TAS^3 Workshop
Architecture Using ZXID and PERMIS

Sampo Kellomäki (sampo@symlabs.com), Symlabs

25-26-27.8.2009 Lisboa

Notes from Buda

IdP integration items * AMQP or SAWS * OCT support, for generating tokens

PEP integration items * Method profiles (stored on PEP machine as configuration, profiles

```
written by app developer) to describe attributes to fee
```

ID Mapper integration * Trust and Privacy Negotiator mechanics

TalkTo

- Jutta:
  - how to integrate the workflow (to mod_auth_saml?)
  - discovery
  - PIP
  - Stack
- Brecht: profiles, interop

Overall Outline for 3 Days

https://portal.tas3.eu/trac/wiki/Meeting/2009-08-25

Venue: R. Padre Damian 6B, Lisboa (behind Centro Cultural Belém)
Sampo: +351-918.731.007

Tue Setup, infra, and demo

Wed ZXID

Thu PERMIS

- Travel arrangements? Usability of Thu and Fri?
- 9 am to 19 pm
- Coffee and tea provided
- Lunch at the near by restaurant
- Dinner plan?

Attendance

- Jeroen
- Marc S.
- Jens
- Tom
- Brian
- Marc Van Collie
- David
- Stijn
- George
- Sampo

Setup, Infra, and Demo Outline (Tue)

1. WiFi connectivity, firewall (full out, nothing in), etc.
   - WPA: ssid="BNGWIZI_Adsl" psk="72JTHPK5ACNA9"
   - Use DHCP (netmask 24 bits, gateway: `192.168.1.1`)
     - After DHCP gives you address, use that as fixed address
   - DNS: OpenDNS `208.67.222.222 208.67.220.220` for external
     - Use `/etc/hosts` for peers after fixed IPs
2. Concrete architecture we are trying to setup
   - Feedback and planning on objectives of each participant
3. Demo of what is there already: SSO and Az
4. CA and setup certs for everybody, Connectivity Test
5. Compile / Package Install for ZXID and PERMIS
6. Output documents from this event?

ZXID Outline (Wed) (1/2)

1 Create your own SP

1. Dummy using ZXID standalone code
2. Hookup to CoT, metadata
3. See it work
4. Integrate to your own code
5. See it work

2 Triggering Az from SSO

3 Using SSO attributes

4 Creating your own WSC

1. Demo of actual web service call, with traces
2. Integrating ZXID code to call existing service

ZXID Outline (Wed) (2/2)

5 Providing your own WSP

1. Integrating ZXID code
2. Service Registration Step
3. Association Step
4. Making web service call: your WSC to your WSP

6 Interop

1. Discovering other people's WSPs
2. Your WSC calling other people's WSP
3. Your WSP being called by other people's WSC

7 mod_auth_saml tutorial

PERMIS Outline (Thu)

Supplied separately by Kent.

Homework Prior to Event (1/2)

https://portal.tas3.eu/trac/wiki/Meeting/2009-08-25/ZXID

The workshop is intended to be on developer or poweruser deployer level. Therefore

- You MUST have C development environment (gcc, ld, make, sed, perl, tar, gunzip) installed. Be sure to install headers as well. You will also need OpenSSL and libCurl development packages. On Windows, install Cygwin with the above (and below) components.
- If you plan to use perl, php, Java, or other scripting solution, be sure to have full development environment for whatever you do. If you do Java, have your Tomcat figured out and working.
- Have a web server (Apache 2.2 recommended) installed and functioning on your laptop.

Homework Prior to Event (2/2)

- Practise creating X509v3 certificates with your tools.
- Have wireshark or similar installed and know how to use it. Also browser plugins like "Tamper Data" for analyzing http traffic may come handy.
- Compile zxid downloaded from `zxid.org`
- Compile PERMIS

CA and Certs

- Jeroen's CA
  - Jeroen to supply more material
- Configuring trust on new root CA at Browser and OpenSSL level
- Self signed certs, openssl command line tutorial
- PEM format (and other formats)
- Role of Metadata, Circle-of-Trust, and Auto-CoT Metadata Exchange based on WKL

Example PEM Cert

—–BEGIN CERTIFICATE—– MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-
jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
Z2VuZXJhbEBpcHNjYS5jb20gQy5JLkYuICBCLUI2MjIxMDY5NTEuMCwG
aXBzQ0EgQ0xBU0VBMSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEuM
aXBzQ0EgQ0xBU0VBMSBDZXJ0aWZpY2F0aW9uIEF1dGhvcml0eTEgMI
DQEJARYRZ2VuZXJhbEBpcHNjYS5jb20wHhcNMDYwNDI2MTUzNjU0W
MTUzNjU0WjCBljELMAkGA1UEBhMCUFQxDzANBgNVBAgTBkxpc2JvY
BxMGTGlzYm9hMRMwEQYDVQQKEwpTeW1sYWJzIFNBMRQwEgYDVQ
aWNlczEYMBYGA1UEAxMPaWRwLnN5bWRlbW8uY29tMSAwHgYJKoZI
ZWxpeEBzeW1sYWJzLmNvbTCBnzANBgkqhkiG9w0BAQEFAAOBjQAw
x5ZjAl06CZcSMVtjoaS2sCbrBq/whwWnuVgbD6gAM9EO9qDDEs9eB5r
iFTWuZy9jdxL5wNgr2Zk8NxytyaznQgAddKLCSqPZh7Dd+U3Z5hoGtL

Sopfj3m6TKOzURgg/Ad/0/cuF9TyCpQprBcpsAECAwEAAaOCAzQwggM
EwQCMAAwEQYJYIZIAYb4QgEBBAQDAgZAMAsGA1UdDwQEAwID+DA
BggrBgEFBQcDATAdBgNVHQ4EFgQUbChmdTnQyOFzW59+dakqD/KX
BBgwFoAUDgdg1DnJG1tdkHsjyNI0nUqaRjkwHAYDVR0RBBUwE4ERZm
bGFicy5jb20wHAYDVR0SBBUwE4ERZ2VuZXJhbEBpcHNjYS5jb20wcgY
QgENBGUWY09yZ2FuaXphdGlvbiBJbmZvcm1hdGlvbiBOT1QgVkFMSU
TEFTRUExIFNlcnZlciBDZXJ0aWZpY2F0ZSBpc3N1ZWQgYnkgaHR0cHM
aXBzY2EuY29tLzAvBglghkgBhvhCAQIEIhYgaHR0cHM6Ly93d3cuaXBzY
L2lwc2NhMjAwMi8wQwYJYIZIAYb4QgEEBDYWNGh0dHBzOi8vd3d3Lr
bS9pcHNjYTIwMDIvaXBzY2EyMDAyQ0xBU0VBMS5jcmwwRgYJYIZIAYb
N2h0dHBzOi8vd3d3Lmlwc2NhLmNvbS9pcHNjYTIwMDIvcmV2b2Nhd
QTEuaHRtbD8wQwYJYIZIAYb4QgEHBDYWNGh0dHBzOi8vd3d3Lmlwc
cHNjYTIwMDIvcmVuZXdhbENMQVNFQTEuaHRtbD8wQQYJYIZIAYb4Q
dHBzOi8vd3d3Lmlwc2NhLmNvbS9pcHNjYTIwMDIvcG9saWN5Q0xBU
MIGDBgNVHR8EfDB6MDmgN6A1hjNodHRwOi8vd3d3Lmlwc2NhLmN

MDIvaXBzY2EyMDAyQ0xBU0VBMS5jcmwwPaA7oDmGN2h0dHA6Ly93
c2NhLmNvbS9pcHNjYTIwMDIvaXBzY2EyMDAyQ0xBU0VBMS5jcmwwP
AQEEJjAkMCIGCCsGAQUFBzABhhZodHRwOi8vb2NzcC5pcHNjYS5jb20
SIb3DQEBBQUAA4GBACan4TGRFHayR38xPkMabzwz9VmCbm0uwPxk
jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuqcrn6w
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
—–END CERTIFICATE—–

Example PEM Cert And Private Key As Used by ZXID

——BEGIN CERTIFICATE—— MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-
jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
(snip) SIb3DQEBBQUAA4GBACan4TGRFHayR38xPkMabzwz9VmCbm0u
jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuqcrn6v
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
——END CERTIFICATE——

——BEGIN RSA PRIVATE KEY—— MIICXQIBAAKBgQDTq7HHlmMCXToJlxl
oMMSz14HmfiUcZjxL3iIVNa5nL2N3EvnA2CvZmTw3HK3JrOdCAB10os.
8nEpzyUJWXpCs9K+kuuJAkAm0b523XnsJmsipA+ZDdyqrUjKDo6WH3
/GeJEXxqIwfcj2lZLp/iIRvG7ICjN/rdWoNImF3HVBRS ——END RSA
PRIVATE KEY——

Example Metadata

<md:EntityDescriptor xmlns:md="urn:oasis:names:tc:SAML:2.0:metada
entityID="https://idp1.zxidp.org:8443/zxididp?o=B"> <md:IDPSSODes
WantAuthnRequestsSigned="1" errorURL="https://idp1.zxidp.org:8443
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
<md:KeyDescriptor use="encryption"> <ds:KeyInfo xmlns:ds="http://w
<ds:X509Data> <ds:X509Certificate> MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-
jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
(snip) jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuc
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
<md:KeyDescriptor use="signing"> <ds:KeyInfo xmlns:ds="http://www
<ds:X509Data> <ds:X509Certificate> MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-

jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
(snip) jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuc
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
<md:ArtifactResolutionService Binding="urn:oasis:names:tc:SAML:2.0:
Location="https://idp1.zxidp.org:8443/zxididp?o=S"/> <md:SingleLog
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp1.zxidp.org:8443/zxididp?o=Q" Re-
sponseLocation="https://idp1.zxidp.org:8443/zxididp?o=Q"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindi
Location="https://idp1.zxidp.org:8443/zxididp?o=S"/> <md:ManageNa
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp1.zxidp.org:8443/zxididp?o=Q" Re-
sponseLocation="https://idp1.zxidp.org:8443/zxididp?o=Q"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bi

Location="https://idp1.zxidp.org:8443/zxididp?o=S"/> <md:NameIDFo
format:persistent</> <md:NameIDFormat>urn:oasis:names:tc:SAML:2.
format:transient</> <md:SingleSignOnService Binding="urn:oasis:nam
Redirect" Location="https://idp1.zxidp.org:8443/zxididp?o=F"/></>
<md:SPSSODescriptor AuthnRequestsSigned="1" WantAssertion-
sSigned="1" errorURL="https://idp1.zxidp.org:8443/zxididp?o=E"
protocolSupportEnumeration="urn:oasis:names:tc:SAML:2.0:protocol"
<md:KeyDescriptor use="encryption"> <ds:KeyInfo xmlns:ds="http://w
<ds:X509Data> <ds:X509Certificate> MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-
jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
(snip) jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuc
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
<md:KeyDescriptor use="signing"> <ds:KeyInfo xmlns:ds="http://www

<ds:X509Data> <ds:X509Certificate> MIIGWTCCBcKgAwIBAgIDA-
JEBMA0GCSqGSIb3DQEBBQUAMIIBEjELMAkGA1UEBhMC RVMxE-
jAQBgNVBAgTCUJhcmNlbG9uYTESMBAGA1UEBxMJQmFyY2Vsb25hM
VQQKEyBJUFMgQ2VydGlmaWNhdGlvbiBBdXRob3JpdHkgcy5sLjEuMC
(snip) jkSenPpwpvomvNfp4G0WJdavd7KnZBbMbnKx1qTMgge/ftBnuc
aHftQ+r2gFYiVX4HEa6NU5AgpiQjme0Vh3Hzs228lVllgsFqv6YbdIyTYIU
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindi
Redirect" Location="https://idp1.zxidp.org:8443/zxididp?o=Q"
ResponseLocation="https://idp1.zxidp.org:8443/zxididp?o=Q"/>
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindi
Location="https://idp1.zxidp.org:8443/zxididp?o=S"/> <md:ManageNa
Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-Redirect"
Location="https://idp1.zxidp.org:8443/zxididp?o=Q" Re-
sponseLocation="https://idp1.zxidp.org:8443/zxididp?o=Q"/>
<md:ManageNameIDService Binding="urn:oasis:names:tc:SAML:2.0:bi

Location="https://idp1.zxidp.org:8443/zxididp?o=S"/> <md:NameIDFo
format:persistent</> <md:NameIDFormat>urn:oasis:names:tc:SAML:2.
format:transient</> <md:AssertionConsumerService Binding="urn:oas
POST-SimpleSign" Location="https://idp1.zxidp.org:8443/zxididp?o=P
index="5"/> <md:AssertionConsumerService Binding="urn:oasis:name
Location="https://idp1.zxidp.org:8443/zxididp?o=P" index="4"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.
Location="https://idp1.zxidp.org:8443/zxididp?o=S" index="3"/>
<md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.
POST" Location="https://idp1.zxidp.org:8443/zxididp?o=P" in-
dex="2"/> <md:AssertionConsumerService Binding="urn:oasis:names:
Artifact" Location="https://idp1.zxidp.org:8443/zxididp" in-
dex="1"/></></></>

Key Concepts

- SP - IdP
- SP/PEP - PDP
- SP/WSC - WSP
- WSP registering itself
- SP/WSP creating an association for the user
- SP/WSC discovering WSP
- Metadata
  - End Point URLs
  - Signing certificate
  - XML-Enc certificate
  - TLS/SSL certificate
- Metadata import
  - WKL method
- Direct Trust by Listing Metadata

Demo

SP `https://lima.tas3.eu:8443/zxidhlo?o=E`

IdP `https://idpdemo.tas3.eu:8443/zxididp?o=B`

PDP `https://lima.tas3.eu:8443/zxididp?o=S`

IdP Selection

ZXID SP Federated SSO (user NOT logged in, no session)

Login Using New IdP

A new IdP is one whose metadata we do not have yet. We need to know the IdP URL (aka Entity ID) in order to fetch the metadata using the well known location method. You will need to ask the adminstrator of the IdP to tell you what the EntityID is.

IdP URL http://r90s.germany.sun.com:8080/amserver-fep-sym-saml-idp Login (A2) Login (P2)

Entity ID of this SP (click on the link to fetch the SP metadata): https://sp1.zxidsp.org:8443/zxidhlo?o=B

Login Using Known IdP

Login to https://lolo:8681/idp.xml (A2) Login to https://lolo:8681/idp.xml (P2)

Login to https://a-idp.liberty-iop.org:8881/idp.xml (A2) Login to https://a-idp.liberty-iop.org:8881/idp.xml (P2)

Login to https://idp.symdemo.com:8880/idp.xml (A2) Login to https://idp.symdemo.com:8880/idp.xml (P2)

Technical options

☑ Create federation, NID Format: Persistent

zxid.org, 0.18 1178728139 libzxid (zxid.org)

Login at IdP

symLABS

e-nabling your business

Symlabs Federated Identity Access Manager

DirectoryScript

Welcome to Id Provider "IdP3 A" Home Login

You may login using various methods (pick your poison)

(be sure browser accepts cookies from the same domain)

1. Cookie login Username: sue Password: ***

If any web site (SP) asks...

The *IdP URL* (Provider ID/Entity ID) of this IdP is https://a-idp.liberty-iop.org:8881/idp.xml

You can cut and paste the above URL to any web site that allows Single Sign-On using *IdP URL* or "Any IdP" or "Other IdP". This mechanism allows the web site (SP) to dynamically join the Circle of Trust of this IdP. This is called *Auto-CoT*.

SSO Successful: Protected Page

SAML Hello World in PHP

- 38 lines of PHP code of which only 22 do something (rest are comments or HTML)
- Complete
  - All profiles are handled
  - Single Logout handled
  - Well Known Location (WKL) metadata exchange handled
- Hides SAML protocol details
- This Hello World can be cut-and-pasted into any PHP application

Initialization once

```
01 <?
02 dl("php_zxid.so");  # Pull in module (.so file)
03 # CONFIG: You must have created /var/zxid directory
04 # CONFIG: You must edit the URL to match your domain
05 $conf = "PATH=/var/zxid/
              &URL=https://sp1.zxidsp.org:8443/zxidhlo.ph
06 $cf = zxid_new_conf_to_cf($conf);
07 ?>
```

- PATH configuration means multiple instances of ZXID can coexist (e.g. virtual hosting of web sites)
- URL configuration determines provider ID, can also be configured via `/var/zxid/zxid.conf`

Per protected page or until session is boot-strapped

```
08 <?
09 $qs = $_SERVER['REQUEST_METHOD'] == 'GET'
10      ? $_SERVER['QUERY_STRING']
11      : file_get_contents('php://input');
12 $res = zxid_simple_cf($cf, -1, $qs, &ses, 0x1814);
13
14 switch (substr($res, 0, 1)) {
15 case 'L': header($res); exit;
16 case '<': header('Content-type: text/xml'); echo $re
```

- Read input and call *zxid_simple()* to handle SAML protocol details
- Act on outcome of *zxid_simple()* as indicated by the first letter
  - L: protocol requires redirect, perform it
  - <: Send out XML data (such as Metadata or SOAP response)

The IdP Selection Page

```
17 case 'n': exit;   # Already handled, do nothing furt
18 case 'e':
19 ?>
20 <title>Please Login Using IdP</title>
21 <h1>Please Login Using IdP</h1>
22 <?=zxid_idp_select_cf($cf, null, 0x1800)?>
23 <?
24 exit;
```

- e: indicates that IdP Selection page needs to be rendered
- *zxid_idp_select()* generates the ZXID standard form
- Altenatively you could supply your own HTML for the form as long as you respect the form field naming convention

Login Successful Case

```
25 case 'd': break;  # Logged in case -- continue after
26 default:  die("Unknown zxid_simple() res($res)");
27 }
28
29 # Parse the LDIF in $res into a hash of attributes $
30
31 foreach (split("\n", $res) as $line) {
32     $a = split(": ", $line);
33     $attr[$a[0]] = $a[1];
34 }
35 ?>
```

- d: login successful, return data is LDIF entry with attributes of SSO

Protected Content with Single Logout and Defederate Buttons

```
36 <title>Protected content, logged in</title>
37 <h1>Protected content, logged in as <?=$attr['cn']?>
38 <?=zxid_fed_mgmt_cf($cf, null, -1, $attr['sesid'], 0
```

- *zxid_fed_mgmt()* generates the Single Log-Out buttons
- This is the place to bootstrap your application's own session

Login Successful: Returned LDIF

```
dn: idpnid=Pa45XAs2332SDS2asFs,affid=https://idp.dem
objectclass: zxidsession
affid: https://idp.demo.com/idp.xml
idpnid: Pa45XAs2332SDS2asFs
authnctxlevel: password
sesid: S12aF3Xi4A
cn: Joe Doe
```

- The LDIF entry is used as convenient format for passing attribute-value pairs from *zxid_simple()* to application
- Some "attributes" are synthesized, others come actually from assertion

Thank You

Sampo Kellomäki (sampo@symlabs.com)
+351-918.731.007

Table 1: SAML and Liberty Open Source Implementations

Product	License	Platform	SAML SP	SAML IdP	WSC	WSP	Disco	People	Interact	Account	Other
ZXID.org	Apache2	C + SWIG	Full	TBA	y	y	WSC	TBA	TBA	TBA	
mod_auth_sam	Apache2	C + SWIG	Full	-	y	y	WSC	TBA	TBA	TBA	
Lasso	GPL2+	C + SWIG	Cert	-	y?	?	WSC				
Authentic	GPL2+	C + Python?	-	Certified	-	-	WSP	?	?	?	
OpenSSO	Java?	pure PHP	Partial	-	-	-	-	-	-	-	
OpenSSO	Java?	Java	Cert	Cert	1.1	1.1	1.1	-	1.1	-	
OpenSAML	?	Java?	Partial	Partial							
OpenLiberty	Apache2	Java	-	-	y	-	WSC	TBA	-	-	AS-WSC
ConorCli	BSD?	C++	-	-	y	-	WSC	WSC?	redir	?	AS-WSC
ConorSvc	BSD?	Java	-	-	-	y	WSP	WSP?	redir	?	AS-WSP

TBA To be announced, on the road map, but not here yet

1.1 Only supports older ID-WSF 1.1 version of the services

C + SWIG C library with language bindings generated usign SWIG

tool. SWIG supports among others C, C++, Perl, PHP, Python, Ruby, and Java language bindings. Generally open source products can be compiled for all popular operating systems such as Unix and Windows.

Additional info available on `openliberty.org`