X.509 Certificates for SAML

Shibboleth and public-key cryptography

SWITCH

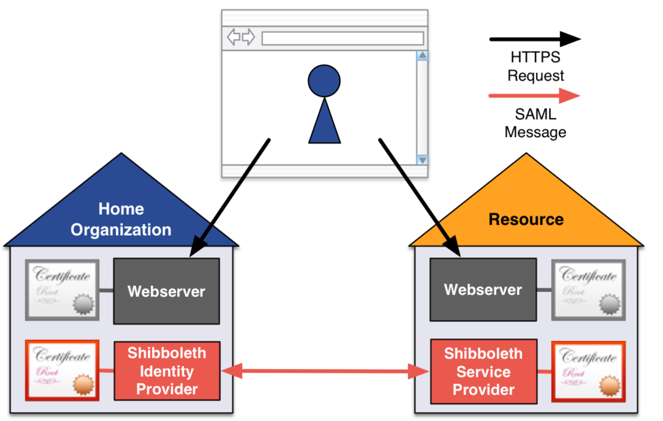
SWITCHaai Team
aai@switch.ch

IdP-to-SP communication

- To secure communications between an IdP and an SP (or vice versa), each entity needs a key pair:
  - for authenticating/signing SAML messages
  - for encrypting SAML messages
- The most popular public-key algorithm today is RSA, typically with a key size of 2048 bits
- X.509 is a standardized, ubiquitous format for public-key data structures, so it's convenient for use as a container for public keys in the SAML world

X.509 certificates: a bird's eye view

- certificate = public key + name + signature
- the two "certificate worlds" of a Shibboleth SP (or IdP):
  - SSL/TLS between the user's browser and the Web server
  - XML Signature and XML Encryption for SAML messaging between the SP and the IdP



X.509 certificate validation

- for SSL/TLS, validating the certificate's name and signature by the browser is absolutely crucial
  - browsers validate the chain against a built-in set of root certificates by verifying the signatures, and *must* make sure that the name is matching (would be vulnerable to MiTM otherwise)
- for the SAML world and the SWITCHaai federation, certificates are fully embedded in the federation metadata, and only the public key is considered by Shibboleth (X.509 is used as a convenient container format)
  - name and signature are basically thrown away

SWITCHaai requirements for SAML embedded certificates

- *either* a self-signed certificate with minimal subject information and X.509v3 extensions (**recommended**, can be created with the `keygen` script bundled with the Shibboleth SP)
- *or* a certificate signed by a well-known CA, with organization validation (OV), chaining to a root trusted by either Microsoft or Mozilla
- further reading:
  - https://www.switch.ch/aai/support/certificates/certificate-acceptance/
    https://www.switch.ch/aai/support/certificates/embeddedcerts-requirements/

Examining X.509 certificates

- Details in text format:

```
openssl x509 -in /etc/ssl/certs/ssl-cert-snakeoil.pem -text
```

- Display the fingerprint only:

```
openssl x509 -in /etc/ssl/certs/ssl-cert-snakeoil.pem \
-noout -fingerprint [-sha256]
```

- Dump a pure Base64 block (e.g. when grabbing from XML metadata – paste into console and end with Ctrl-D):

```
openssl base64 -d | openssl x509 -inform der -text
```