SWAMID 2.0

howto

"Vad var det för fel på 1.0?"

Andy Polyakov

Problemen vi försöker lösa är...

- Obekväm hantering av attributrelease
- Metadata saknar giltighetstid
- Jobbigt med interfederation

Vad vi har försökt åstadkomma i 2.0:

- Balans mellan arbetsinsats och kontroll av attributrelease
- "Always On"-interfederation
- Införa MDUI och andra extensions
- Explicita kontroller för caching
- Metadata Terms-of-Use
- Acceptans från firmatecknare

Teknisk migrering till SWAMID 2.0

- Allt Nytt!
  - nya metadata URL(er)
  - nya metadata-element
  - ny discovery (... jbn har pratat om det redan)
  - ny hantering av interfederation
  - ny hantering av attributrelease

Nya URLer

- swamid-2.0.xml
  - alla entitys som SWAMID registrerar
  - alla entitys från alla downstreams
- swamid-idp.xml
  - alla IdPer registrerade i 1.0 och 2.0
- swamid-idp-transitive.xml
  - alla IdPer i 1.0 och 2.0 inklusive interfederation

Nya metadata-element

- cacheDuration
  - 8h
    - läs metadata var 8:e timme
- validUntil
  - 1 dygn
    - metadata är giltigt 1 dygn

Ny hantering av inter-federation

- Förr

varje SP och IdP måste lägga till kalmar,

edugain etc manuellt

- Nu
  - alla interfederation downstreams är inkluderat i swamid-2.0.xml

Vad ska jag göra med min SP?

Byt swamid-1.0.xml till swamid-idp-transitive.xml

Vad ska jag göra med min IdP?

Lägg till swamid-2.0.xml

Ny hantering av attributrelease

- Alternativ 1:
  - Gör samma attributrelease-regel för https://md.swamid.se/md/swamid-2.0.xml som du idag har för swamid-1.0.xml.
    - FARLIGT!

Exempel (attribute-filter.xml)

```
<AttributeFilterPolicy id="releaseStandardAttributesToFederations">
        <PolicyRequirementRule xsi:type="basic:OR">
            <basic:Rule xsi:type="saml:AttributeRequesterInEntityGroup"
groupID="http://md.swamid.se/md/swamid-1.0.xml" />
            <basic:Rule xsi:type="saml:AttributeRequesterInEntityGroup"
groupID="http://md.swamid.se/md/swamid-2.0.xml" />
        </PolicyRequirementRule>
        <!-- rules go here -->
</AttributeFilterPolicy>
```

Ny hantering av attributrelease

- Alternativ 2:
  - Gör manuell attributrelease för varje tjänst.
    - Säkert
    - Arbetsintensivt

Ny hantering av attributrelease (2)

- Alternativ 3:
  - Använd entity-attribute-baserad filtrering!
    - Enkelt
    - Säkert

Exempel (i metadata...)

```
<EntityDescriptor entityID="https://foo.example.com">
    <Extensions>
      <EntityAttributes>
        <Attribute
          NameFormat="urn:oasis:names:tc:SAML:2.0:attrname-format:uri"
          Name="http://macedir.org/entity-category">
         <AttributeValue>unique-identifiers</AttributeValue>
        </Attribute>
      </EntityAttributes>
    </Extensions>
  </EntityDescriptor>
```

Exempel (i attribute-filter.xml)

```
<AttributeFilterPolicy id="identifiersReleasePolicy">
    <PolicyRequirementRule xsi:type="saml:AttributeIssuerEntityAttributeExactMatch"
                attributeName="http://macedir.org/entity-category"
                attributeValue="unique-identifiers" />

      <AttributeRule attributeID="eduPersonPrincipalName">
         <PermitValueRule xsi:type="basic:ANY" />
      </AttributeRule>
      <AttributeRule attributeID="email">
         <PermitValueRule xsi:type="basic:ANY" />
      </AttributeRule>

  </PolicyRequirementRule>
</AttributeFilterPolicy>
```

next...

- PEER-installation (självservice)
- OpenID Connect GW
- Gäst-IdP
- OTP ?
- Token Service ?
  - SAML -> applikationslösenord, cert, etc etc
- ??