Shibboleth SP asennus

Asennusvaihtoehdot

Shibboleth Service Provider -palvelimen asentamiseen voi käyttää automaatiotyökalu Ansiblelle suunniteltua esimerkkiroolia tai tehdä asennus käsin esimerkiksi allaolevien vaiheiden mukaisesti. Ansible-rooli ja alla kuvatut vaiheet ovat esimerkkejä ohjelmiston asentamisesta. Haka-verkoston kumppani tai jäsen vastaa palvelusopimuksen mukaisesti, että asennus noudattaa verkoston vaatimuksia ja että palvelua ylläpidetään niin, että vaatimusten mukaisuus säilyy vielä ensiasennuksen jälkeen.

Löydät ansible-roolin seuraavasta linkistä: https://github.com/CSCfi/ansible-role-shibboleth-sp

Asennus

Tässä ohjeessa käydään läpi Shibboleth SP ohjelmiston asentaminen RedHat -käyttöjärjestelmälle käyttäen valmiiksi paketoituja asennuspaketteja. Useimmiten jokaisella organisaatiolla on oma jakelukanavansa, josta ohjelmistot asennetaan. Tiedustelkaa sitä omalta IT-ylläpidoltanne. Kanavien käyttö on suotavaa koska tällöin myös ohjelmiston päivittäminen voidaan hoitaa automaattisesti ja hallitusti.

Shibboleth SP asennetaan Apachen HTTP-palvelimen kylkeen, jolle paketissa tulee "mod_shib_22.so" -moduli. Kyseinen moduli toimii ainoastaan Apache versio 2.2:sen kanssa. Tällä hetkellä valmiiksi paketoituna saadaan RedHat 6:seen asti vain versiossa Apachen versiossa 2.2 toimiva moduli, seiskasta eteenpäin käytetään Apachen versio 2.4:sta sekä mod_shib_24.so:ta.

Kun jakelukanavat on konfiguroitu kuntoon voidaan siirtyä itse shibbolethin asentamiseen.

```
yum install shibboleth
```

Konfigurointi

Kun shibboleth on asennettu voidaan siirtyä konfigurointiin. Konfiguraatiotiedostot löytyvät ennalta arvattavasta lokaatiosta, "/etc/shibboleth" hakemistosta.

shibboleth2.xml

ApplicationDefault elementin olennaisin kohta on attribuutin entityID arvo, tämä on shibboleth SP:n tuleva entityID jota käytetään jatkossa monessakin paikassa. Aseta arvoksi oman ympäristösi mukainen arvo. Hakassa myös vaaditaan Autentikointi pyyntöjen allekirjoitus, tämä tapahtuu asettamalla "signing" attribuutille arvon "front" tai "true".

```
<ApplicationDefaults entityID="https://testsp.funet.fi/shibboleth" REMOTE_USER="eppn persistent-id targeted-id"
signing="front" encryption="false">
```

SSO elementtiin konfiguroidaan kirjautumislähde. Kirjautumislähde voi olla joku yksittäinen IdP tai sitten Discovery Service (ent. WAYF). SSO elementtiin annetaan siis joko attribuutti entityID yksittäisen IdP:n tapauksessa tai sitten discoveryURL.

```
Yksittäinen IdP

<SSO entityID="https://idp.csc.fi/idp/shibboleth"> SAML2 </SSO>

tai Haka-test-federaation (testipuolen) Discovery Service

<SSO discoveryProtocol="SAMLDS" discoveryURL="https://testsp.funet.fi/shibboleth/WAYF"> SAML2 </SSO>

tai Haka-federaation (tuotannon) Discovery Service

<SSO discoveryProtocol="SAMLDS" discoveryURL="https://haka.funet.fi/shibboleth/WAYF"> SAML2 </SSO>
```

Errors elementtiin tulee yhteystiedot jotka annetaan virhesivulla. Tämä on hyvä täyttää jotta käyttäjät osaavat ottaa yhteyttä oikeaan paikkaan. Valitettavan usein tämä jätetään täyttämättä ja helposti otetaan yhteyttä joko federaation tai IdP:n ylläpitoon vaikka monesti ongelma on paikallisella SP:llä.

```
 <Errors supportContact="haka@csc.fi" logoLocation="/shibboleth-sp/logo.jpg" styleSheet="/shibboleth-sp/main.
css"/>
```

MetadataProvider elementti on olennainen osa Shibboleth SP:n toimintaa, tässä elementissä määritellään luottosuhteiden lähteet, ovat ne sitten paikallisia metadatoja tai ulkoisia. Oleellista on että allekirjoitus tarkistetaan aina vaikka metadata haettaisiinkin SSL:n takaa. Federaatioissa IdP:n ja SP:den määrät elävät ja on hyvä pitää huolta siitä että metadatat päivitetään säännöllisesti (vähintään kerran vuorokaudessa). Metadatoilla on myös parasta ennen ajat ja vanhentunutta metadataa ei saa käyttää (validUntil), tämänkin takia automaattisesta päivityksestä on hyvä huolehtia.

Alla olevassa esimerkissä haetaan Haka-federaation tuotannon metatiedot kerran tunnissa ja allekirjoitus tarkistetaan haka-sign-v8.pem varmenteella. Tämän lisäksi esimerkissä näytetään myös kuinka voidaan rajoittaa luottosuhde vain yhteen IdP:hen whitelistausta hyväksi käyttäen. Esimerkissä myös tarkistetaan että metadasta löytyy validiteetti aika ja myös ettei tämä ylitä 30:ntä päivää, muussa tapauksessa se hylätään. Aika annetaan sekunteina.

```
Haka-federaation metadata (tuotanto)

<MetadataProvider type="XML" url="https://haka.funet.fi/metadata/haka-metadata.xml" backingFilePath="haka-
metadata.xml" reloadInterval="3600">
 <MetadataFilter type="Signature" certificate="/etc/pki/tls/certs/haka-sign-v8.pem"/>
 <MetadataFilter type="Include">
  <Include>https://idp.csc.fi/idp/shibboleth</Include>
 </MetadataFilter>
 <MetadataFilter type="RequireValidUntil" maxValidityInterval="2592000"/>
</MetadataProvider>

tai Haka-test-federaatio (testi)

<MetadataProvider type="XML" url="https://haka.funet.fi/metadata/haka_test_metadata_signed.xml"
backingFilePath="haka_test_metadata_signed.xml" reloadInterval="3600">
 <MetadataFilter type="Signature" certificate="/etc/pki/tls/certs/haka_testi_2018_sha2.crt"/>
 <MetadataFilter type="Include">
  <Include>https://testidp.funet.fi/idp/shibboleth</Include>
 </MetadataFilter>
 <MetadataFilter type="RequireValidUntil" maxValidityInterval="2592000"/>
</MetadataProvider>
```

CredentialResolver elementissä määritellään avainparit joita käytetään IdP:lle lähetettävien viestien allekirjoittamiseen sekä vastaanotettavien viestien salauksen purkamiseen. Näiden avainten pitää vastata avaimia, jotka olet myös asettanut niiden federaatioiden metatietoihin joihin SP:llä on luottosuhde.

```
  <CredentialResolver type="File" key="/etc/pki/tls/private/testsp.funet.fi.pem" certificate="/etc/pki/tls/certs
/testsp.funet.fi.crt"/>
```