AACLI

Attribute Authority, Command Line Interface (AACLI)

The Shibboleth attribute authority (AA) is the part of a provider that's responsible for the inflow and outflow of attributes. Each time an IdP participates in a SAML transaction, the AA undertakes a number of steps to prepare attributes to be sent:

1. The AA collects attributes from source systems
   ⚠️ Note that with JDBC only Application Managed Connections can be tested, since the AACLI does not run in a container.
2. The attributes are processed according to rules and dependencies defined in the resolver;
3. The resulting attributes are filtered according to filter policies, SAML metadata information, and attribute query information.
4. The attributes are then encoded into SAML attribute statements which may be sent to a relying party.

The attribute authority command line interface (AACLI) allows deployers to exercise their configurations and view the information that would likely be sent back to the relying party for a given SAML transaction. As it is not possible to specify **every** piece of information that goes into the attribute authority in a running system, the results are only an approximation of what would really be returned.

Running the Command

The attribute authority command line interface is located in the `$IDP_HOME/bin` directory and is called `aacli.sh` on Unix systems and `aacli.bat` on Windows systems. It may take the following information:

Parameter	Required / Optional	Use
--configDir	Required	Directory containing the configuration information for the system. If not specified and the `IDP_HOME` environment variable is set, defaults to `$IDP_HOME/conf`.
--principal	Required	Principal name (user id) of the person to retrieve the attributes about
--requester	Optional	The SAML entity ID that is requesting the attributes (entity ID of the Service Provider)
--issuer	Optional	The SAML entity ID of the producer/issuer of the attributes
--authnMethod	Optional	The authentication method URI that the principal was authenticated with
--saml1	Optional	A no-value argument that indicates the resulting attributes should be SAML 1 formated instead of SAML 2
--springExts	Optional	Colon-delimited list of files containing Spring extension configurations
--help	Optional	Displays the help message for the tool

Information will be returned in SAML 2 AttributeStatement format (or SAML 1 AttributeStatement format if the `--saml1` parameter is given).

Example Commands

```
> ./aacli.sh --configDir=conf/ --principal=jsmith

OR

> bin\aacli.bat --configDir=conf/ --principal=jsmith --requester=http://example.org/sp --saml1
```

Examples with sample results (Unix)

When no attributes would be released, you will see the "No attribute statement" message:

```
../bin/aacli.sh --configDir=. --principal=jv11
No attribute statement.
```

For your IdP whose AA is configured to release uid, eduPersonPrincipalName, and eduPersonPrimaryAffiliation, successful results will look something like this:

```
../bin/aacli.sh --configDir=. --principal=jv11

<?xml version="1.0" encoding="UTF-8"?><saml:AttributeStatement xmlns:saml="urn:oasis:names:tc:SAML:2.0:
assertion">
   <saml:Attribute FriendlyName="uid" Name="urn:oid:0.9.2342.19200300.100.1.1" NameFormat="urn:oasis:names:tc:
SAML:2.0:attrname-format:uri">
      <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:type="xs:string">jv11</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute FriendlyName="eduPersonPrimaryAffiliation" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.5"
NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:type="xs:string">staff</saml:AttributeValue>
   </saml:Attribute>
   <saml:Attribute FriendlyName="eduPersonPrincipalName" Name="urn:oid:1.3.6.1.4.1.5923.1.1.1.6" NameFormat="
urn:oasis:names:tc:SAML:2.0:attrname-format:uri">
      <saml:AttributeValue xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001
/XMLSchema-instance" xsi:type="xs:string">jv11@cornell.edu</saml:AttributeValue>
   </saml:Attribute>
</saml:AttributeStatement>
```

Example command to see what attributes would be released to the testshib.org service provider, whose entity ID is https://sp.testshib.org/shibboleth-sp:

```
../bin/aacli.sh --principal=jv11 --configDir=../conf --requester=https://sp.testshib.org/shibboleth-sp
```

If you receive an exception when you run the aacli.sh script, you may be able to find out more information about the error in the IdP's log file, usually in $IDP_HOME/logs/idp-process.log.