Ubilogin 4.0

Web Services Clients (KATVE)

1. Introduction

1.1. About this document

This document is intended for developers planning, developing or configuring a web services client that is going to access services that are using www.tunnistus.fi as their Identity Provider. The www.tunnistus.fi Identity Provider is based on Ubilogin Web Services Identity Provider (WSIDP). The web services client uses Liberty Alliance ID-WSF 2.0 Authentication Service for authenticating to WSIDP and Oasis Open SAML 2.0 to communicate authentication, authorization and attribute information from WSIDP to Service Providers.

Note: some element contents of the XML message examples in this document are stripped for brevity, see [Message-exchange] for complete XML messages.

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 ECP.

2. Ubilogin for Web services clients

Ubilogin WSIDP is based on SOAP 1.1, SAML 2.0 and Liberty ID-WSF 2.0 Authentication Service specifications. Web services clients 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 Service Providers by getting an authorization from WSIDP.

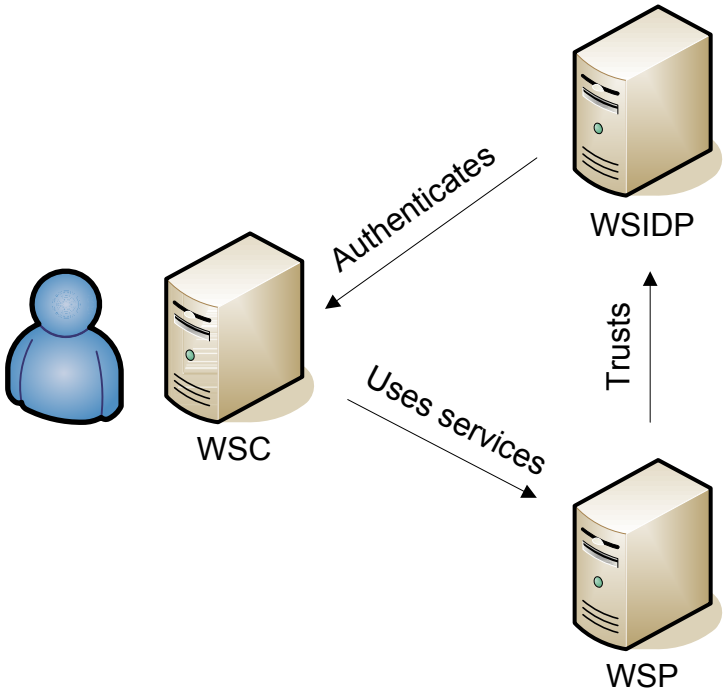

Figure 1. The WSIDP authenticates the Web Services Client and Web Services Providers trust the assertions of WSIDP about WSC's identity

2.1. The necessary software components

To implement a Web Services Client the following basic components are needed:

1. **An XML parser**
2. **A SOAP and PAOS stack** [SOAP] [PAOS]
   SOAP and PAOS bindings are used in communications between WSC, WSP and WSIDP. See chapters 2.3 and 2.5 for details on the usage of SOAP and PAOS.
3. **An ID-WSF authentication stack with SASL support** [ID-WSF] [ID-WSF-SOAP] [RFC2222]
   See chapters 2.3 and 2.4 for details on the usage of ID-WSF and SASL with WSIDP
4. **Support for Web Services Security SAML Token Profile** [WSS-SAML]
   WSS SAML Token Profile is used in transmitting the SAML assertions. See chapter 2.5.
5. **The functionality to parse the validity period from a SAML-assertion** [SAML-Core]
   Note, that very minimal support for SAML is needed. WSC only needs to be able to parse the validity period from the assertions it receives, if it wants to keep these assertions cached for further use. See chapter 2.3 for details.

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.

2.2. WSIDP endpoint URLs

The WSC needs to obtain the Ubilogin WSIDP's endpoint URL for the ID-WSF Authentication Service. The endpoint path is **/idwsf/sasl** and thus in www.tunnistus.fi the endpoint URL is **http://www.tunnistus.fi/wsidp/idwsf/sasl**.

The endpoint URL for the SAML Single Sign-On Service is conveyed in the *samlp:IDPList* SOAP Header of the AuthnRequest message issued by the WSP (see Listing 8).

2.3. Getting SAML assertions from WSIDP

Web services client needs to authenticate itself to WSIDP using ID-WSF Authentication Service in order to get assertions that are later sent back to WSIDP when requesting access to a specific Service Provider. This authentication can happen at the start up of the web services client or it could be done right after Service Provider sends a SAML authentication request and before the request is forwarded to WSIDP. (See chapter 2.4.)

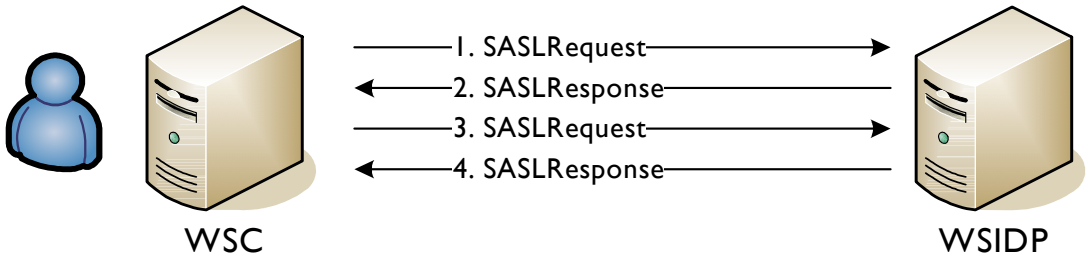

Figure 2. ID-WSF Authentication Service

1. Web services client suggests an authentication mechanism (for example PLAIN or KATSO) and optionally sends credentials.
2. WSIDP chooses the authentication mechanism and optionally sends a challenge to Web services client.
3. Web services client provides a response to the challenge.
4. If the authentication succeeds, WSIDP returns a SAML assertion to Web services client.

The result of this authentication is a SAML assertion from WSIDP to the Web services client. Web services client does not need to understand these assertions beyond the point of checking their time window of validity. Web services client should save the assertions for later when requesting access to a specific Web Service Provider (see chapter 2.4).

The first request that the client sends to the WSIDP can already contain the authentication information as described in [ID-WSF]. If the authentication can be carried out with the information offered in the first request and there is no need for a challenge-reponse messaging, then the steps 2 and 3 from Figure 2 can be omitted.

SASL Authentication

The SASL protocol [RFC4422] uses a number of different authentication mechanisms. PLAIN authentication mechanism for SASL is described in [RFC4616]. In PLAIN mechanism, the value of the Data element has three fields: the authorization identity (identity to login as), followed by an US-ASCII NUL character, followed by the authentication identity (identity whose password will be used), followed by an US-ASCII NUL character, followed by the clear-text password.

Listing 1. The contents of SASL authentication data using the PLAIN mechanism

```
Authorization identity<nul>authentication identity<nul>password
```

To distinguish between different authentication methods using the same SASL mechanism in Ubilogin WSIDP, the authentication identity must be prefixed with the authentication method's name and a colon.

Katso authentication methods

There are two Katso authentication methods implemented in www.tunnistus.fi: Katso Password and Katso One Time Password (Katso OTP). Katso Password uses the SASL PLAIN mechanism. Katso OTP uses a SASL mechanism called KATSO [KATSO]. These methods are described in more detail below.

Katso Password

Katso Password is a password authentication method that uses the Katso user repository as its authentication source. When authenticating with the Katso Password method, the client sends the authentication information using PLAIN mechanism. The value of the authentication identity must be prefixed with "*katsole:*" (see Listing 2). The Authorization identity field is not used.

Listing 2. The contents of SASL authentication data using Katso Password

```
<nul>katsole:username<nul>password
```

Listing 3. An example SASL request using Katso Password authentication method. The authentication data is included in the initial request. Note: the Data element contents are Base64 encoded.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:Action>urn:liberty:sa:2006-08:SASLRequest</wsa:Action>
    <sbf:Framework version="2.0"/>
    <wsa:MessageID>urn:uuid:f23f5b87-3656-4ecf-89b8-
8d41111fb4b3</wsa:MessageID>
    <wsa:ReplyTo>
      <wsa:Address>http://www.w3.org/2005/08/addressing/anonymous
</wsa:Address>
    </wsa:ReplyTo>
    <wsse:Security>
      <wsu:Timestamp>
        <wsu:Created>2007-06-29T04:55:57.567Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sa:SASLRequest mechanism="PLAIN" xmlns:sa="urn:liberty:sa:2006-08">
      <sa:Data>AGthdHNvbGU6dXNlcgBwYXNz<sa:Data>
    </sa:SASLRequest>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Listing 4. An example SASL response to a successful Katso Password authentication. Some element contents stripped for brevity, see [Message-exchange] for complete XML messages.

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:Action>urn:liberty:sa:2006-08:SASLResponse</wsa:Action>
    <sbf:Framework version="2.0"/>
    <wsa:MessageID>urn:uuid:2b132c12-6d88-4296-bd5e-
5a23fd52ad5a</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:626072e3-37db-40c5-ad20-
c5d28e0471c3</wsa:RelatesTo>
    <wsse:Security>
      <wsu:Timestamp>
        <wsu:Created>2007-06-29T05:10:00.410Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sa:SASLResponse serverMechanism="PLAIN" xmlns:sa="urn:liberty:sa:2006-
08">
      <lu:Status code="OK" xmlns:lu="urn:liberty:util:2006-08"/>
      <wsa:EndpointReference
xmlns:wsa="http://www.w3.org/2005/08/addressing">
        <wsa:Address>https://www.tunnistus.fi/wsidp/saml2/SingleSignOnService
</wsa:Address>
        <wsa:Metadata>
          <disco:ServiceType xmlns:disco="urn:liberty:disco:2006-08">
urn:oasis:names:tc:SAML:2.0:protocol</disco:ServiceType>
          <disco:ProviderID xmlns:disco="urn:liberty:disco:2006-08">
https://www.tunnistus.fi/wsidp</disco:ProviderID>
          <sbf:Framework version="2.0" xmlns:sbf="urn:liberty:sb"/>
          <disco:SecurityContext xmlns:disco="urn:liberty:disco:2006-08">
            <disco:SecurityMechID>urn:liberty:security:2006-
08:TLS:Bearer</disco:SecurityMechID>
            <sec:Token usage="urn:liberty:security:tokenusage:2006-
08:SecurityToken" xmlns:sec="urn:liberty:security:2006-08">
              <saml:Assertion
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
ID="_dbfb4581b849385a140253f867b7b4a1ad0768c5" IssueInstant="2007-06-
29T05:10:00.394Z" Version="2.0">
                <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://www.tunnistus.fi/wsidp</saml:Issuer>
                <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">
                  …signature removed…
```



```
                </ds:Signature>
                <saml:Subject>
                  <saml:NameID> dGhpcy1pcy1hLWZpY3Rpb25hbC1uYW1laWQtZGF0YQo=
</saml:NameID>
                  <saml:SubjectConfirmation
Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
                    <saml:SubjectConfirmationData NotOnOrAfter="2007-06-
29T05:20:00.394Z" Recipient="https://www.tunnistus.fi/wsidp"/>
                  </saml:SubjectConfirmation>
                </saml:Subject>
                <saml:Conditions NotBefore="2007-06-29T05:10:00.394Z"
NotOnOrAfter="2007-06-29T05:20:00.394Z">
                  <saml:AudienceRestriction>
                    <saml:Audience>https://www.tunnistus.fi/wsidp
</saml:Audience>
                  </saml:AudienceRestriction>
                </saml:Conditions>
                <saml:AuthnStatement AuthnInstant="2007-06-29T05:10:00.332Z">
                  <saml:SubjectLocality/>
                  <saml:AuthnContext>
                    <saml:AuthnContextClassRef>
urn:oasis:names:tc:SAML:2.0:ac:classes:unspecified
</saml:AuthnContextClassRef>
                  </saml:AuthnContext>
                </saml:AuthnStatement>
              </saml:Assertion>
            </sec:Token>
          </disco:SecurityContext>
        </wsa:Metadata>
      </wsa:EndpointReference>
    </sa:SASLResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

The SAML Assertion is returned in a *sec:Token* element as in Listing 4. The validity period can be checked from the *saml:Conditions* element. The assertion in Listing 4 is valid for 10 minutes starting from 2007-06-29 05:10:00.394. Note that all the timestamps are set in the UTC time zone.

Katso OTP

Katso OTP is a strong authentication method which is based on a user id, password and one time password. Katso user repository is used for Katso OTP authentication. Katso OTP uses KATSO SASL mechanism described in [KATSO]. In KATSO mechanism, the WSIDP can reply with a SASL Status code “Continue” containing a server challenge as the Data element’s

contents. The challenge contains the Katso OTP serial number. In that case the client should fix the authentication data and send another SASLRequest.

Listing 5. The contents of SASL authentication data using Katso OTP

```
username<nul>password<nul>one-time-password
```

Refer to [KATSO] for more detailed discussion.

KTYVI

KTYVI is also a password authentication method which is in use in www.tunnistus.fi. It uses the KTYVI LDAP directory as its authentication source. When authenticating with the KTYVI method, the client sends the authentication information using PLAIN mechanism and the authentication identity value prefixed with *"katve:" (see Listing 6)*. Authorization identity is not used.

Listing 6. The contents of SASL authentication data using KTYVI

```
<nul>katve:username<nul>password
```

2.4. SASL Status Comments

In some situations the SASLResponse message Status element may contain an extra comment attribute, which describes various special situations concerning the Katso identifiers. The comment field can occur only while using the Katso Password or Katso OTP authentication method. The comment attribute can contain one of the following strings: *"Initialized"*, *"Founded"*, *"Locked"*, *"Disabled"*, *"OutOfOTP"* or *"AccountIsExpiring"*.

Listing 7. An example of a SASLResponse with a Katso-specific status comment attribute "Initialized"

```
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
    <wsa:Action>urn:liberty:sa:2006-08:SASLResponse</wsa:Action>
    <sbf:Framework version="2.0"/>
    <wsa:MessageID>urn:uuid:2b132c12-6d88-4296-bd5e-
5a23fd52ad5a</wsa:MessageID>
    <wsa:RelatesTo>urn:uuid:626072e3-37db-40c5-ad20-
c5d28e0471c3</wsa:RelatesTo>
    <wsse:Security>
      <wsu:Timestamp>
        <wsu:Created>2007-06-29T05:10:00.410Z</wsu:Created>
      </wsu:Timestamp>
    </wsse:Security>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
    <sa:SASLResponse serverMechanism="KATSO" xmlns:sa="urn:liberty:sa:2006-
08">
      <sa:Status code="Abort" comment="Initialized"/>
```



```
    </sa:SASLResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Five different comment attributes may occur only along an *"Abort"* status code:

- *"Initialized"*: The Katso identifier is initialized but not active. The user can activate his identifier in the Katso administration.
- *"Founded"*: The Katso identifier is founded but not active. The user will get a confirmation in email when the identifier is activated.
- *"Locked"*: The Katso identifier is locked. The user can unlock the identifier in the Katso administration. However, if the identifier is a subidentifier, it can be unlocked only by the organization's master user.
- *"Disabled"*: The Katso identifier is disabled or cancelled. The user should contact Katso support at katso@vero.fi or by phone at 010 320 560.
- *"OutOfOTP"*: The Katso identifier is out is one-time passwords. The user can not authenticate using the Katso OTP authentication method without acquiring a new one-time password list.

The *"AccountIsExpiring"* comment attribute may occur only along a *"Continue"* or *"OK"* status code. It means that there are not many Katso one-time passwords left. The user should be reminded, that he should print a new one-time password list.

2.5. Accessing services from Service Provider

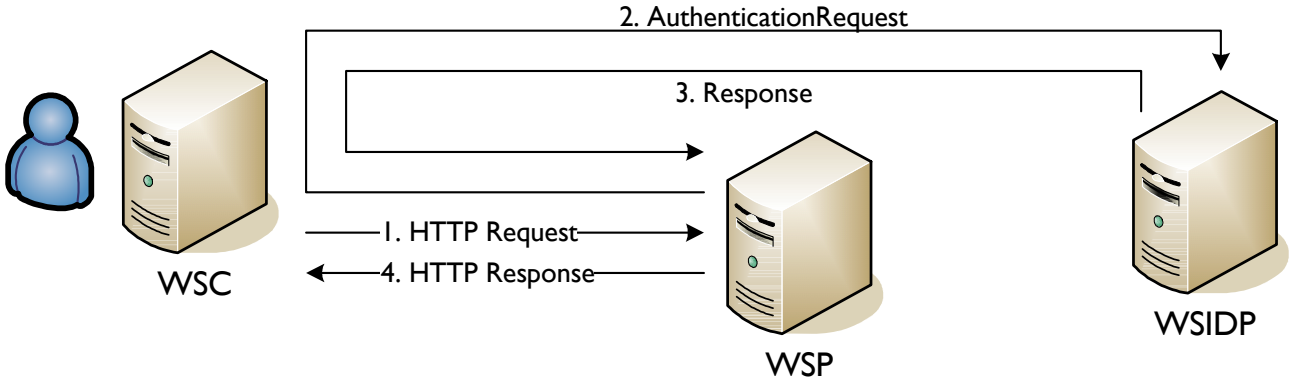

Figure 3. SAML authentication to Service Provider

1. Web Services Client tries to use a service from Web Service 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 chapter 2.3).
3. WSIDP creates a SAML response or a SOAP error message and returns that to the Client which forwards it to the WSP.
4. WSP checks the response returned by WSIDP and responds to Client's original request (1).

Listing 8. An example of the SAML AuthnRequest message sent over PAOS protocol from WSP to the client

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

  <SOAP-ENV:Header>

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

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

      <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://serviceprovider.com/sp1</saml:Issuer>

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

        <samlp:IDPEntry Loc="https://www.tunnistus.fi/wsidp/saml2/
SingleSignOnService" ProviderID="https://www.tunnistus.fi/wsidp"/>
```

```
      </samlp:IDPList>

    </ecp:Request>

    <ecp:RelayState xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1">6760f2bbcefefc72</ecp:RelayState>

  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <samlp:AuthnRequest xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
AssertionConsumerServiceURL="https://serviceprovider.com/sp1/AssertionConsume
r" Destination="https://www.tunnistus.fi/wsidp/saml2/SingleSignOnService"
ID="_7467a3ef77968403d9196c29207c360e6c24649b" IssueInstant="2006-03-
10T07:58:18.498Z" ProtocolBinding="urn:oasis:names:tc:SAML:2.0:bindings:SOAP"
Version="2.0">

      <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://serviceprovider.com/sp1</saml:Issuer>

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

        …signature removed…

      </ds:Signature>

      <samlp:Scoping>

        <samlp:IDPList>

          <samlp:IDPEntry Loc="https://www.tunnistus.fi/wsidp/saml2/
SingleSignOnService" ProviderID="https://www.tunnistus.fi/wsidp"/>

        </samlp:IDPList>

        <samlp:RequesterID>https://serviceprovider.com/
sp1</samlp:RequesterID>

      </samlp:Scoping>

    </samlp:AuthnRequest>

  </SOAP-ENV:Body>

</SOAP-ENV:Envelope>
```

Before sending the request from WSP to WSIDP the client must add the assertions it has earlier received from WSIDP to the SOAP headers using *<wsse:Security>* element as described in [WSS-SAML] and remove the *<paos:Request>, <ecp:RelayState>,* and *<ecp:Request>* headers as stated in [SAML-Profiles].

If the Assertion attached in the AuthnRequest has expired, WSIDP will response with SOAP Fault with the *wsse:InvalidSecurityToken* faultcode.

Listing 9. An example SAML response from WSIDP to the client. Before forwarding the message to the WSP, the Client should remove the ecp:Response header element and add the ecp:RelayState header.

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

  <SOAP-ENV:Header>
```

```
    <ecp:Response xmlns:ecp="urn:oasis:names:tc:SAML:2.0:profiles:SSO:ecp"
xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
AssertionConsumerServiceURL="https://serviceprovider.com/sp1/AssertionConsume
r" soapenv:actor="http://schemas.xmlsoap.org/soap/actor/next"
soapenv:mustUnderstand="1"/>

  </SOAP-ENV:Header>

  <SOAP-ENV:Body>

    <samlp:Response xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
xmlns:samlp="urn:oasis:names:tc:SAML:2.0:protocol"
Destination="https://serviceprovider.com/sp1/AssertionConsumer"
ID="_207b4c2c29ddb1a3389a391c4e9ca70b801ee660"
InResponseTo="_7467a3ef77968403d9196c29207c360e6c24649b" IssueInstant="2006-
03-10T07:58:32.733Z" Version="2.0">

      <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://www.tunnistus.fi/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="_63a7280e97cba802a613d4be46500299ab65a2cb"
IssueInstant="2006-03-10T07:58:32.733Z" Version="2.0">

        <saml:Issuer Format="urn:oasis:names:tc:SAML:2.0:nameid-
format:entity">https://www.tunnistus.fi/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:unspecified">user</saml:NameID>

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

            <saml:SubjectConfirmationData
InResponseTo="_7467a3ef77968403d9196c29207c360e6c24649b" NotOnOrAfter="2006-
03-10T08:08:32.733Z"
Recipient="https://serviceprovider.com/sp1/AssertionConsumer"/>

          </saml:SubjectConfirmation>

        </saml:Subject>

        <saml:Conditions NotBefore="2006-03-10T07:58:32.733Z"
NotOnOrAfter="2006-03-10T08:08:32.733Z">

          <saml:AudienceRestriction>

            <saml:Audience>https://serviceprovider.com/sp1</saml:Audience>

          </saml:AudienceRestriction>
```

```
        </saml:Conditions>
        <saml:AuthnStatement AuthnInstant="2006-03-10T07:58:29.702Z"
SessionNotOnOrAfter="2006-03-10T08:58:32.812Z">
          <saml:SubjectLocality/>
          <saml:AuthnContext>
            <saml:AuthnContextDeclRef>https://www.tunnistus.fi/wsidp/saml2/
names/ac/otp.katso.1</saml:AuthnContextDeclRef>
          </saml:AuthnContext>
        </saml:AuthnStatement>
        <saml:AttributeStatement>
          <saml:Attribute Name="tfi.version">
            <saml:AttributeValue>katso-1.0</saml:AttributeValue>
          </saml:Attribute>
          <saml:Attribute Name="tfi.kid">
            <saml:AttributeValue>user</saml:AttributeValue>
          </saml:Attribute>
          <saml:Attribute Name="tfi.personname ">
            <saml:AttributeValue>John User</saml:AttributeValue>
          </saml:Attribute>
        </saml:AttributeStatement>
      </saml:Assertion>
    </samlp:Response>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Before sending the response from WSIDP to WSP, the client must the remove the *<ecp:Response >* header and add the *<ecp:RelayState>* and possibly a *<paos:Response>* header as stated in [SAML-Profiles].

For more complete example messages refer to [MessageExchange].

3. References

[ID-WSF] Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification

http://www.projectliberty.org/liberty/content/download/871/6189/file/liberty-idwsf-authn-svc-v2.0.pdf

[ID-WSF-SOAP] Liberty ID-WSF SOAP Binding Specification

http://www.projectliberty.org/liberty/content/download/897/6267/file/liberty-idwsf-soap-binding-v2.0.pdf

[KATSO] The Katso SASL Mechanism

[Message-exchange] Ubilogin WSIDP Message Exchange Example

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

[RFC4616] The PLAIN Simple Authentication and Security Layer (SASL) Mechanism
http://www.ietf.org/rfc/rfc4616.txt

[RFC4422] Simple Authentication and Security Layer (SASL)
http://www.ietf.org/rfc/rfc4422.txt

[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-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

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

[SolutionGuide] Ubilogin Single Sign-On – Tuotekuvaus

[WSS-SAML] Web Services Security: SAML Token Profile
http://docs.oasis-open.org/wss/oasis-wss-saml-token-profile-1.0.pdf

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

