Eszközök

- Eszközök
- SamlSign
- Xmltooling Log4cpp patch

Eszközök

- uApprove: User-consent (beleegyezési nyilatkozat) eszköz
- SamlSign: Metaadat aláíró és ellenőrző eszköz

SamlSign

Parancssoros eszköz, melyhez debian alatt az `opensaml2-tools` csomagot kell telepíteni. A program kétféle üzemmódban képes működni: **metaadat aláírása** és **metaadat ellenőrzése**.

Metaadat aláírása

```
samlsign -s -k /path/to/mainkey.key -f /path/to/metadatatosign.xml
```

Alapértelmezés szerint a samlsign az eredményeket az alapértelmezett kimenetre írja ki (STDOUT), így célszerű ezt egy új fájlba átirányítani:

```
samlsign -s -k /path/to/mainkey.key -f /path/to/metadatatosign.xml >
/path/to/metadatasigned.xml
```

Metaadat ellenőrzése

```
samlsign -c /path/to/maincert.crt -f /path/to/metadatatosign.xml
```

Samlsign legfontosabb kapcsolói

- -s ez határozza meg, hogy aláírunk, vagy ellenőrzünk. Ha megadtuk kapcsolóként, akkor a program megpróbálja aláírni a megadott xml fájlt, ha nem, akkor ugyanezt a fájlt ellenőrizni fogja.
- -f az ellenőrzendő/aláírandó fájl elérhetősége **abszolút útvonallal** megadva
- -k a privát kulcs elérhetősége **abszolút útvonallal** megadva
- -c az ellenőrzésre használt publikus kulcs elérhetősége **abszolút útvonallal** megadva
- További részletes leírás a samlsign man oldalán

További fontos tudnivalók A samlsign nem szereti a metadatában szereplő `Organization`-nel kapcsolatos adatokat, mivel ilyen tag-ekben kötelezően megadandó `xml:lang` attribútumot `lang`-ra alakítja át, ami által viszont nem lesz érvényes (valid) maga a metaadat, így pl. a shibboleth sem fog tudni vele mit kezdeni. A **megoldás** (nem szép, de hasznos): az aláírás előtt álló metaadatokból ki kell szedni az `Organization`-nel kapcsolatos adatokat. Ezek után már gond nélkül aláírja és az eredmény is érvényes lesz.

Apró trükk a privát kulcs kinyerésére `jks`-ből

Tekintettel arra, hogy a `keytool` nem teszi lehetővé a privát kulcs kihalászását JavaKeystore-ból, így külső segítséget kell igénybe vennünk. A segédalkalmazás `ExportPrivateKey` névre hallgat, és innen letölthető egy darab zip fájl. Használata rendkívül egyszerű:

```
java -jar ExportPrivateKey.zip {jks fájl elérhetősége} JKS {jks jelszó} {alias} {célfájl}
```

Ezek után a létrehozott kulccsal már használhatjuk is a samlsign-t.

Xmltooling Log4cpp patch

```
--- xmltooling/soap/impl/SOAPClient.cpp.orig    2008-03-14 23:50:29.000000000 +0100
+++ xmltooling/soap/impl/SOAPClient.cpp 2008-04-25 13:14:29.000000000 +0200
@@ -89,8 +89,11 @@
     XercesJanitor<DOMDocument> janitor(doc);

     Category& log = Category::getInstance(XMLTOOLING_LOGCAT".SOAPClient");
-    if (log.isDebugEnabled())
-        log.debugStream() << "received XML:\n" << *(doc->getDocumentElement()) <<
logging::eol;
+    if (log.isDebugEnabled()) {
+        string serializedXml;
+        XMLHelper::serialize (doc->getDocumentElement(),serializedXml,false);
+        log.debugStream() << "received XML:\n" << serializedXml << logging::eol;
+    }

     auto_ptr<XMLObject> xmlObject(XMLObjectBuilder::buildOneFromElement(doc-
>getDocumentElement(), true));
     janitor.release();
```