOtjDh7h2GriUx4+vACoE6DrJHuv9swmKsFDPz

Gjg7h83uotVobMKn9X4sVy6FUInoy2Wx8/85vWdE83vymiIBlshEyd96+/ir3jR9GScSfBcL0v0j

DXUPaMzWqcaXp6mX0bM=

</ds:Modulus>

            <ds:Exponent>AQAB</ds:Exponent>

          </ds:RSAKeyValue>

        </ds:KeyValue>

      </ds:KeyInfo>

    </md:KeyDescriptor>

    <md:SingleSignOnService
Location='https://www.tunnitus.fi/wsidp/saml2/SingleSignOnService'
Binding='urn:oasis:names:tc:SAML:2.0:bindings:SOAP'></md:SingleSignOnService>

  </md:IDPSSODescriptor>

</md:EntityDescriptor>
```

2.3. Registering a Web Services Provider to Ubilogin Management

A Web Services Provider has to be registered to Ubilogin before WSIDP can act as an Identity Provider to WSP. Registration happens by importing WSP's metadata [SAML-Meta] to Ubilogin Directory through Ubilogin Management.

Web Services Provider requirements and metadata

The format of metadata describing the Service Providers configuration and supported features is described in [SAML-Meta]. The AuthnRequests the SP sends should always be signed, so the AuthnRequestsSigned attribute in SPSSODescriptor element should be set to true. The SP must provide one RSA public key that it will use to sign requests. For Web Services Providers there must be an assertion consumer service endpoint using SOAP binding.

Listing 2. WSP metadata example

```
<EntityDescriptor entityID='urn:uuid:2ace2b26-da5c-4161-93b9-629e557f403f'
ID='_12345'>
  <SPSSODescriptor
protocolSupportEnumeration='urn:oasis:names:tc:SAML:2.0:protocol'
AuthnRequestsSigned='true'>
    <KeyDescriptor use='signing'>
      <ds:KeyInfo>
        <ds:KeyValue>
          <ds:RSAKeyValue>
            <ds:Modulus>
ssNEdM4j+narW2xRGyTWc4aodD+hwQhLtj4+IZMlinprbdLYdvRZIua2af2Qr82t/b93Oge69VMo
qw9gJj3p1FM+ahMKhGaV6da690CPE8uckA3yuxabv1VZAd8n7SM4NnNQIiwLVXFw57XWnwUpzHf5
lBtnJYjiJIskgottp80=
</ds:Modulus>
            <ds:Exponent>AQAB</ds:Exponent>
          </ds:RSAKeyValue>
        </ds:KeyValue>
      </ds:KeyInfo>
    </KeyDescriptor>
    <AssertionConsumerService Location='https://example.com/sp1/sp.jsp'
index='0' isDefault='true'
Binding='urn:oasis:names:tc:SAML:2.0:bindings:SOAP' />
  </SPSSODescriptor>
</EntityDescriptor>
```

Importing the metadata through Ubilogin Management

Import the metadata from Ubilogin Management as described in [Management].

2.4. Requesting authentication from a client

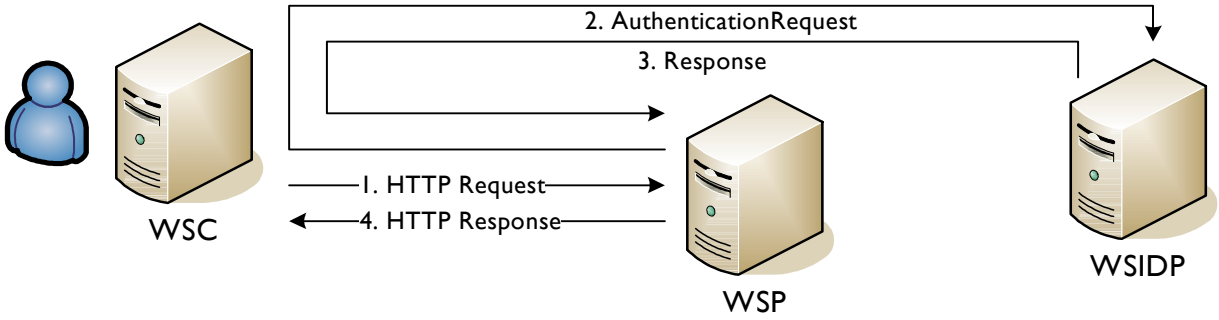

Figure 2. Client tries to access services from WSP and is redirected to get an authentication from WSIDP

1. Web Services Client tries to use a service from Web Services Provider
2. If the WSP does not recognize the WSC it sends a SAML AuthnRequest message using PAOS protocol. The WSC adds to the request the assertions it has received from the WSIDP earlier or if the client has not authenticated to the WSIDP yet it can authenticate before continuing to send the authentication request to the WSIDP (see [Ubilogin-WSC] for more information).
3. WSIDP creates a SAML response or a SOAP error message and returns that to client that forwards it to WSP.
4. WSP checks the response returned by WSIDP and responds to client's request.

Listing 3. An example SAML AuthnRequest sent over PAOS protocol

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/'>

  <SOAP-ENV:Header>

    <paos:Request xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
soapenv:actor='http://schemas.xmlsoap.org/soap/actor/next'
service='urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
responseConsumerURL='https://example.com/sp1/sp'
xmlns:paos='urn:liberty:paos:2003-08'
xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
soapenv:mustUnderstand='1' />

    <ecp:Request xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
soapenv:actor='http://schemas.xmlsoap.org/soap/actor/next'
xmlns:ecp='urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
soapenv:mustUnderstand='1'>

      <saml:Issuer Format='urn:oasis:names:tc:SAML:2.0:nameid-
format:entity'>urn:uuid:2ace2b26-da5c-4161-93b9-629e557f403f</saml:Issuer>

      <samlp:IDPList xmlns:samlp='urn:oasis:names:tc:SAML:2.0:protocol'>

        <samlp:IDPEntry ProviderID='https://example.com/wsidp'
Loc='https://example.com/wsidp/saml2/SingleSignOnService' />

      </samlp:IDPList>

    </ecp:Request>
```

```
    <ecp:RelayState xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
soapenv:actor='http://schemas.xmlsoap.org/soap/actor/next'
xmlns:ecp='urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
soapenv:mustUnderstand='1'>8e907ce9e519d204</ecp:RelayState>

  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <samlp:AuthnRequest xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
ProtocolBinding='urn:oasis:names:tc:SAML:2.0:bindings:SOAP'
ID='_2e6a82962ea4a0a0e2d8f495cf78fcee608202a4'
AssertionConsumerServiceURL='https://example.com/sp1/sp' Version='2.0'
IssueInstant='2005-11-07T12:32:17.396+02:00'
Destination='https://example.com/wsidp/saml2/SingleSignOnService'
xmlns:samlp='urn:oasis:names:tc:SAML:2.0:protocol'>

      <saml:Issuer Format='urn:oasis:names:tc:SAML:2.0:nameid-
format:entity'>urn:uuid:2ace2b26-da5c-4161-93b9-629e557f403f</saml:Issuer>

      <ds:Signature xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>

        …signature removed…

      </ds:Signature>

      <samlp:Scoping>

        <samlp:IDPList>

          <samlp:IDPEntry ProviderID='https://example.com/wsidp'
Loc='https://example.com/wsidp/saml2/SingleSignOnService' />

        </samlp:IDPList>

        <samlp:RequesterID>urn:uuid:2ace2b26-da5c-4161-93b9-
629e557f403f</samlp:RequesterID>

      </samlp:Scoping>

    </samlp:AuthnRequest>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

The value of the AssertionConsumerServiceURL attribute in the AuthnRequest element and the value of the responseConsumerURL attribute in the PAOS header must both be the same as the value of the AssertionConsumerURL in the metadata for that Service Provider. The value of the Destination attribute in AuthnRequest element must be the URL specified in SAML metadata for the WSIDP for the used binding. The value of the issuer must be the same as the value of the entityID attribute in the EntityDescriptor element in the metadata for the Service Provider.

Listing 4. An example SAML Response coming from IDP via WSC

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV='http://schemas.xmlsoap.org/soap/envelope/'>

  <SOAP-ENV:Header>

    <ecp:RelayState xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
soapenv:actor='http://schemas.xmlsoap.org/soap/actor/next'
xmlns:ecp='urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp'
xmlns:soapenv='http://schemas.xmlsoap.org/soap/envelope/'
soapenv:mustUnderstand='1'>8e907ce9e519d204</ecp:RelayState>
```

```
  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <samlp:Response xmlns:saml='urn:oasis:names:tc:SAML:2.0:assertion'
ID='_c5729432f0b5dc55ef45a026c7be2eb5a510858a'
Destination='https://example.com/sp1/sp' IssueInstant='2005-11-
09T11:44:41.585Z' Version='2.0'
xmlns:samlp='urn:oasis:names:tc:SAML:2.0:protocol'
InResponseTo='_03888238fd636b6a384c673068b17479a88a8d4c'>

      <saml:Issuer Format='urn:oasis:names:tc:SAML:2.0:nameid-
format:entity'>https://example.com/wsidp</saml:Issuer>

      <ds:Signature xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>

        …signature removed…

      </ds:Signature>

      <samlp:Status>

        <samlp:StatusCode Value='urn:oasis:names:tc:SAML:2.0:status:Success'
/>

      </samlp:Status>

      <saml:Assertion ID='_0e0674864b46f6a68b4fc5762de74ec855d50a4d'
IssueInstant='2005-11-09T11:44:41.585Z' Version='2.0'>

        <saml:Issuer Format='urn:oasis:names:tc:SAML:2.0:nameid-
format:entity'>https://example.com/wsidp</saml:Issuer>

        <ds:Signature xmlns:ds='http://www.w3.org/2000/09/xmldsig#'>

          …signature removed…

        </ds:Signature>

        <saml:Subject>

          <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-
format:X509SubjectName" NameQualifier="ldap://localhost/dc=identityprovider,
dc=com">uid=user,ou=Persons,dc=identityprovider,dc=com</saml:NameID>

          <saml:SubjectConfirmation
Method='urn:oasis:names:tc:SAML:2.0:cm:bearer'>

            <saml:SubjectConfirmationData NotOnOrAfter='2005-11-
09T11:54:41.585Z' Recipient='https://example.com/sp1/sp'
InResponseTo='_03888238fd636b6a384c673068b17479a88a8d4c' />

          </saml:SubjectConfirmation>

        </saml:Subject>

        <saml:Conditions NotBefore='2005-11-09T11:44:41.585Z'
NotOnOrAfter='2005-11-09T11:54:41.585Z'>

          <saml:AudienceRestriction>

            <saml:Audience>urn:uuid:2ace2b26-da5c-4161-93b9-
629e557f403f</saml:Audience>

          </saml:AudienceRestriction>

        </saml:Conditions>

        <saml:AuthnStatement AuthnInstant='2005-11-09T11:44:41.210Z'
SessionNotOnOrAfter='2005-11-09T12:44:41.585Z'>

          <saml:SubjectLocality />
```

```
          <saml:AuthnContext>
            <saml:AuthnContextDeclRef>
https://example.com/wsidp/saml2/names/ac/password.1</saml:AuthnContextDeclRef
>
          </saml:AuthnContext>
        </saml:AuthnStatement>
        <saml:AttributeStatement>
          <saml:Attribute Name="role">
            <saml:AttributeValue>manager</saml:AttributeValue>
          </saml:Attribute>
        </saml:AttributeStatement>
      </saml:Assertion>
    </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The value of the issuer element of the SAML Response is the same as the value of entityID attribute in the EntityDescriptor element in the metadata for the WSIDP.

Assertions created by the WSIDP

The assertions that WSIDP creates are standard SAML2 ECP profile assertions as described in [SAML-Core], [SAML-Bindings] and [SAML-Profiles]. All AuthnStatements created by WSIDP contain the authentication context declaration reference (AuthnContextDeclRef) of the authentication method that used to authenticate the client at WSIDP.

3. References

[Management] Ubilogin Server – Management Guide

[PAOS] Liberty Reverse HTTP Binding for SOAP Specification
http://www.projectliberty.org/liberty/content/download/2008/13941/file/liberty-paos-v1.0.pdf

[SAML-Profiles] Profiles for the OASIS Security Assertion Markup Language (SAML)
http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf

[SAML-Bindings] Bindings for the OASIS Security Assertion Markup Language (SAML)
http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf

[SAML-Core] Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML)
http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf

[SAML-Meta] Metadata for the OASIS Security Assertion Markup Language (SAML)
http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf

[SOAP] Simple Object Access Protocol (SOAP) 1.1
http://www.w3.org/TR/2000/NOTE-SOAP-20000508/

[SolutionGuide] Ubilogin Single Sign-On – Tuotekuvaus

[Ubilogin-WSC] Ubilogin 4.0 – Web Services Client

[XMLSig] XML-Signature Syntax and Processing
http://www.w3.org/TR/xmldsig-core/

4. Contact Information

Ubisecure Solutions, Inc.

www.ubisecure.com
info@ubisecure.com
support@ubisecure.com

<firstname.lastname>@ ubisecure.com

Tekniikantie 14
FIN-02150 Espoo, FINLAND

tel. +358-9-2517 7250
fax +358-9-2517 7070

Registered in Espoo, Finland
reg. nr. FI17487214

About Ubisecure

Ubisecure Solutions, Inc. is a leading partner in providing advanced authentication and authorization solutions for Internet, Intranet and Extranet services. Ubisecure provides application developers, integrators, solution providers and end-user organizations with IT-security software solutions that maximize the competitive advantage of its customers. The Ubisecure product line consists of Ubilogin solutions for authentication and Web Single Sign On access to Internet and Intranet/Extranet services, Ubipass VPN-authentication and Ubisignature electronic signatures. Ubisecure provided authentication utilizes ordinary GSM handsets, challenge-response SMS-messages, one-time passwords in Java-phones, smart cards, Windows Integrated Authentication as well as various third party vendor services and products. Ubisecure has offices in Finland and Sweden.

For more information, visit Ubisecure 's web site at www.ubisecure.com

