AAI Account checking

Or how to find and kill zombie users :-)

SWITCH

Serving Swiss Universities

Lukas Hämmerle
lukas.haemmerle@switch.ch

Bern, 16. September 2009

Example scenario

Assume you operate an e-Learning platform for users of multiple Home Organisations. The platform uses the following AAI attributes to automatically create and update

- Unique ID
- Surname
- Given name
- Mail address

Personal Attributes

- Home Organization
- Affiliation

Unpersonal Attributes

But after some months ...

... you will face the problem of Zombie users

= Accounts of users that don’t exist anymore at the Home Organisation they originally came from.

Zombie users are not an AAI but a general problem! But AAI provides a solution for it.

Anti-Zombie strategies

How can one identify zombie users and delete them?

- **The old-fashioned way ;-)**
  http://www.ehow.com/how_2058687_fight-zombies.html
- **The cumbersome way**
  Ask user admin of organization X if user Y still has an account... for all users in question.
- **The non-AAI way**
  Analyze the last-login timestamp of local user accounts.
- **The recommended AAI way**
  Use the persistent-Id/eduPersonTargetedID to make an attribute query. The next slides focus on this approach.

Key idea of the recommended way

The assumption:

We assume that a Home Organisation deletes a user's personal attributes if user leaves organisation.

The approach:

1. Make an attribute query to user's IdP
2. Analyze attributes that are returned for user
3. If any personal attributes exist, assume account still exists

The issue:

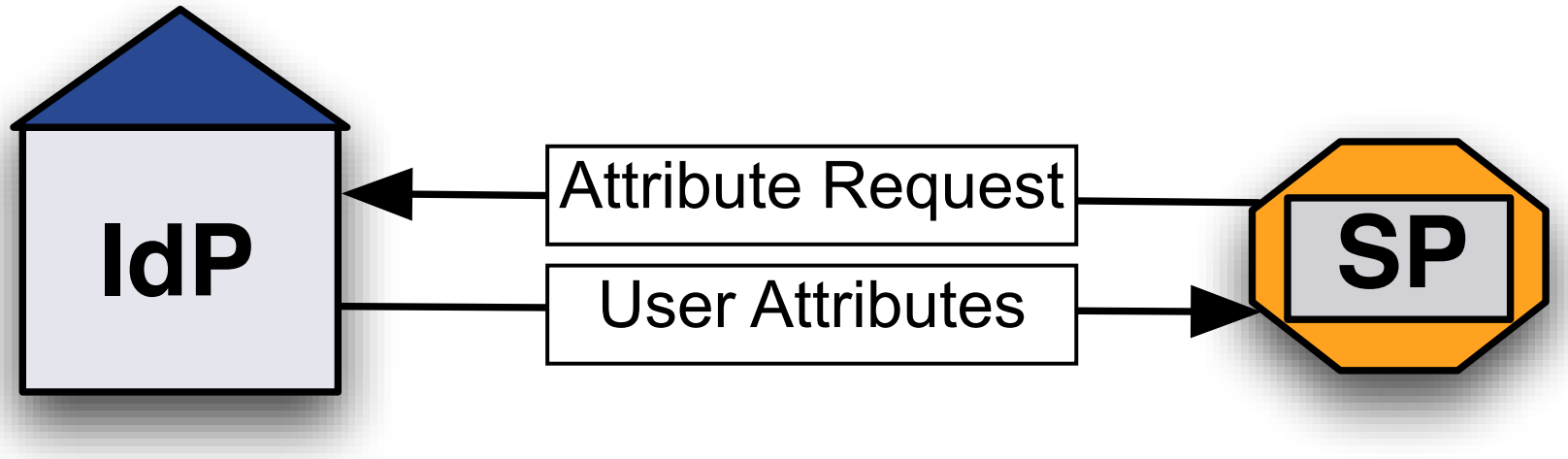
How to easily make an SSL back-channel attribute query to the Identity Provider without the user's involvement?

Thank Scott for the resolvertest binary

Resolvertest binary comes with SP 2.2 or newer. As the name suggests it can be used to resolver/query attributes.

In our case, we would use:

```
resolvertest -saml2 \
-f urn:oasis:names:tc:SAML:2.0:nameid-format:persistent \
-n 26662bf3-f15e-418e-89f4-467788ff650b \
-i https://aai-logon.switch.ch/idp/shibboleth
```



Resulting resolvertest output

Case: Account still exists

```
Shib-Person-surname: Hämmerle
Shib-InetOrgPerson-givenName: Lukas
Shib-SwissEP-UniqueID: 498752@switch.ch
Shib-Person-telephoneNumber: +41 44 268 1505
Shib-SwissEP-HomeOrganizationType: other
Shib-SwissEP-HomeOrganization: switch.ch
```

Personal Attributes

Using this output one could also update local user data.

Case: Account probably doesn't exist anymore

```
Shib-SwissEP-HomeOrganizationType: other
Shib-SwissEP-HomeOrganization: switch.ch
```

Unpersonal Attributes

Why not using a script to make life easier?

We created a script for you:

```
./accountChecking.sh -v \
-i '26662bf3-f15e-418e-89f4-467788ff650b'
Checking NameID: 26662bf3-f15e-418e-89f4-467788ff650b
 against Identity Provider: https://aai-logon.switch.ch/idp/shibboleth
Found personal attribute with uniqueID: 498752@switch.ch
 -> User account with NameID 26662bf3-f15e-418e-89f4-467788ff650b still
exists
```

Feeding a file with multiple NameIDs is also possible.

Output then will be CSV file with result of operation.

(Beta) Script can be downloaded from:

http://www.switch.ch/aai/downloads/accountChecking.sh

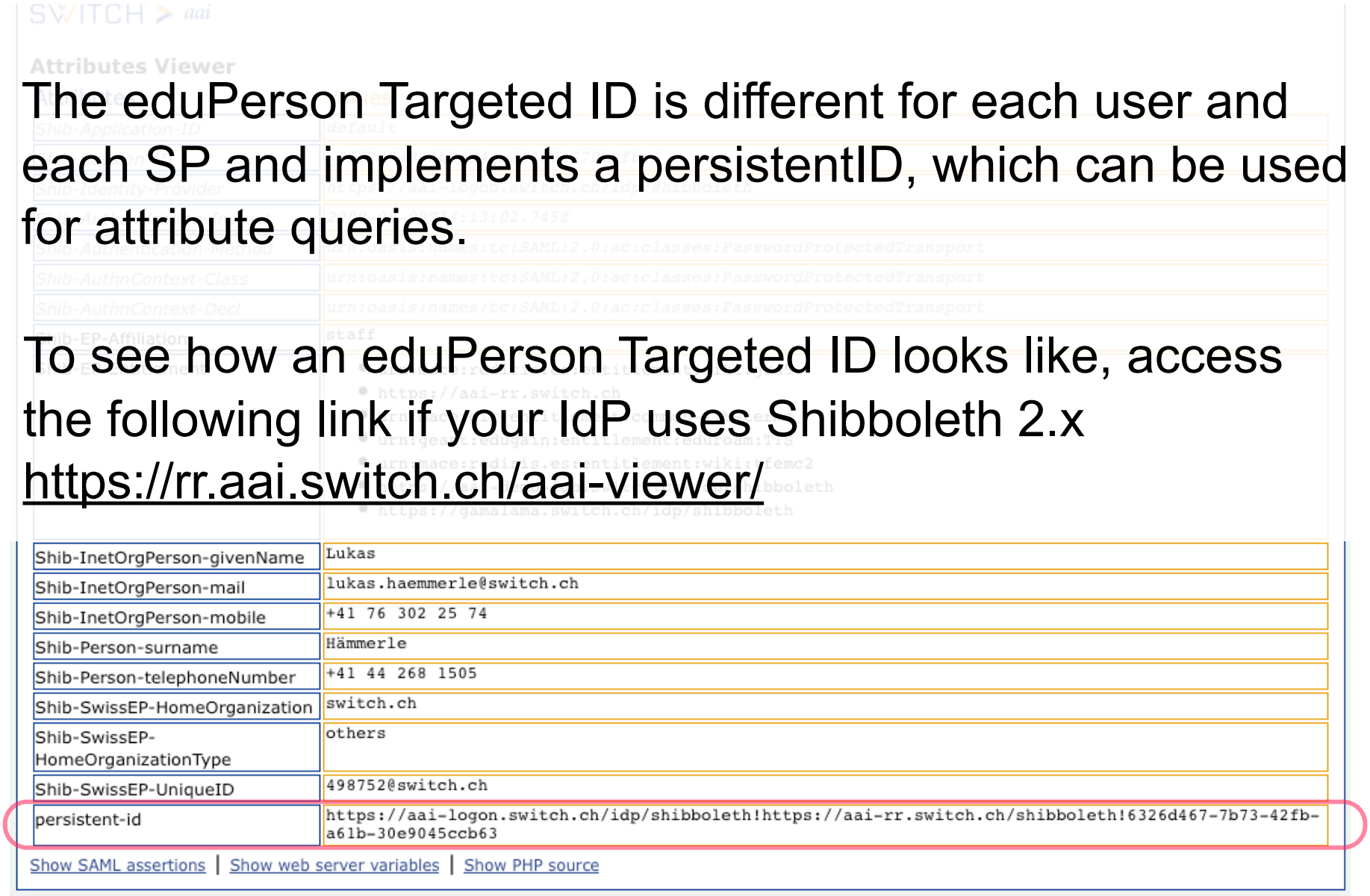
Requirements for account checking/updating

1. Shibboleth Service Provider 2.x is required
2. User's 2.x IdP must have adapted config (see last slide)
3. **One must know the persistent-ID of a user**
   In Resource Registry add eduPerson Targted ID as required attribute to Resource Description (for now):

eduPerson Targted ID is a persistent ID

The eduPerson Targeted ID is different for each user and each SP and implements a persistentID, which can be used for attribute queries.

To see how an eduPerson Targeted ID looks like, access the following link if your IdP uses Shibboleth 2.x
https://rr.aai.switch.ch/aai-viewer/



Shibboleth 2.x IdP configuration change

In attribute-resolver.xml all that has to be added is this:

```
<resolver:PrincipalConnector
  xsi:type="pc:StoredId"
  xmlns:pc="urn:mace:shibboleth:2.0:resolver:pc"
  id="saml2Persistent"
  nameIDFormat="urn:oasis:names:tc:SAML:2.0:nameid-format:persistent"
  storedIdDataConnectorRef="myStoredId" />
```

In the near future, we will send you an email with detailed instructions on how to add the above to the Shibboleth 2.x IdP configuration.

Short Summary

- With Shibboleth 2.x administrators can use:
  - Account existence checking
  - Account data updating
- Only minor configuration change on IdP is needed
- Resource admins who want to use this feature must declare eduPerson Targeted ID attribute in Resource Registry as required for now
  - This value then has to be stored somewhere for later use
  - Use of persistentID instead transientID may become default