Ilpo Lehtinen

Haka-federaation integrointi CyberSecurityGameen

Opinnäytetyö

Tietotekniikka

Toukokuu 2016

KYAMK
University of Applied Sciences

KYAMK
University of Applied Sciences

Tekijä/Tekijät	Tutkinto	Aika
Ilpo Lehtinen	Insinööri	Toukokuu 2016

Opinnäytetyön nimi

Haka-federaation integrointi CyberSecurityGameen

36 sivua
4 liitesivua

Toimeksiantaja

Kyamk

Ohjaaja

Lehtori Niina Salmi

Tiivistelmä

Työn tavoitteena on toteuttaa Kyamkin CyberSecurityGame - projektiin autentikaatiomoduuli, joka käyttää käyttäjälähteenään HAKA-federaatiota. Työssä tutustutaan Shibboleth - palvelinten toimintaan, rakennetaan laboratorioympäristössä federoimaton Shibboleth Service Provider - Identity Provider -kaksikko ja pohditaan miten lopullinen moduuli voisi toimia.

Työ keskittyy Shibboleth - palvelinten pystyttämiseen. Työssä käsitellään myös LDAP-protokollan toimintaa Microsoftin Active Directoryn tulkitsemana.

Shibboleth-palvelimista on saatavilla Windows-versiot, mutta työssä palvelimet asennetaan CentOS7 - Linuxille helpomman pakettienhallinnan ja käytettävyydeltään parempien tekstieditorien vuoksi. Active Directory -palvelimena käytetään Windows Server 2012R2:a. HTTP-palvelimena toimii Apache. CyberSecurityGamea ohjelmoidaan Visual C++ 2015:llä ja Marmaladella.

Työn lopputuloksena oli laboratorioympäristössä toimiva Shibboleth-esimerkkiympäristö. Työtä jatkokehittäessä pitäisi toteuttaa testihakaan liitettävä, jatkuvasti päällä oleva Shibboleth - palvelin, sessiot Shibbolethille ulkoistava välityspalvelin. Tämän koneen avulla lopullisen autentikaatiomoduulin voisi toteuttaa ja testata.

Asiasanat
shibboleth, Linux, Windows, xml, saml, apache

KYAMK
University of Applied Sciences

Author (authors)	Degree	Time
Ilpo Lehtinen	Bachelor of Engineering	May 2016
Thesis Title		
Integrating CyberSecurityGame and Haka - Federation		36 pages 4 pages of appendices
Commissioned by		
KyUAS		
Supervisor		
Lecturer Niina Salmi		

Abstract

The goal of this thesis was to implement an authentication module for KyUAS' CyberSecurityGame project. The module would authenticate users against Finnish HAKA-federation. Haka is the identity federation of the Finnish universities, polytechnics and research institutions. The work describes how Shibboleth servers work, how an unfederated test environment is constructed with both Service Provider and Identity Provider - servers. The report ends with ideas on how the final authentication module would work.

The thesis focuses on constructing Shibboleth servers. The report also explains LDAP-protocol as interpreted by Microsoft's Active Directory – server. The Shibboleth servers were set up in CentOS7 – Linux environment because of the ease of package management and better usability of the text editors. Windows 2012R2 was used as the operating system for the Active Directory virtual machine. Apache was used as the http server. CyberSecurityGame was programmed with Visual C++ 2015 and Marmalade.

When developing this project further, a test Shibboleth server should be created and joined to the Haka -test federation. Afterwards the final authentication module could be developed and tested against this test server. When the authentication module is complete, the test Shibboleth server could easily be moved to the production federation.

Keywords

shibboleth, Linux, Windows, xml, saml, apache

SISÄLLYS

Termit, lyhenteet

Istunto	Palvelimen ja asiakkaan välinen keino ylläpitää tilaa.
TCP	Tietoliikenneprotokolla, jolla tietokoneet voivat lähettää toisilleen viestejä verkossa. TCP-protokolla on virheenkorjaava, jokaisesta vastaanotetusta paketista lähetetään takaisin kuittaus. Jos kuittausta ei kuulu, paketti lähetetään uudestaan.
Marmalade	Kirjasto, joka tekee alustariippumattomien kaksi- ja kolmiulotteisten pelien tekemisestä helpompaa.
C++	Suorituskykyinen olio-ohjelmointikieli.
Raknet	Verkkokirjasto, joka tekee TCP-protokollan puhumisesta C++:lla helpompaa.
JDBC	Java DataBase Connection, Sunin kehittämä protokolla, jolla java-ohjelmistot juttelevat tietokannoille
SSO	Single Sign-On, sarja palveluja, joista yhteen kirjauduttuaan käyttäjä on kirjautunut kaikkiin.
RPM	Red Hatin käyttöjärjestelmästä johdetuissa käyttöjärjestelmissä käytettävä pakettistandardi.
Shell	Komentorivitulkki, tai käyttöliittymä, jolla päästään käsiksi käyttöjärjestelmän matalammille tasoille. Shellissä toiminnot suoritetaan kirjoittamalla ne tekstinä ja painamalla enteriä
GNOME	GNU Network Object Model Environment. Työpöytäympäristö, joka toteuttaa Linux-käyttöjärjestelmille ikkunoinnin.
Emacs	Richard Stallmanin 70-luvulla aloittama tekstieditori, joka on lähtöisin Lisp-ympäristöistä, ja on käännetty lähes jokaiselle kuviteltavissa olevalle käyttöjärjestelmälle ja tietokoneelle.
Git	Hajautettu versionhallinta. Jos ohjelmisto ei löydy pakettienhallinnasta, se yleensä on ladattavissa gitillä.
JDK	Java Development Kit. Java-kielen kehitystyökalut, joita tässä työssä tarvitaan Identity Providerin rakennukseen.

IP-osoite IP-osoitteilla yksilöllistetään tietokoneet verkossa. Staattinen IP tarkoittaa osoitetta, jonka taataan pysyvän samana aina.

baseDN DN tulee sanoista distinguished name. DN-kyselyillä viitataan LDAP-puun tiettyihin solmuihin. BaseDN osoittaa puun juureen.

XPATH XPATH on XML-kyselykieli.

WAYF/DS Where Are You From - palvelulla selvitetään mihin federaation jäsenorganisaatioon käyttäjä kuuluu.

Socket Abstraktio, jonka avulla internet ja lähiverkot toimivat. Kirjoitettuasi sockettiin dataa, toisella puolen oleva kone saa kyseisen datan lukiessaan sockettia.

Välityspalvelin

Palvelin, joka näkee HTTP-pyynnön, mahdollisesti käsittelee sitä ja lähettää sen eteenpäin jollekin toiselle palvelimelle. Näin saadaan tärkeät palvelimet suojattua ulkomaailmalta, mutta silti päästetään ulkomaailma niihin käsiksi hyvin määriteltyjä polkuja pitkin.

OID Object identifier. OID on yleiskäyttöinen, kansainvälisesti vain yhteen kohteeseen liitettävä numerosarja, joka yksilöi kohteen yksiselitteisesti.

RPM-pakettilähde

RPM-pakettilähde on palvelin, josta RPM-paketteja käyttävät pakettienhallinnat hakevat pakettinsa.

ddns-osoite Dynamic DNS. DNS on protokolla, jonka avulla internetosoitteet, kuten http://google.com, muutetaan numeerisiksi ip-osoitteiksi, kuten 62.78.98.168. Dynamic DNS:n tapauksessa palvelinkoneelta puuttuu staattinen IP-osoite, joten aina saadessaan uuden se kertoo uuden osoitteensa dns-palvelimille.

1 JOHDANTO

Autentikaatio on monimutkainen prosessi, jossa useimmilla ei ole varaa tehdä virheitä. Jos autentikaation ohjelmoija ei ajattele kaikkia mahdollisia syötteitä, saattaa hänen ohjelmistonsa tulostaa tietokannasta kaiken datan. Tietyillä syötteillä autentikaatiokoodi saattaa poistaa kaiken tärkeän datan. Varmuuskopioiden tärkeys korostuu tilanteissa, joissa ohjelmoija kirjoittaa ensimmäistä kertaa autentikaatiokoodia.

Tässä työssä tutkitaan onko olemassa olevassa projektissa mahdollista käyttää jonkun muun kirjoittamaa autentikaatiokoodia. Kyamkin CyberSecurityGame - projektin muu toiminnallisuus on hyvällä mallilla, mutta kyseisestä palvelin-asiakas-arkkitehtuuria noudattavasta ohjelmistosta puuttuu hyvälaatuinen autentikaatiokerros.

Työssä ollaan kiinnostuneita myös siitä, saadaanko muiden korkeakoulujen opiskelijat kirjautumaan CyberSecurityGame-palveluun. Tämän suunnitellaan toimivan suomalaisen, CSC - Tieteen tietotekniikan keskus Oy:n ylläpitämän HAKA-federaation avulla. HAKA käyttää Shibboleth - palvelimia, joiden asetusten määrittely vaikuttaa työläältä. Shibbolethin kanssa tarvitaan http-palvelimia, joten työssä tutkitaan myös Apachen, Tomcatin ja yleisesti Linuxin asetusten määrittelyä.

2 CYBERSECURITYGAME

2.1 Pelin toimintaperiaate

CyberSecurityGame on työnimi Kymenlaakson ammattikorkeakoulun projektille, jolla on tarkoitus sekä markkinoida koulun kyberturvallisuuslaboratoriota että opettaa kyberturvallisuutta hyökkäysten ja puolustautumisen näkökulmasta. Korkean tason tiivistelmä pelin suunnitellusta toimintatavasta on seuraava: pelaaja ottaa yhteyden pelin palvelimeen valitsemallaan asiakaslaitteella. Palvelin luo häntä varten virtuaalikoneita erilaisilla Linux- ja Windows-käyttöjärjestelmäversioilla. Jokainen on uusi virtuaalikone on joko suojattu parhaan yleistietämyksen mukaan tai niihin on jätetty tarkoituksella tietty aukko. Virtuaalikoneen oletuskäyttäjän työpöydälle jätetään tekstitiedosto, ja kun käyttäjä kopioi tämän tekstitiedoston sisällön CyberSecurityGame - asiakasohjelmaan, pääsee hän etenemään pelissä.

Asiakasohjelmien on suunniteltu toimivan sekä puhelimessa että internetselaimessa. Peli-istunnot pidetään tiukasti palvelimella, ja ainoa mitä asiakasohjelma näkee, on käyttöliittymävisualisaatio pelin tilasta. Palvelin pitää huolta myös käyttöliittymäpuusta, ja lähettää puun muutokset TCP-yhteyden yli asiakkaalle.

Käyttöliittymäkirjastona toimii Marmalade. Marmalade sisältää kääntäjät iOS-, Android-, Windows Phone- ja Windows Metro -ympäristöille, sekä emulaattorin, joka matkii puhelimen toimintaa mahdollisimman tarkasti pelin ollessa päällä. Projektinhallinnasta ja tekstieditorista vastaa Microsoft Visual Studio 2015. Logiikka ohjelmoidaan C++ - kielellä. Verkkokoodi kirjoitetaan Raknet-kirjastolla.

Lisäksi projektissa on osia, jotka käyttävät PHP/Apachea, Redis-tietokantaa ja C#-kieltä. Osaa palvelimista on käytännöllisintä ajaa Windowsilla, osaa Linuxilla, mutta yleisesti koodi on mahdollisimman yhteensopivaa. Taustaohjelmistot ovat asennettavissa kummallekin ympäristölle, eikä puhelinasiakasohjelma käytä alustakohtaisia virityksiä.

2.2 Eettinen puoli

Eivätkö tietoturvahyökkäykset ole lain vastaisia?

”Joka aiheuttaakseen toiselle haittaa tai taloudellista vahinkoa dataa syöttämällä, siirtämällä, vahingoittamalla, muuttamalla tai poistamalla taikka muulla niihin rinnastettavalla tavalla oikeudettomasti estää tietojärjestelmän toiminnan tai aiheuttaa sille vakavaa häiriötä, on tuomittava tietojärjestelmän häirinnästä sakkoon tai vankeuteen enintään kahdeksi vuodeksi.” (Rikoslaki luku 38 pykälä 7, 2015).

Jos tarkoitus on tuottaa vahinkoa, on metodisi laiton. Hyökkääjän katsotaan syyllistyneen lain hengen mukaiseen haittaan, vaikka hyökkääjän tarkoitus olisi vain tutkia eikä tuottaa vahinkoa. Jos ylläpitäjän kanssa sovitaan miten ja milloin tietomurtotutkimukset toteutetaan, lain hengessä niistä ei rangaista.

”Tietomurrosta tuomitaan myös se, joka tietojärjestelmään tai sen osaan tunkeutumatta

1. *teknisen erikoislaitteen avulla tai*
2. *muuten teknisin keinoin turvajärjestelyn ohittaen, tietojärjestelmän haavoittuvuutta hyväksi käyttäen tai muuten ilmeisen vilpillisin keinoin*

oikeudettomasti ottaa selon 1 momentissa tarkoitetussa tietojärjestelmässä olevasta tiedosta tai datasta. Yritys on rangaistava. Tätä pykälää sovelletaan ainoastaan tekoon, josta ei ole muualla laissa säädetty ankarampaa tai yhtä ankaraa rangaistusta.” (Rikoslaki luku 38 pykälä 8, 2015.)

2.3 Ongelma ja ehdotettu ratkaisu

Opinnäytetyössä keskitytään autentikaatio-ongelman ratkaisuun. Kyamkin virtuaalisointiratkaisu on lisensoitu ilmaisena opiskelijakäyttöön, joten pelipalvelimen pitäisi automaattisesti hyväksyä pelaajat, joilla on käyttäjätili suomalaisessa korkeakoulussa, ja hylätä kaikki muut. Tämä onnistuisi helpoiten liittämällä palvelinohjelmisto suomalaiseen Haka-federaatioon.

”Haka on Suomen käytetyin korkeakoulujen ja tutkimuslaitosten käyttäjätunnistusjärjestelmä, jolla on noin 290 000 loppukäyttäjää. Haka-käyttäjätunnistus on myös väylä yli 160 palveluun. Hakan palveluihin kirjaudutaan yli 11 miljoonaa kertaa vuodessa.

Haka perustuu luottamusverkostoon. Verkoston jäsenet - tutkijat, opiskelijat ja muu henkilöstö - voivat käyttää kotiorganisaationsa käyttäjätunnuksia kirjautuessaan moniin eri palveluihin. Myös käyttäjien henkilötiedot siirtyvät turvallisesti palveluihin kirjauduttaessa. Haka on yhteensopiva muiden pohjoismaiden korkeakoulujen luottamusverkostojen kanssa, joten käytettävissäsi ovat myös pohjoismaiset palvelut.

Kotiorganisaation tietohallinto vastaa käyttäjiensä käyttäjätiedoista ja henkilöllisyyden todentamisesta. Hakassa olevien palvelujen käyttäjätiedot saadaan suoraan käyttäjän kotiorganisaatiosta.” (CSC 2014).

Haka perustuu Shibboleth-palvelimiin, joiden toiminnan ja konfiguroinnin opiskeluun meni todella paljon aikaa opinnäytetyötä tehdessä.

3 SHIBBOLETH

Shibboleth on Shibboleth Consortiumin kehittämä palvelinkokonaisuus. Tämä kokonaisuus tarjoaa käyttäjien- ja sessionhallintapalvelut HTTP:tä käyttäviin ohjelmistoihin. Shibbolethin federaatioiden avulla voit kirjautua samaan federaatioon kuuluvan naapuriorganisaation palveluihin oman organisaatiosi tunnuksilla. Seuraavaksi käydään läpi hieman pohjatietoa, jonka avulla Shibbolethin toiminta on helpompi ymmärtää.

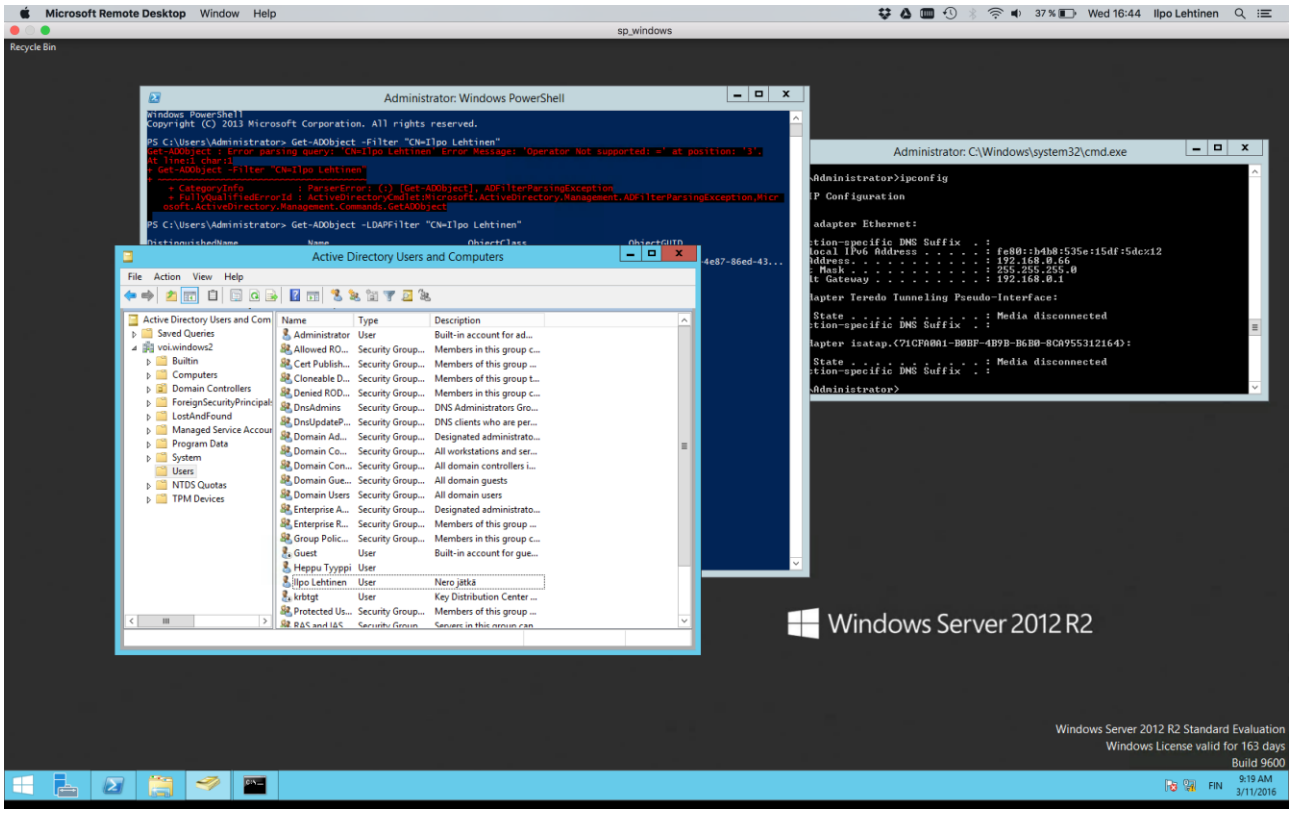
3.1 LDAP ja Active Directory

LDAP-tietokannat koostuvat hierarkiaan järjestetystä käyttäjädatasta. Microsoftin LDAP-tietokantatoteutuksessa, Active Directoryssä, yleisiä olioita ovat mm. maat, organisaatiot, ihmiset ja laitteet. Active Directory ei ole ainoa olemassa oleva LDAP-palvelin, esimerkiksi OpenLDAP tarjoaa samat palvelut ja toimii Windowsin ulkopuolella, mutta koska Active Directory on yleisempi ja OpenLDAP on vaikeampi konfiguroida ja käyttää, tässä työssä esitellään Active Directoryn toimintaa.

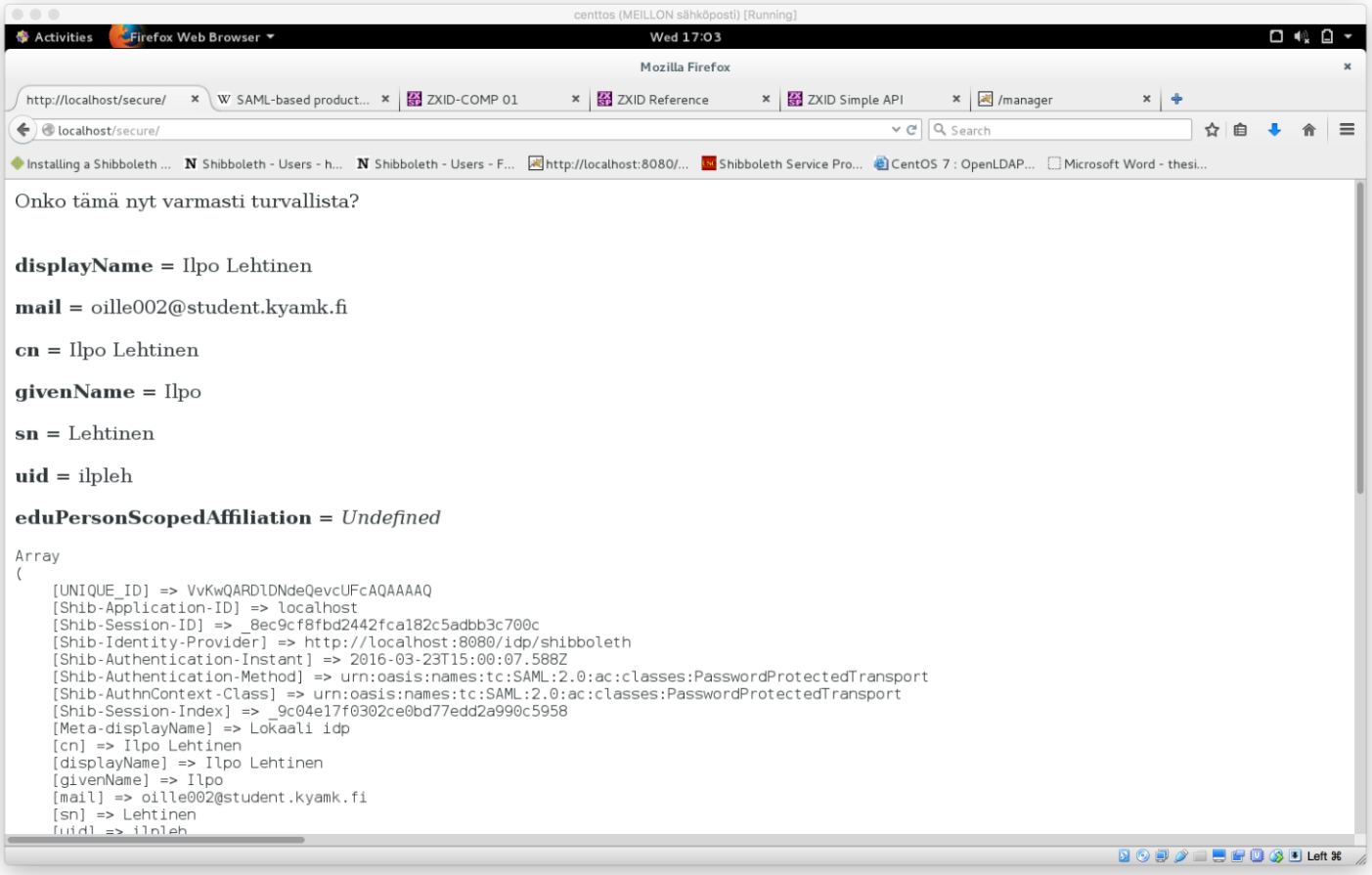
Active Directoryn palveluita käytetään tietoverkon käyttäjien ja palvelujen hallintaan. Active Directoryä vasten pystyy autentikoimaan LDAP-protokollaa puhuvalla asiakasohjelmalla. Käyttäjiä ei tarvitse lisätä käsin jokaiselle työasemalle, joita voi olla satoja, kun verkossa on AD-palvelin, vaan ne osaavat kysyä AD-palvelimelta, onnistuiko käyttäjän kirjautuminen annetuilla tunnuksilla. Tämä toimii muuallakin kuin työasemakirjautumisissa. Esimerkiksi

Kyamkin Moodle- ja SoleOPS-instanssit ovat ulkoistaneet autentikaation Active Directorylle.

LDAP-protokollassa olio on sarja avain-arvo -rivejä. Esimerkiksi seuraavassa kuvassa käyttäjän Ilpo Lehtinen LDAP-kuvaukseen kuuluvat seuraavat rivit.



Kuva 1 - Näkymä Active Directorystä

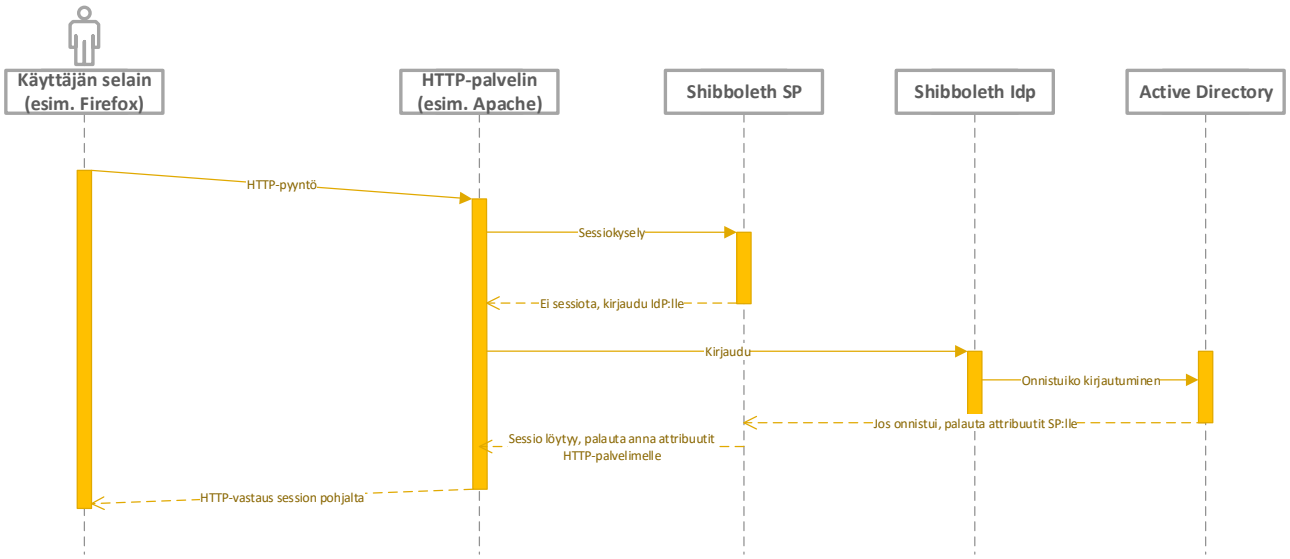


Kuva 2 - PHP-skripti, joka tulostaa Active Directoryltä Identity Providerin kautta tulevien attribuuttien arvot käyttäjältä Ilpo Lehtinen

3.2 IIS & Apache

Internet Information Server on Microsoftin HTTP-palvelin (Microsoft 2016). Apache on Apache Software Foundationin kehittämä avoimen lähdekoodin HTTP-palvelin (Apache Software Foundation 2016). Selain luo yhteyden palvelinkoneen porttiin 80, jota jompikumpi http-palvelimista kuuntelee. Palvelin vastaanottaa pyynnön, tarkistaa mitä osoitetta käyttäjä on selaimessaan hakenut, ja ohjaa pyynnön eteenpäin sen mukaan. Yleisimmät kohteet ovat PHP-skriptien tulkki, ASP(.NET)-skriptien tulkki, tai Apache Tomcat -java-palvelin. Shibboleth asennetaan laajennusosana näihin palvelimiin. Se liittyy prosessiin host-otsikon tulkkauksen ja lopullisen resurssin palauttamisen väliin, luoden valtuutetuille käyttäjille istunnon ja hyläten valtuuttamattomat.

Seuraavaksi yksinkertaistettu kaavio siitä, mitä tapahtuu, kun käyttäjä pyytää Shibbolethin suojaaman sivun.



Kuva 3 - Kaavio http-pyynnöstä Shibboleth - palvelinten läsnä ollessa

3.3 SAML

Security Assertion Markup Language on XML-pohjainen kehys käyttäjäautentikoinnin, käyttöoikeuksien ja attribuuttien välitykseen (Cover 2010). Viestit kulkevat Service Provider -palvelimen ja Identity Provider -palvelimen välillä. Erilaisten toimialueiden omistamien Identity Provider -

palvelinten muodostamaa kokonaisuutta, joissa palvelut toteuttavat yhteisen SSO:n, kutsutaan federaatioksi (Killeen 2012).

3.4 Shibboleth Identity Provider

Identity Provider on Javalla kirjoitettu ohjelmisto, joka asennetaan Tomcat-palvelimeen. Se vastaanottaa autentikaatiopyynnöt ja luo valtuutukset konfiguroidun LDAP-palvelimen antaman datan perusteella. Jos autentikaatio onnistui, valtuutuksen mukana lähetetään konfiguraation mukaisesti joukko attribuutteja, jotka voivat tulla LDAP-palvelimelta, ulkoisesta JDBC:tä tukevasta SQL-tietokannasta, tai jotka voidaan generoida Identity Providerin sisällä Javascriptillä.

3.5 Shibboleth Service Provider

Service Providerilla tarkoitetaan web-palveluntarjoajaa, yleensä HTTP-palvelinta. Service Provider delegoi autentikoinnin ja valtuutuksen luotetuille Identity Provider -koneille. SAML-mallissa Service Provider luottaa täysin siihen mitä federaatio sille kertoo. (empoweredID 2016).

Shibboleth Service Provider on Apacheen tai IIS:iin asennettava moduuli, joka toteuttaa SSO:n (Shibboleth Consortium 2015). Kun käyttäjältä tulee HTTP-pyyntö Service Providerin hallinnoimaan polkuun, Service Provider tarkistaa onko käyttäjällä voimassaolevaa istuntoa. Istunnon ollessa olemassa Service Provider hakee konfiguroidut käyttäjään liittyvät attribuutit Identity Providerilta. Se liittää attribuutit HTTP-pyyntöön ja antaa lopullisen, pyydetyn skriptin, oli se sitten PHP:tä, ASPia tai jotain muuta, tehdä niillä mitä haluaa. Jos istuntoa ei ole, Service Provider palauttaa HTTP-uudelleenohjauksen (koodi 302) Identity Providerin kirjautumisosoitteeseen. Identity Provider autentikoi käyttäjän, uudelleenohjaa takaisin osoitteeseen johon käyttäjä oli menossa, ja kun Service Provider pyytää attribuutteja, luovuttaa ne.

Service Providerin voi konfiguroida suodattamaan pyyntöjä täysin attribuuttien perusteella, tiputtamaan attribuutteja pyynnöstä pois jos ne täyttävät jonkin ehdon, ja muuttamaan niitä. Nämä muunnokset ohjelmoidaan XML-kielellä jota ei tässä työssä käsitellä.

4 LABORATORIOYMPÄRISTÖN RAKENTAMINEN

Seuraavaksi rakennettiin laboratorioympäristö, jonka tehtävä oli opettaa perusperiaatteet Shibbolethista ja federaatioista käytännössä. Tästä saatuja kokemuksia suunniteltiin käytettäväksi lopullisen, testihakaan liitettävän virtuaalikoneen rakentamisen yhteydessä.

4.1 CentOS:n ja Service Providerin asennus

Shibboleth Service Provider on ladattavissa Windows Installer- ja RPM-paketeissa. Shibboleth-palvelimet asennettiin virtuaalikoneelle CentOS7-käyttöjärjestelmään, koska siellä pakettien hallinta ja asetustiedostojen tekstipohjainen määrittely on helpompaa. Uusin Centos7 Minimal on ladattavissa osoitteesta http://centos.org. Kun käyttöjärjestelmä oli asennettu, huomattiin ettei minimaaliversiossa ole käyttöliittymää: se käynnistyi shelliin. Graafinen työpöytäympäristö ja nettiselain tarvittiin testaamiseen, joten seuraavat ajettiin pääkäyttäjän terminaalissa

```
yum -y groups install "GNOME Desktop"
yum -y install emacs git java-1.8.0-openjdk.x86_64
systemctl enable gdm.service
systemctl set-default graphical.target
systemctl start gdm.service
```

Nämä komennot asensivat GNOME-työpöydän riippuvuuksineen, asettivat käyttöjärjestelmän oletuksena käynnistymään graafiseen tilaan ja käynnistivät GNOME-työpöydän. Rivi 2 asentaa myös emacsin, gitin ja Java Development Kitin, joita tarvitaan Identity Providerin asentamiseen ja molempien asetusten määrittelyyn. GNOMEen kirjautumisen jälkeen tarvitsimme loput työkalut.

```
yum -y install firefox httpd php tomcat.noarch tomcat-admin-webapps.noarch #asennetaan selain, http-palvelin Service Providerille ja java-palvelin Identity Providerille
```

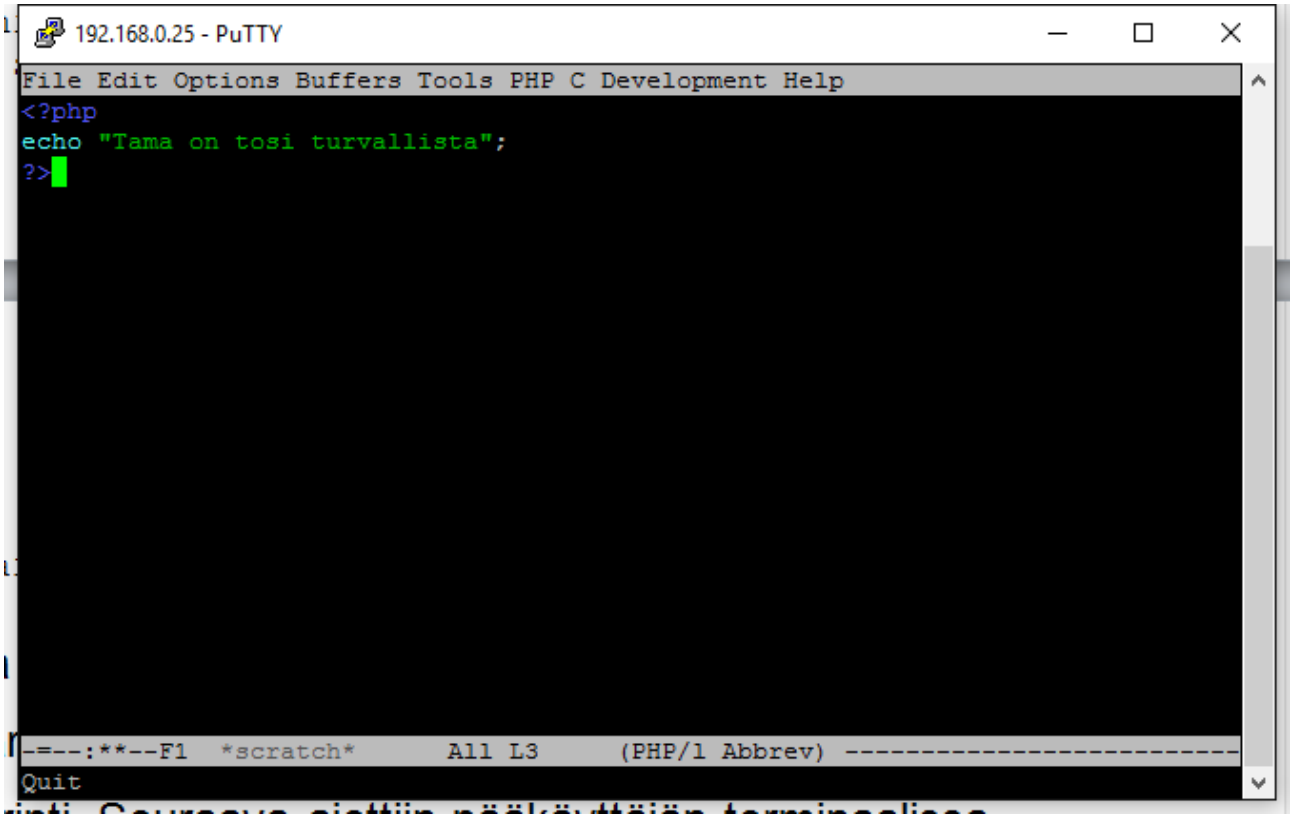
Pakettienhallintaan oli lisättävä merkintä Shibboleth Consortiumin pakettilähteestä jotta Service Provider oli mahdollista asentaa. Seuraava koodi ajettiin pääkäyttäjän terminaalissa:

```
curl -o /etc/yum.repos.d/security:shibboleth.repo
http://download.opensuse.org/repositories/security://shibboleth/CentOS_7/security:shibboleth.repo
yum -y install shibboleth.x86_64
systemctl restart httpd shibd
```

Oletusasetuksilla Service Provider suojaa osoitetta http://localhost/secure. Laboratorioympäristössä ei ollut oikeaa php-ohjelmaa suojattavaksi. Tarvittiin testiskripti. Seuraava ajettiin pääkäyttäjän terminaalissa

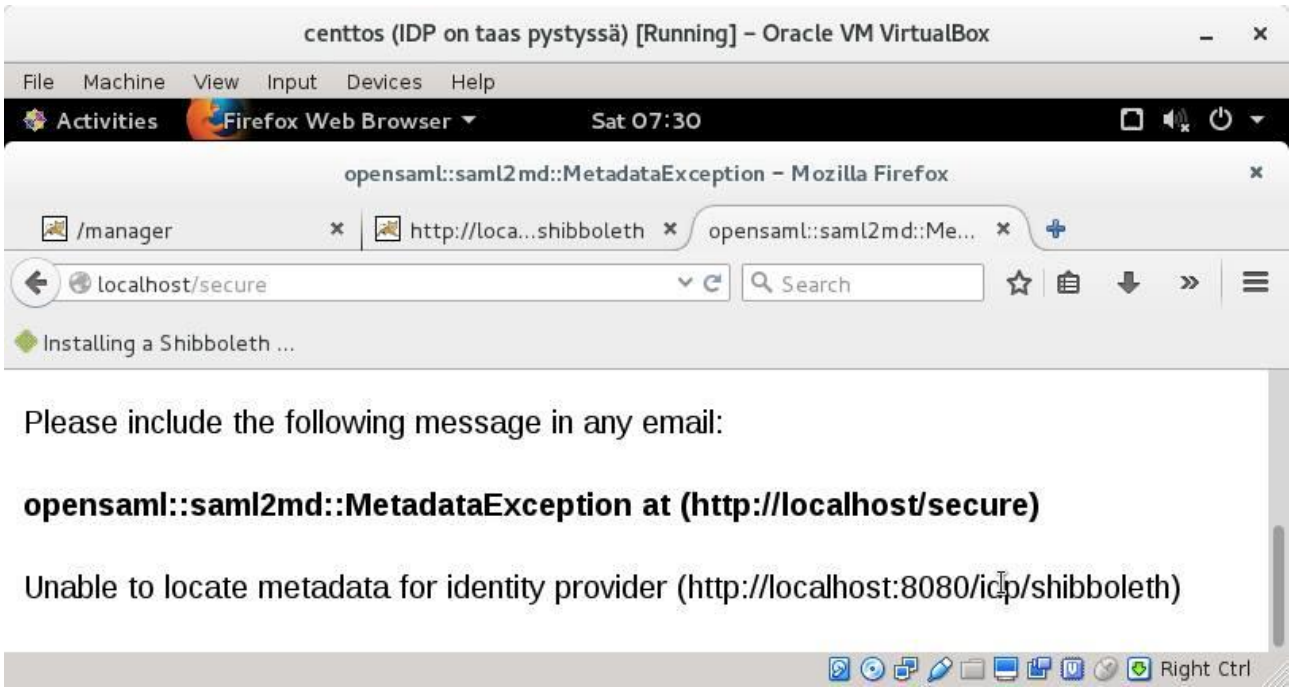
```
mkdir --parents /var/www/html/secure
emacs /var/www/html/index.php
```

Tähän tiedostoon kirjoitettiin seuraava skripti



Kuva 4 - Testiskripti

Tämän jälkeen kun selaimella siirryttiin osoitteeseen http://localhost/secure, tuli Shibbolethin virhesivu näkyviin:



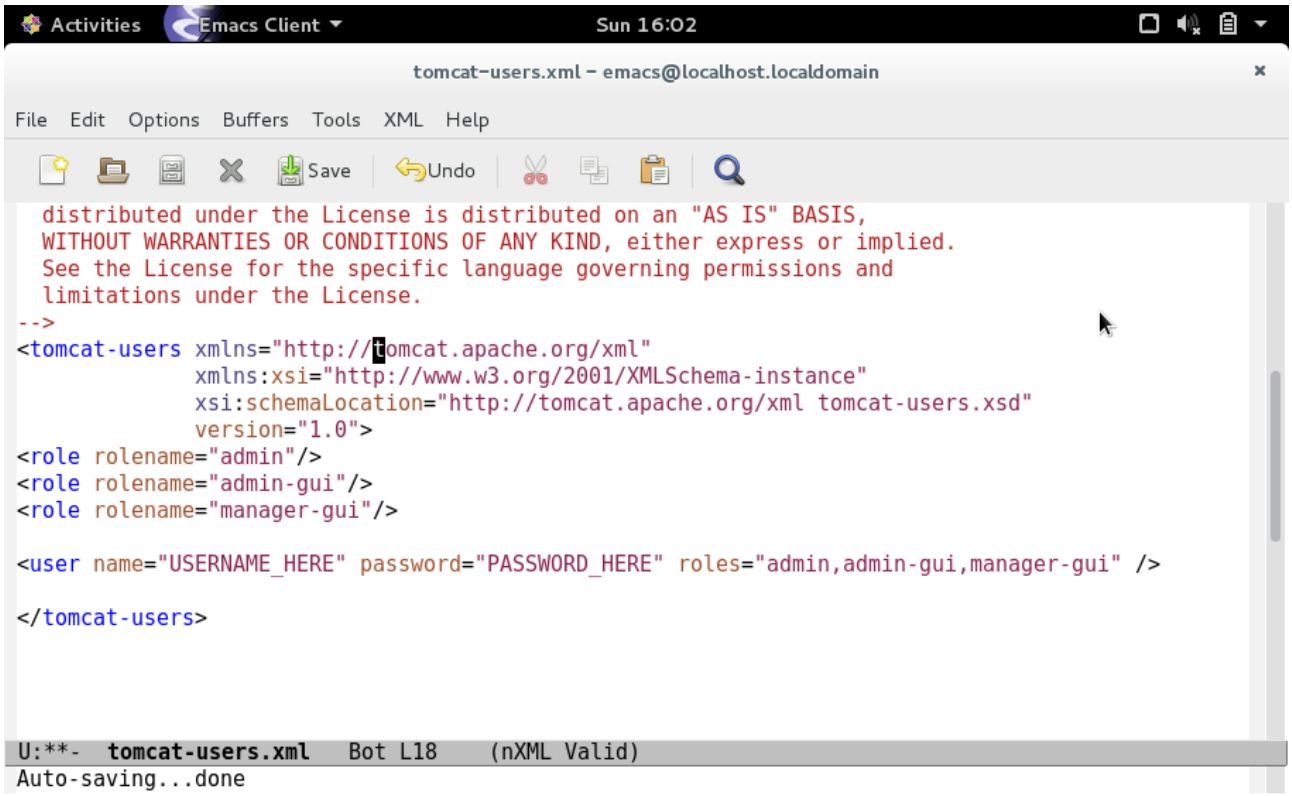
Kuva 5 - Tekstit eivät ole samat, mutta tästä kuvasta näkee SP:n virhesivujen tyylin

4.2 Identity Providerin asennus

Seuraavat rivit ajettiin pääkäyttäjän terminaalissa.

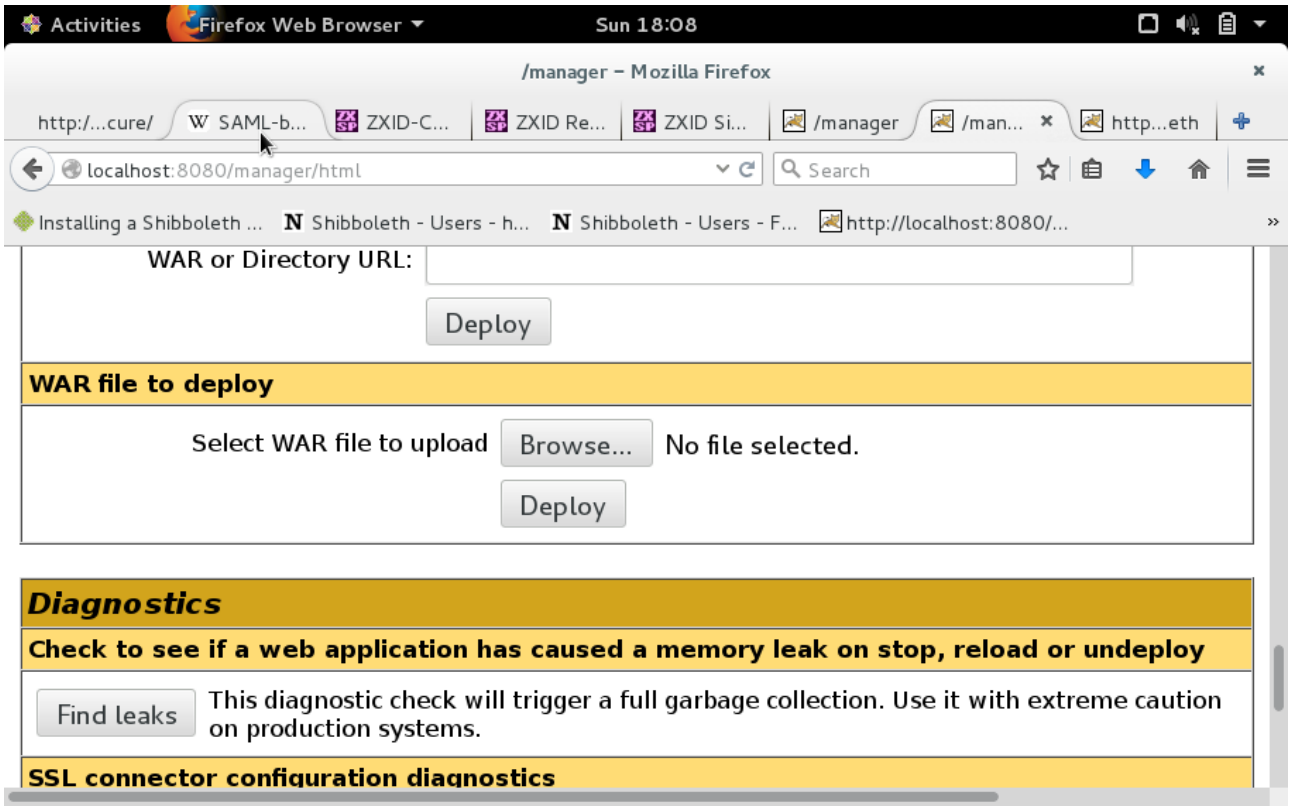
```
cd ~
curl -o idp.zip http://shibboleth.net/downloads/identity-provider/latest/shibboleth-identity-provider-3.2.1.zip
unzip idp.zip
cp -r idp /opt/idp
cd /opt/idp
sh ./bin/build.sh
```

Skripti kysyy kysymyksiä Identity Providerin suunnitellusta sijainnista ja sertifikaateista. Tässä työssä emme laboratoriovaiheessa tarvitse salausta, joten kaikki sertifikaattikentät jäivät tyhjiksi. Skripti build.sh loi /opt/idp/war -kansioon idp.war-paketin, joka sisälsi Identity Providerin suoritettavan ohjelmakoodin. Seuraavaksi se piti asentaa tomcat-palvelimeen, mikä tarkoittaa että tarvittiin pääsy tomcat manageriin. Tiedosto /opt/tomcat/conf/tomcat-users.xml avattiin tekstieditoriin. Asetustiedostoon määriteltiin pääkäyttäjä seuraavan kuvan esimerkin mukaan:

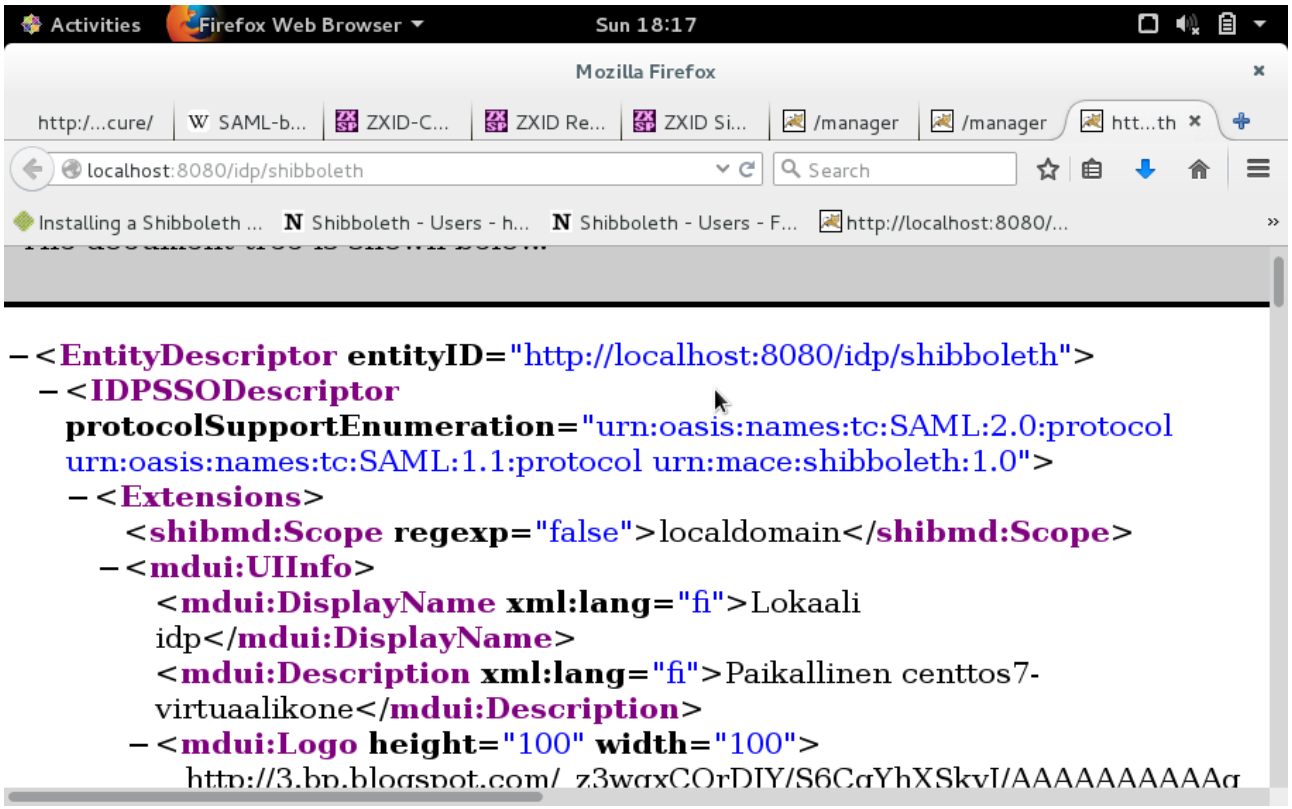


Kuva 6 - Roolit ovat tärkeitä. Nämä asetukset mahdollistavat käyttäjän kirjautumisen manageriin tunnuksilla USERNAME_HERE/PASSWORD_HERE. roles-attribuutin arvo on tärkeä

Tomcat-palvelu käynnistettiin uudelleen. Selain avattiin osoitteeseen http://localhost:8080, manager-nappia klikattiin ja juuri asetetulla tunnusparilla kirjauduttiin. "WAR file to deploy"-kenttään annettiin edellä rakennettu idp.war-paketti, deploy-nappulaa painettiin ja osoitteesta http://localhost:8080/idp/shibboleth tarkastettiin että palvelin vastaa pyyntöihin.



Kuva 7 - Tomcat manager. Tätä kautta asennetaan javalla kirjoitetut palvelinohjelmistot, jotka tunnistaa .war - tiedostopäätteestä



Kuva 8 - Jos Identity Provider vastaa /idp/shibboleth - polusta omalla metadatallaan, on se asentunut

4.3 Identity Providerin asetusten määrittely

Työssä käytettiin LDAP-lähteenä Windows 2008R2 -virtuaalikonetta, joka oli määritelty ylläpitämään omaa toimialuettaan. Virtuaalikoneen staattinen IP oli 192.168.0.66, baseDN oli voi.Windows2, toimialueen lyhennelmä VOI2. Työryhmään asennettiin kaksi käyttäjää, VOI2\Administrator ja VOI2\ilpleh, joista molemmat olivat Domain Administrators - ylläpitäjäryhmässä. Salaamattomien LDAP-palvelinten portti on yleensä 389, ja Active Directory noudattaa tätä tapaa. Windowsin ja Active Directoryn asennus ohitetaan tässä dokumentissa.

Jotta Identity Provider - palvelin saatiin käyttämään Active Directory - konetta osoitteessa 192.168.0.66 käyttäjädatan lähteenä, tehtiin tiedostoon /opt/idp/conf/ldap.properties seuraavat muutokset:

```
idp.authn.LDAP.authenticator            = directAuthenticator

## Connection properties ##
idp.authn.LDAP.ldapURL                  = ldap://192.168.0.66:389
idp.authn.LDAP.useStartTLS              = false
idp.authn.LDAP.useSSL                   = false
#idp.authn.LDAP.connectTimeout              = 3000

#idp.authn.LDAP.sslConfig                = certificateTrust

# If you require other user attributes in your app, they must be set the first time here
idp.authn.LDAP.returnAttributes         = cn,sn,givenName,displayName,uid,mail

idp.authn.LDAP.baseDN                   = DC=voi,DC=Windows2
idp.authn.LDAP.userFilter               = (sAMAccountName={0})
# bind search configuration
# for AD: idp.authn.LDAP.bindDN=adminuser@domain.com

idp.authn.LDAP.bindDN                   = ilpleh@voi.Windows2
# cn=Administrator,cn=Users,dc=voi,dc=Windows2
idp.authn.LDAP.bindDNCredential         = {PASSWORD HERE}
idp.authn.LDAP.subtreeSearch                                = true

# Format DN resolution, used by directAuthenticator, adAuthenticator
# for AD use idp.authn.LDAP.dnFormat=%s@domain.com
idp.authn.LDAP.dnFormat                 = %s@voi.Windows2

idp.attribute.resolver.LDAP.returnAttributes    = *
```

Asetuksen idp.authn.LDAP.dnFormat yllä olevassa kommentissa näkyy, että Active Directoryn kanssa on käytettävä uid@base.dn - muotoa DN-asetuksissa. Tässä asetuksessa %s korvataan kirjautumisen yhteydessä Identity Providerin kirjautumislomakkeella käyttäjän antamalla käyttäjänimellä.

Asetus idp.authn.LDAP.subtreeSearch mahdollistaa sen, että Identity Providerin voi asettaa kirjautumaan LDAP-hakemiston juuritasolle, ja se osaa hakea puun alioksista käyttäjäoliota.

Asetuksessa idp.attribute.resolver.LDAP.returnAttributes määritellään että Active Directorystä palautetaan kaikki olioon liittyvät attribuutit. Tämä määriteltiin kertaalleen myös asetuksessa idp.authn.LDAP.returnAttributes.

LDAP-liitos on määritelty. Seuraavaksi nämä LDAP-asetukset otettiin käyttöön attribute-resolver.xml - tiedostossa, joka sijaitsi samassa kansiossa ldap.propertiesin kanssa. Sinne syötettiin juurielementin alle seuraava DataConnector-elementti:

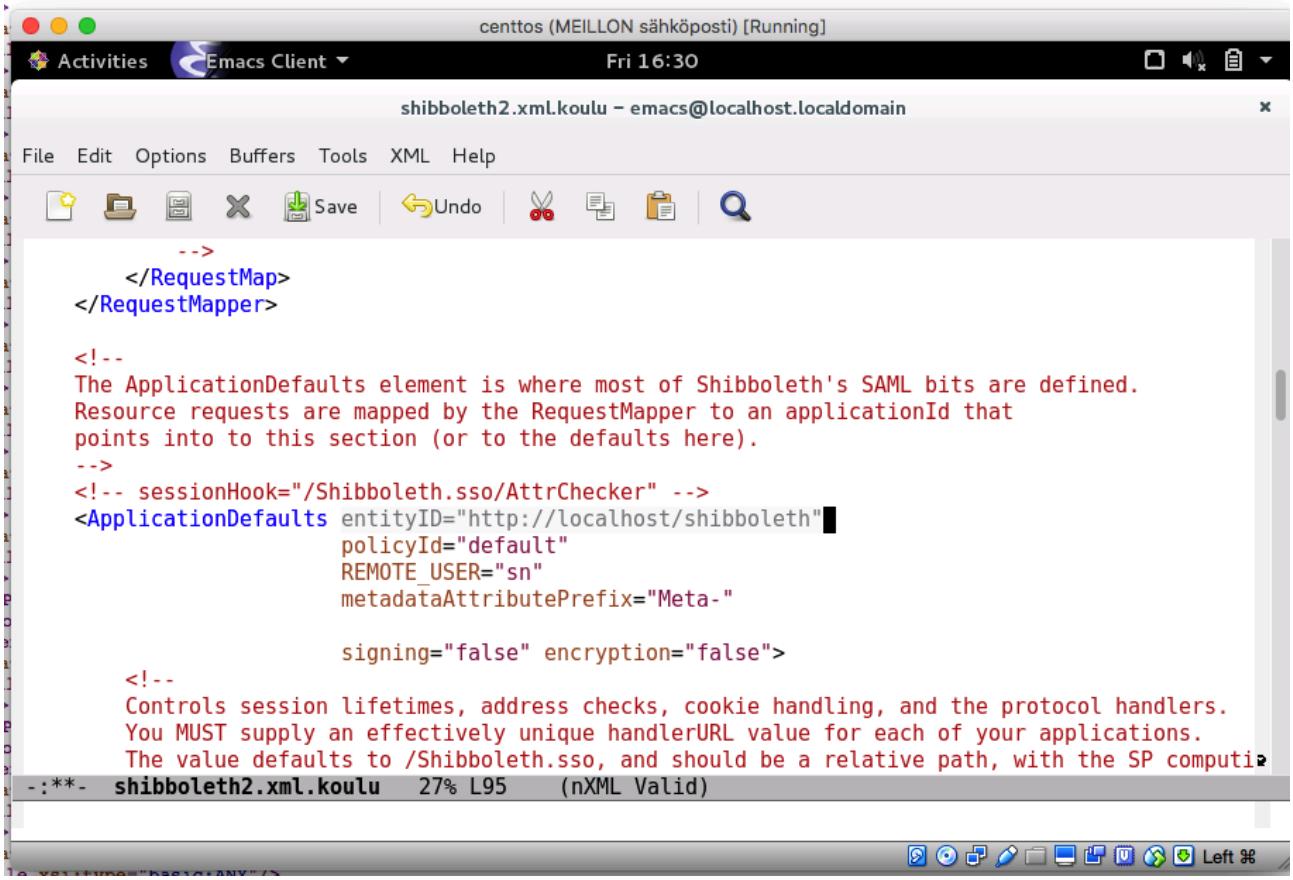
```
<resolver:DataConnector id="myLDAP" xsi:type="dc:LDAPDirectory"
    ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
    baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
    principal="%{idp.attribute.resolver.LDAP.bindDN}"
    principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
    useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:false}">
    <dc:FilterTemplate>
       <![CDATA[
          %{idp.attribute.resolver.LDAP.searchFilter}
       ]]>
    </dc:FilterTemplate>

<dc:ReturnAttributes>%{idp.attribute.resolver.LDAP.returnAttributes}</dc:ReturnAttributes
>

  </resolver:DataConnector>
```

Tiedostoon attribute-resolver.xml piti määritellä nimi-oid - yhteydet Active Directoryn attribuuteille. Katso kokonainen tiedosto liitteestä 1.

Tiedostossa attribute-filter.xml määriteltiin Service Providerille entityID:llä ”http://localhost/shibboleth” pyytävälle koneelle vapautettavaksi kaikki attribuutit jotka Active Directoryltä saadaan. Jos Identity Provider kuuluisi federaatioon, käytettäisiin kyseisen federaation ylläpitäjän rakentamaa attribute-filter.xml:ää koska laboratorioympäristö on vain 2 koneen kokoinen. Käytämme liitteen 2 mukaista attribute-filter.xml:ää.

4.4 Service Providerin asetusten määrittely

Seuraavat muutokset tehtiin tiedostoon /etc/shibboleth/shibboleth2.xml. Jokainen ”entityID=”https://sp.example.org/shibboleth”” korvattiin merkkijonolla ”entityID=”http://localhost/shibboleth”.



Kuva 9 - Uusi EntityID

XPath-polussa /SPConfig/RequestMapper/RequestMap/Host määritellään Shibbolethin suojaamat kansiot. Jos haluttaisiin suojata muuta kuin http://localhost/secure - kansion, ne tulisi lisätä tähän <Path> - elementteinä. Koska tässä vaiheessa työtä pelkkä onnistunut kirjautuminen riittää, jätetään tämä elementti rauhaan.

Xpath-polussa /SPConfig/ApplicationDefaults attribuutti REMOTE_USER määrittelee mitä attribuutteja Identity Provider - palvelimelta on aina saatava. Jos tässä attribuutissa pyydetyt attribuutit puuttuvat, käyttäjä ei voi autentikoida Service Providerin suojaamaan palveluun. Tässä työssä attribuutin arvo on REMOTE_USER=”uid”, koska Active Directory - koneelta tulevaa uidia on suunniteltu käytettävän käyttäjänimenä.

Rivit, joilla mainitaan http-polku /Shibboleth.sso/AttrChecker poistettiin. Kyseistä polkua ei käytetä rpm-pakettilähteestä tulevassa oletus-shibboleth2.xml:ssä, mutta koska työn /etc/shibboleth2.xml - tiedosto pohjautui Kyamkin vanhassa tuotantoympäristössä olleeseen asetustiedostoon, siellä oli merkintöjä kyseiseen polkuun. AttrChecker

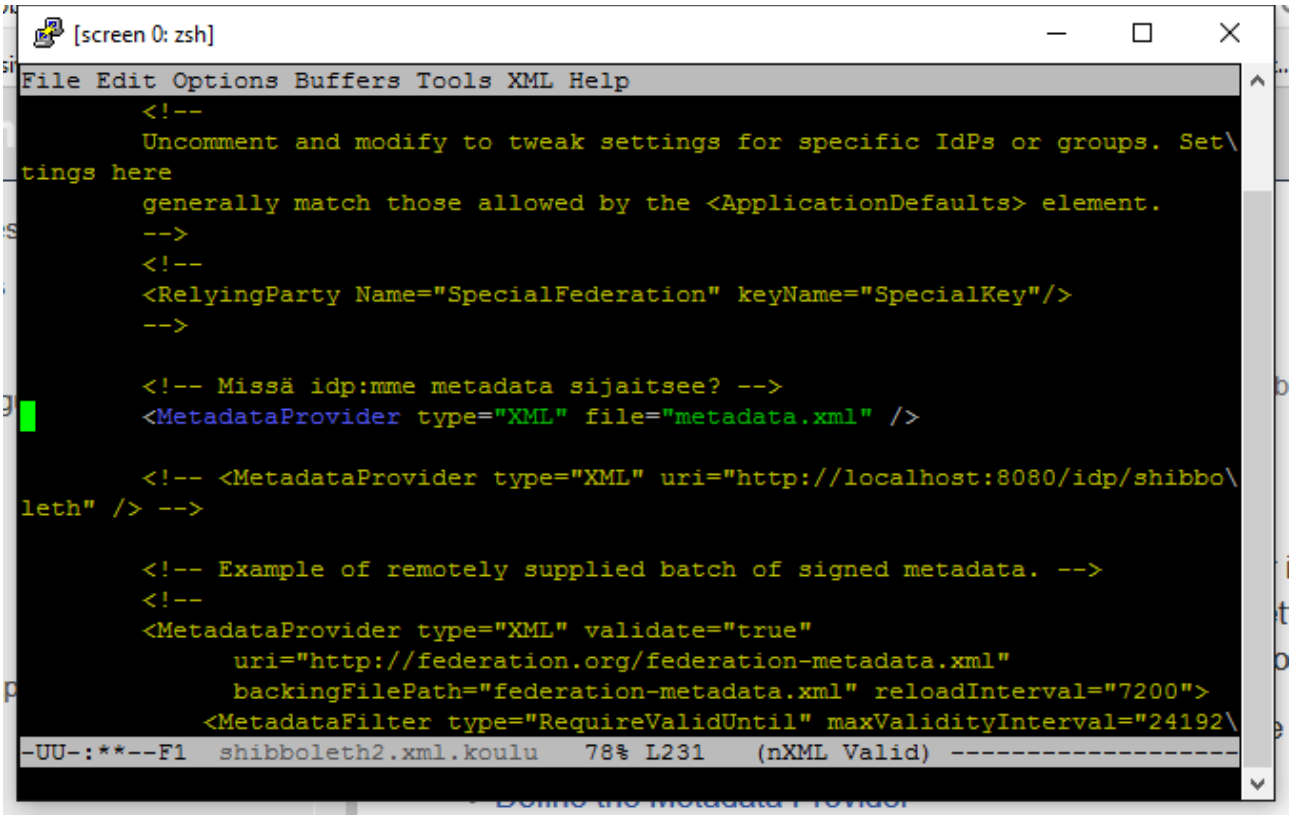
varmistaa että pyynnössä on vaadittavia attribuutteja, ja se on laboratorioympäristössämme aivan turha.

XPath-polussa /SPConfig/ApplicationDefaults/Errors löytyy ylläpitäjien yhteystiedot. Ne jätettiin ennalleen, koska työn laboratoriovaihetta ei tullut muu kuin ylläpitäjä näkemään. Lopullisessa Service Provider asennuksessa tähän tultaisiin laittamaan saman henkilön sähköpostiosoite, joka on annettu CSC:lle palvelua haka-federaatioon rekisteröidessä.

Seuraavaksi Service Providerille pitää kertoa missä Identity Provider sijaitsee. Identity Provider julkaisee http-polussa /idp/shibboleth xml-tiedoston, jonka Service Provider osaa lukea. Tiedosto kannattaa ladata palvelimen tiedostojärjestelmään. Näin tehtiin ajamalla seuraava pääkäyttäjän terminaalissa:

```
cd /etc/shibboleth
curl -o metadata.xml http://localhost:8080/idp/shibboleth
```

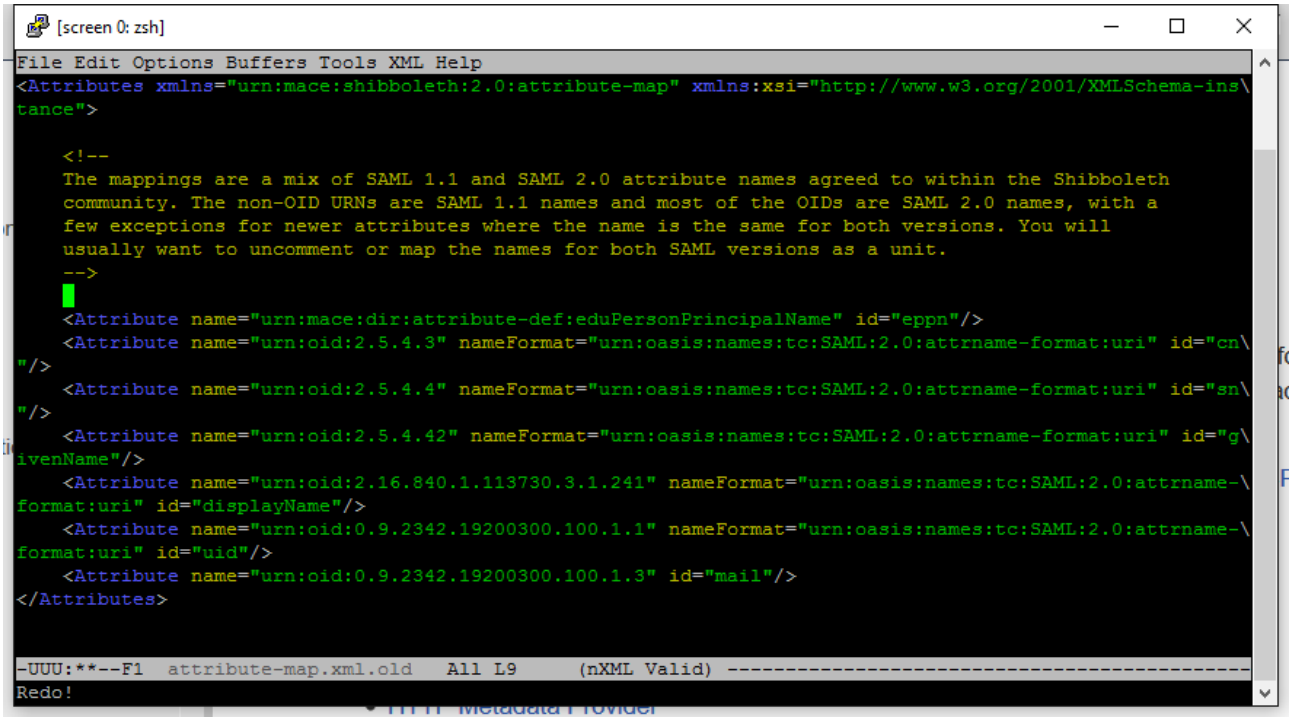
jonka jälkeen shibboleth2.xml - tiedostoon lisättiin seuraava rivi



Kuva 10 - MetadataProvider lisätty Shibbolethin asetustiedostoon

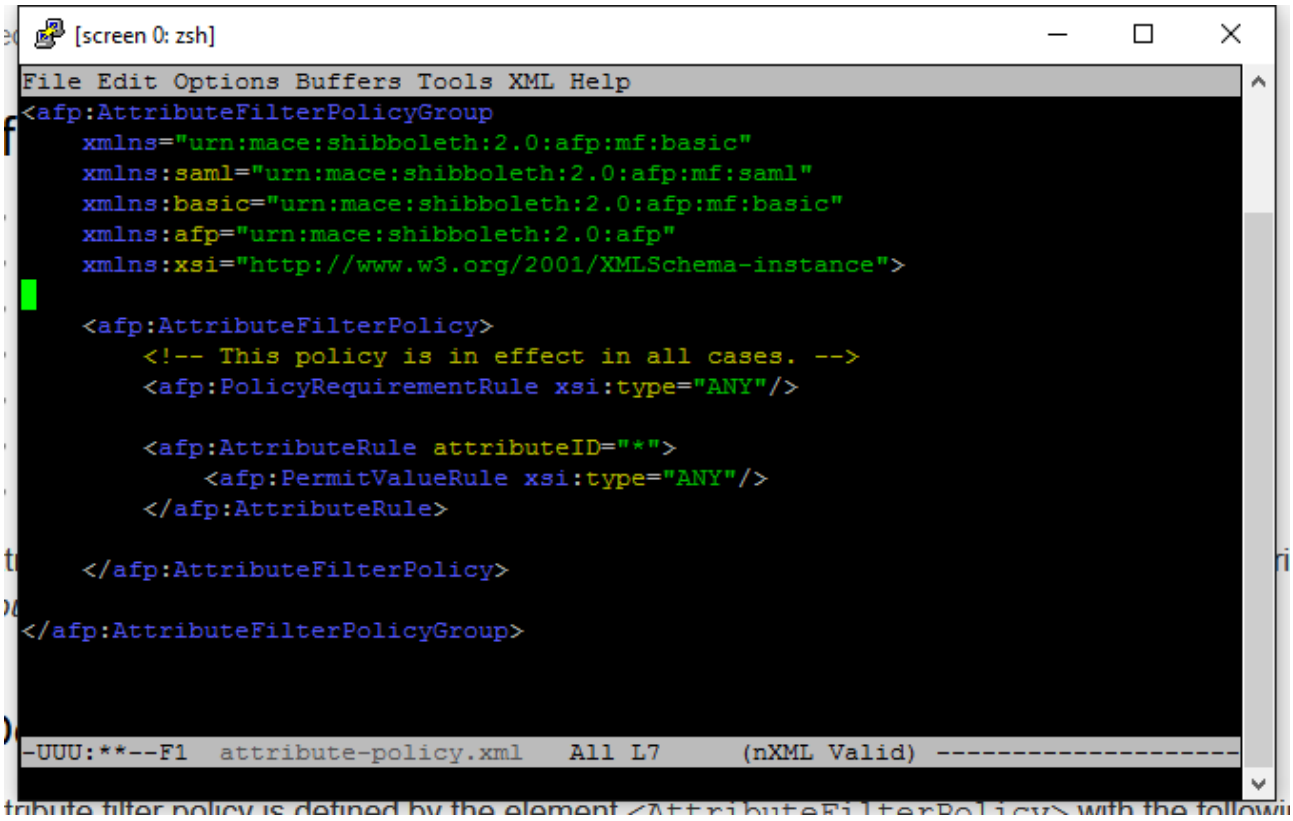
Service Providerille oli kerrottava minkä nimisiin attribuutteihin mikäkin SAML-viesteissä kulkeva OID-symboli liitetään. Se tehtiin /etc/shibboleth/attribute-

map.xml - tiedostossa. Seuraavassa on lisätty asetukset AD-attribuuteille cn, sn, givenName, displayName, uid ja email. Lisäksi tiedostossa määriteltiin eppn - attribuutti, joka työssä jäi turhaksi.



Kuva 11 - attribute-map.xml

Viimeisenä silauksena /etc/shibboleth/attribute-policy.xml piti asettaa hyväksymään kaikki attribuuttijoukot. Attribute-policy.xml voisi hyväksyä ja hylätä kirjautumisia tarkistamalla kirjautumisyrityksen attribuutit tämän tiedoston ehtoja vasten, mutta xml-pohjaisena kielenä ehtokieli on hyvin vaikeaselkoista ja monimutkaista.



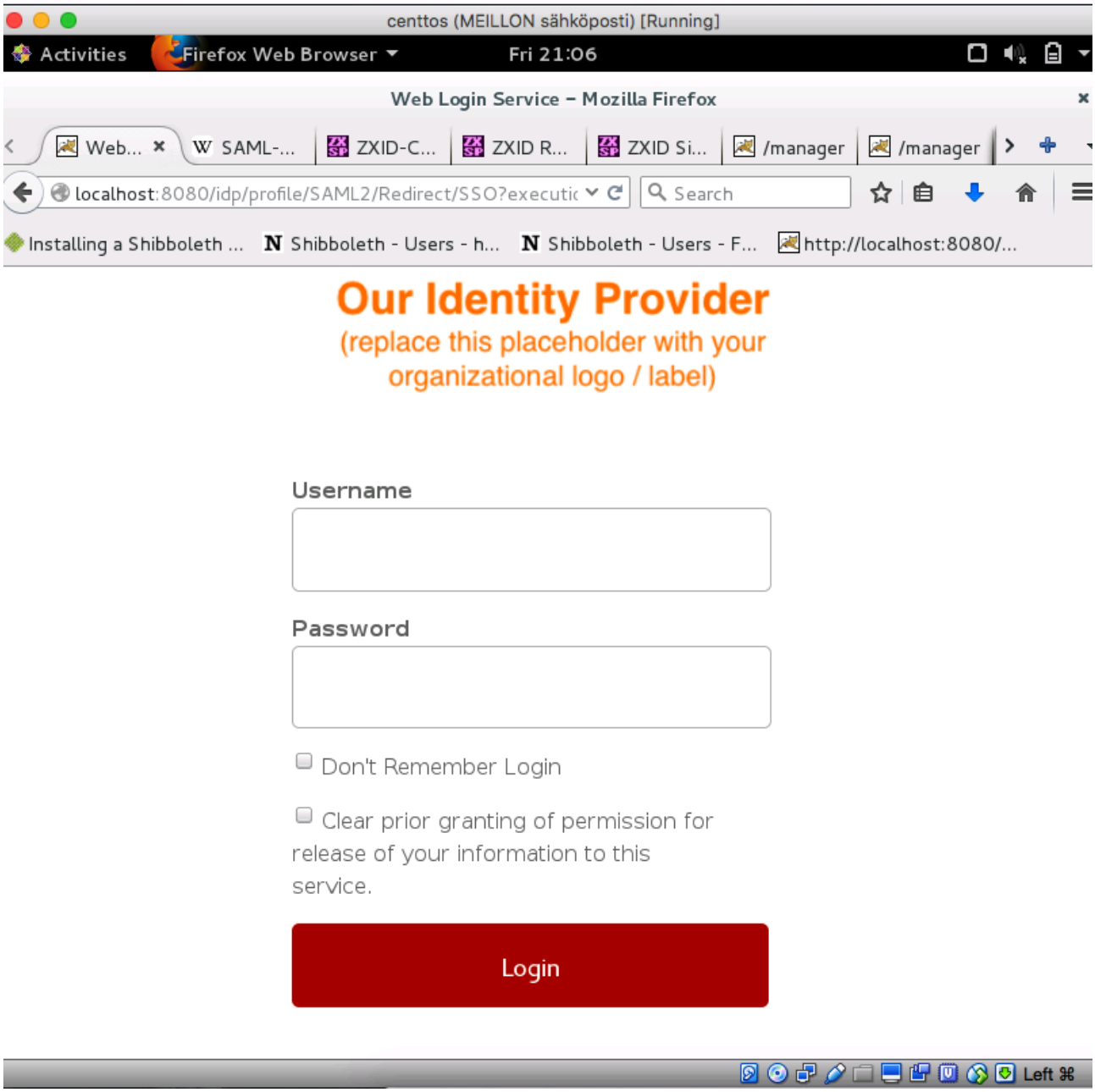
Kuva 12 - attribute-policy.xml joka hyväksyy kaiken

4.5 Toimiva ympäristö

Seuraavaksi käynnistettiin pääkäyttäjän terminaalissa kaikki oleelliset palvelut uudelleen:

```
systemctl restart shibd httpd tomcat
```

ja avattiin selaimessa suojattu sivusto http://localhost/secure, ensimmäisenä tuli näkyviin Identity Providerin kirjautumissivu.



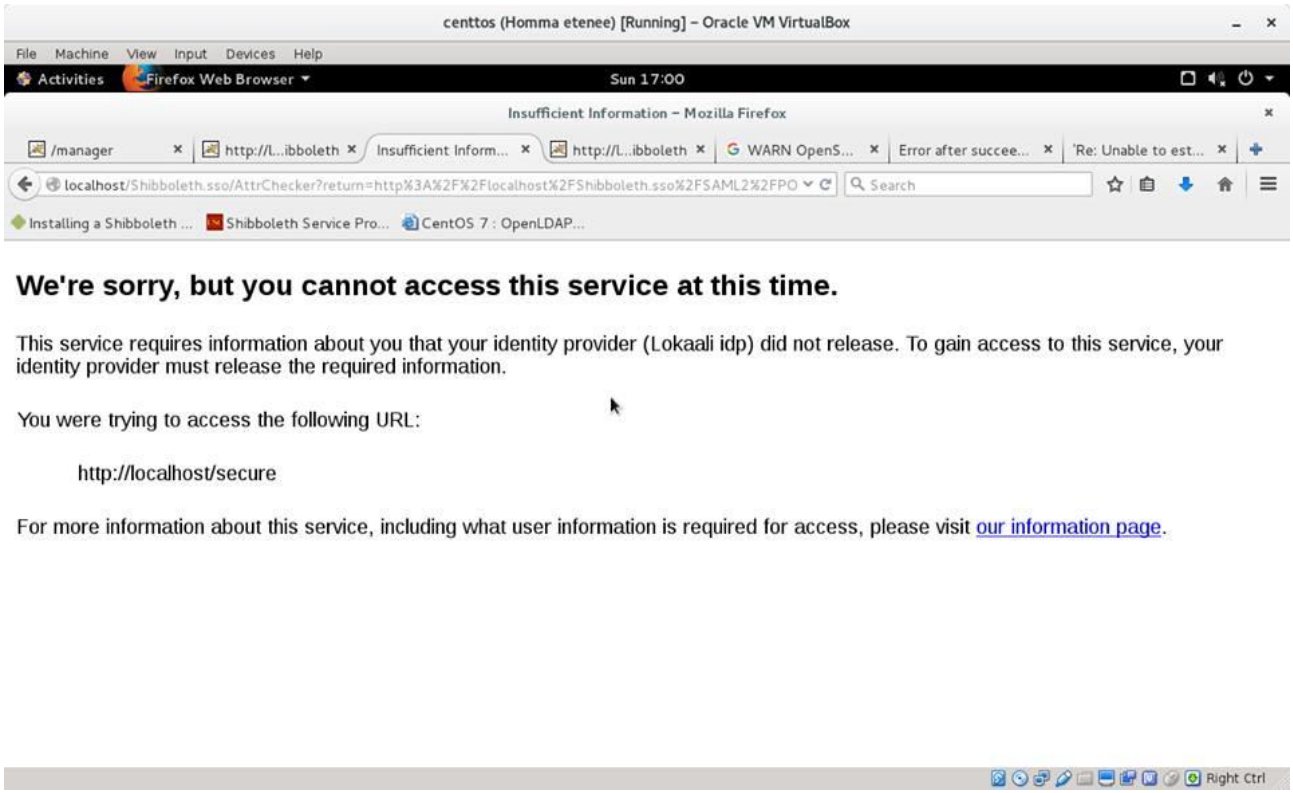
Kuva 13 - Identity Providerin oletuskirjautumissivu

Tämä sivun sisältö löytyi laboratorioympäristössämme polusta /opt/shibboleth-idp/views/login.vm. Kyseessä oli Apache Velocity -kuvauskielellä toteutettu käyttöliittymä. Koska työssä ei ollut oleellista tehdä kustomoitua ympäristöä, jätettiin näkymä oletuksen mukaiseksi.

4.6 Virhetilanteet

Virhetilanteilta ei voi välttyä. Ensimmäinen merkki virheestä oli yleensä kun selaimella yritti päästä käsiksi suojattuun polkuun, ja Shibboleth tulostaa ennen - tai viimeistään välittömästi jälkeen - kirjautumista virheviestin kuvan 3 tyylejä käyttäen. Joskus virheviesti voi olla ihan selkeä, kuten kuvassa kolme (Identity Provideriin ei saada yhteyttä? Tulostakaamme käyttäjälle virheviesti

”Identity Provideriin ei saada yhteyttä”), mutta joskus virheviesti kertoi ”Identity Provider ei vapauttanut tämän Service Providerin vaatimia attribuutteja" huolimatta siitä että Identity Provider oli asetettu vapauttamaan kaikki, kuten seuraavassa kuvassa.



Kuva 14 - Identity Provider joka vapauttaa kaikki attribuutit ei vapauta tarpeeksi attribuutteja

Kuvan virhetilanne johtui siitä että Service Providerissa on 2-3 paikkaa joissa voi konfiguroida vaadittavat attribuutit, joista viimeistä ei oltu huomattu.

Tällaisten virhetilanteiden selvittäminen ei ole helppoa. Ensimmäinen vaihe oli ajaa, mielellään pääkäyttäjän terminaalissa, komento ’shibd -t’. Se tarkistaa Service Providerin asetustiedoston ja kertoo jos löytää virheitä. Se on hyvin tarkka, joten se saattaa tulostaa huomautuksia joille ei tarvitse tehdä mitään. Esimerkiksi laboratorioympäristöstä puuttuva SSL-salaus, joka tuotantoympäristössä olisi paha moka, tuotti huomautuksia, joille ei ilman validia sertifikaattia voinut tehdä mitään.

Jos selaimen virheviesti ei ole järkevä eikä ’shibd -t’ huomaa mitään virheen arvoista, saattoi virhe löytyä joko Service Providerin lokista, jonka sijainti oli /var/log/shibboleth/shibd.log, tai Identity Providerin lokista polusta /opt/shibboleth-idp/logs/idp-process.log.

Työssä lokitasoja piti määritellä korkeammalle. Oletusasetuksin SAML-viestejä ei kirjoitettu mihinkään, mikä teki niiden sisällön tarkkailusta vaikeaa. Tämä ongelma korjattiin poistamalla risuaidat, jotka estävät asetuksen käyttöönoton, tiedoston /etc/shibboleth/shibd.logger riveiltä #log4j.category.OpenSAML.MessageDecoder=DEBUG ja #log4j.category.Shibboleth.RequestMapper=DEBUG.

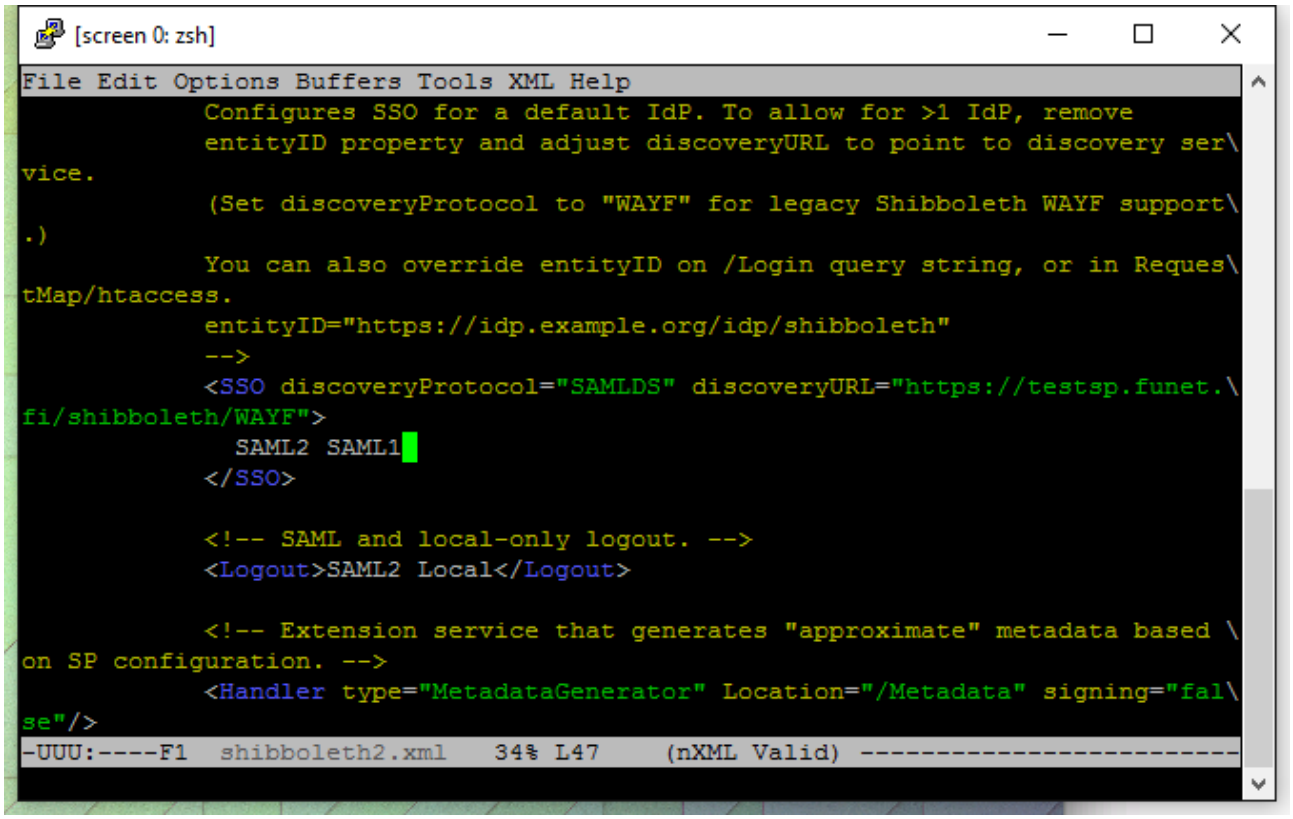
Identity Providerin lokitaso määriteltiin tiedostoon /opt/shibboleth-idp/conf/logback.xml. Laboratorioympäristössä tehtiin seuraavat muutokset:

```
  <variable name="idp.loglevel.ldap" value="WARN" />
--->
  <variable name="idp.loglevel.ldap" value="DEBUG" />

  <logger name="net.shibboleth.idp" level="${idp.loglevel.idp:-INFO}"/>
--->
  <logger name="net.shibboleth.idp" level="DEBUG" />

  <logger name="org.opensaml.saml" level="${idp.loglevel.opensaml:-INFO}"/>
--->
  <logger name="org.opensaml.saml" level="DEBUG" />

  <logger name="org.ldaptive" level="${idp.loglevel.ldap:-WARN}"/>
--->
  <logger name="org.ldaptive" level="DEBUG"/>
```

4.7 Testifederaatioon liitettävän instanssin rakentaminen

Lopullisen Service Providerin rakentamisessa ensimmäinen askel oli hankkia kone, jota oli mahdollista pitää jatkuvasti päällä. Laboratorioympäristö oli rakennettu henkilökohtaiselle pöytäkoneelle (Active Directory -virtuaalikone) ja kannettavalle (Shibboleth -palvelimia ylläpitävä virtuaalikone). Tavoitteena oli tällä kertaa rakentaa tuotantoympäristöön pienellä vaivalla siirrettävissä oleva, Hakan testifederaatioon liitettävä Shibboleth - ympäristö.

Kyamk antoi virtuaalikoneen, mutta ei alidomainia. Osoitteesta http://noip.com rekisteröitiin ddns-osoite jotta palvelimeen viitattaessa ei olisi tarvinnut muistaa sen IP:tä. Koneelle asennettiin Service Provider edellä selitettyjen ohjeiden mukaan, entityID:llä ”http://{ddns-osoite}/shibboleth”. XPath-polkuun /SPConfig/ApplicationDefaults/Istuntons/SSO oli lisättävä testihakan WAYF/DS - hakupalvelun osoite seuraavan kuvan osoittamalla tavalla:

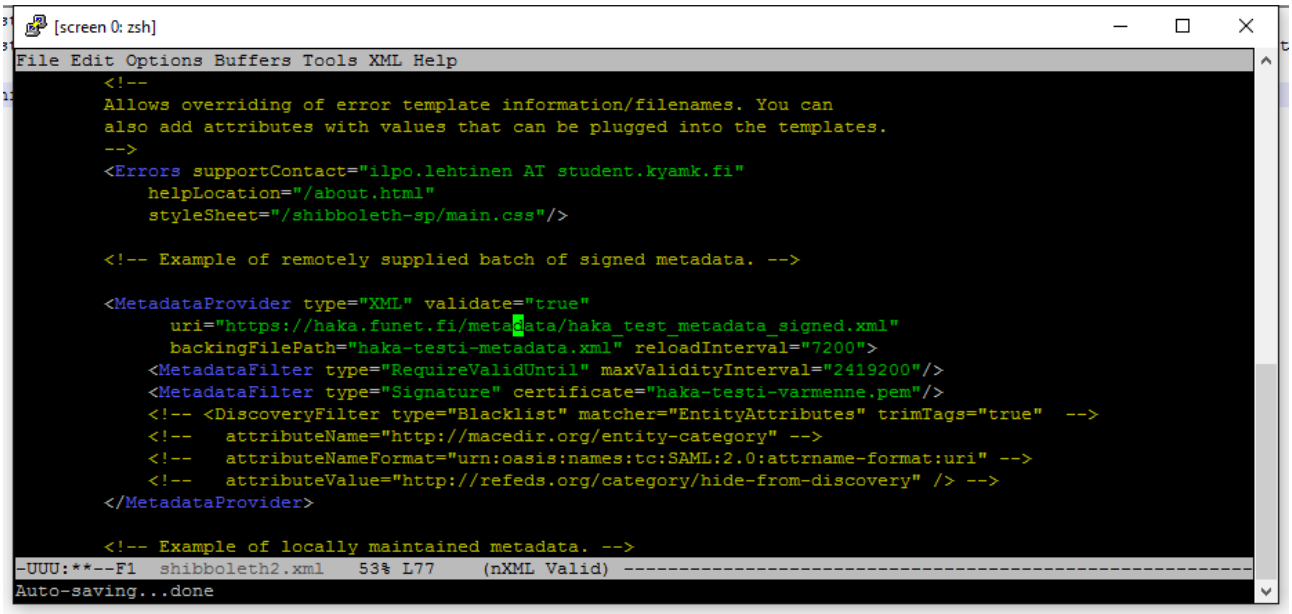


Kuva 15 - WAYF-asetus

Seuraavaksi tarvitsimme Haka Testi - federaation metadatan allekirjoituksineen. Allekirjoituksen hyödyntäminen ei ollut yksinkertaista, koska Service Provider halusi sen .pem-muotoisena, mutta CSC tarjoaa sen .crt - päätteellä. Seuraava ajettuna pääkäyttäjän terminaalissa korjasi ongelman:

curl -o haka-testi-metadata.xml https://haka.funet.fi/metadata/haka_test_metadata_signed.xml
curl -o haka-testi-varmenne.crt
”https://confluence.csc.fi/download/attachments/31195585/haka_testi_2015_sha2.crt?version=1&modificationDate=1430212953940&api=v2&download=true”
openssl x509 -inform PEM -in haka-testi-varmenne.crt -out haka-testi-varmenne.pem -text

Jonka jälkeen shibboleth2.xml:ään lisättiin seuraava MetadataProvider - tagi:



Kuva 16 - MetadataProvider - tagiin lisätty hakan metadata allekirjoituksineen

Kun metadata oli liitetty, ’shibd -t’ ei tulostanut virheilmoituksia, eikä lokeissa näkynyt mitään korjattavaa, oli aika lisätä palvelun tiedot CSC:n resurssirekisteriin. Sivustolle https://rr.funet.fi/rr/menu.php kirjauduttiin omilla koulutunnuksilla, siirryttiin Manage SPs - välilehdelle, klikkattiin Add a new Service Provider - linkkiä ja lisättiin palvelun tiedot. Palvelu oli tärkeää liittää Haka Testiorganisaatioon.

”Yleisesti ottaen Haka-operoinnin vahva näkemys on, että kehitys- ja testivaiheessa ei pidä käyttää aitoja henkilötietoja. Lähtökohtaisesti Hakan tuotantometadataan ei pitäisi rekisteröidä testi- tai kehitysympäristöä, vaan testaaminen ja kehittäminen pitäisi toteuttaa Hakan testiympäristössä. Jos palvelun testaaminen on välttämätöntä toteuttaa tuotanto-IdP:n kanssa, pitäisi riskejä rajata vähintään niin, että luottosuhde tehdään paikallisesti kyseisen organisaation IdP:n ja testattavan palvelun välillä sen sijaan, että testattava palvelu rekisteröidään varsinaiseen Haka-metadataan.” (Laalo 2015).

Työssä päästiin niin pitkälle että testipalvelin ohjasi Hakan kirjautumissivulle, ja kirjautumisen jälkeen tulosti Unexpected Error - virheviestiä.

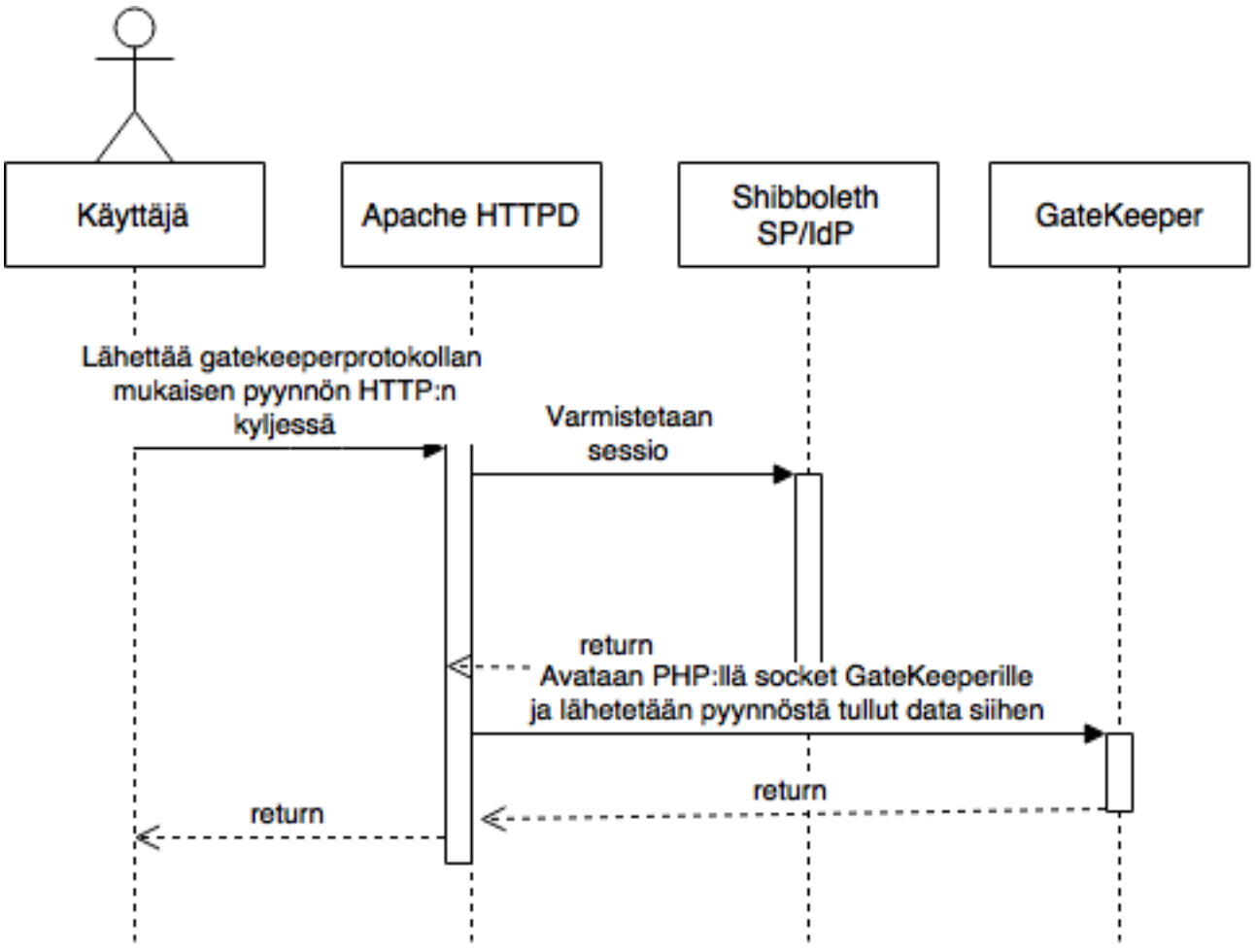
5 OHJELMOINTI

5.1 Mitä koodi tekee nyt?

C++-gatekeeper palvelin pystyttää Raknetillä TCP-palvelinsocketin. Protokolla on huonosti kuvattu. Kirjautuminen toimii niin että palvelimella on oma pyyntötyyppinsä pyhitetty kirjautumiselle. Kirjautumispyynnön tullessa tarkistetaan vastaavatko pyynnön tunnukset kovakoodattuja Kyamk-tunnuksia. Jos eivät, kysytään Redis-tietokannalta saako tämä käyttäjä kirjautua. Muut pyynnöt eivät varmista onko istuntoa olemassa, mikä on ilmiselvä tietoturvariski. Tämä ongelma olisi helppo korjata jos kaikki kommunikaatio tapahtuisi HTTP:llä, mutta nyt tilanne ei ole niin helppo.

5.2 Mitä muutoksia tarvitaan?

Helpointa olisi kirjoittaa asiakasohjelman kommunikaatiokerros upottamaan gatekeeperille tarkoitettu data HTTP-pyynnön sisään. Esimerkkiarkkitehtuuri voisi olla esimerkiksi tällainen:



Kuva 17 - Arkkitehtuuri Shibboleth-välityspalvelimelle

Apache toimii tässä arkkitehtuurissa välityspalvelimena, johon pääsyä Shibboleth valvoo. Jos Shibbolethilta tulee uudelleenohjauskoodi 302 asiakasohjelmaan, se pyytää tunnukset käyttäjältä, muodostaa salatun yhteyden Identity Provideriin ja yrittää autentikoida. Jos autentikaatio onnistuu, palautuu Identity Providerilta sessioeväste, joka täytyy liittää vastedes kaikkiin gatekeeperin HTTP-välityspalvelimelle meneviin pyyntöihin.

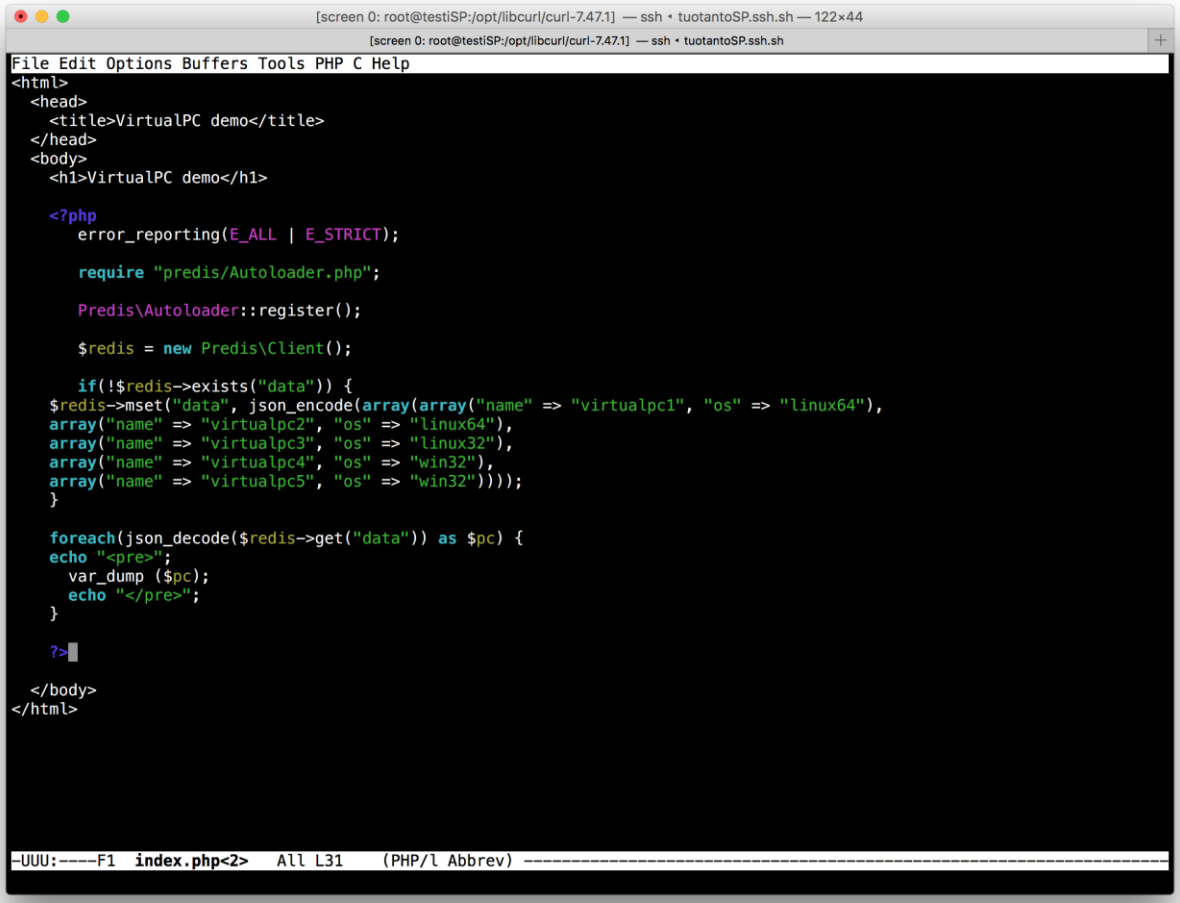
Tarkoittaisiko tämä että Gatekeeper - protokollaan tulisi tehdä muutoksia? Ei. Tallentaakseen tilan http-palvelin liittää vastaukseensa Set-Cookie -otsikon (Barth 2011). Sessiomerkintä tulisi tähän. Gatekeeperille menevä data tulee HTTP-pyynnössä otsikkotietojen jälkeen.

Kun Shibboleth hyväksyy sessioevästeen, ajaa Apache PHP-skriptin, joka lukee pyynnön gatekeeper-osan ja Shibbolethilta tulleet käyttäjän yksilöivät attribuutit. Sen jälkeen skripti avaa socket-yhteyden gatekeeperiin, ja lähettää lukemansa pyynnön sinne ja jää odottamaan vastausta. Minkä gatekeeper lähettää takaisin PHP-skripti lähettää asiakasohjelmalle HTTP-vastauksena.

Shibboleth on suunniteltu tällaisia tilanteita varten. HTTP-protokollaa käyttävän internetohjelman kanssa sessionhallinnan ja kirjautumisen voi ulkoistaa täysin Shibbolethille. Se ylläpitää istunnon, ja jos salaus on asennettu oikein, sillä kirjautuminen on turvallista, eikä varsinaisen ohjelmiston toteuttaja voi luoda autentikaatioprosessiin tietoturva-aukkoja vahingossa, koska hän ei kirjoita sitä. Service Provider pysyy ajan tasalla Linuxissa pitämällä käyttöjärjestelmän ajan tasalla, koska Linuxin pakettienhallinta osaa pitää myös käyttäjän asentamat ohjelmat ajan tasalla.

5.3 Työn viimeistely

Sovittiin että opinnäytetyö olisi tehty kun testihakaan liitetty palvelin saadaan tulostamaan Shibbolethin suojaamana Redis-kannasta testidataa. Tällaisesta virtuaalikoneesta työn jatkokehittäjän olisi helppo lähteä liikkeelle, kun ei tarvitsisi harjoitella Shibbolethin asetusten määrittelyä. Koulun myöntämän virtuaalikoneen kanssa oli ongelmia, mutta Redis-esimerkkikoodi valmistui ajallaan.



Kuva 18 - Redis esimerkkikoodi

6 JOHTOPÄÄTÖKSET

Tähän raporttiin on dokumentoitu ympäristön pystytys mahdollisimman tarkasti ja hyvin. Tämä raportti nopeuttanee jatkokehittäjän työtä parilla kuukaudella. Shibbolethin asetukset on monimutkaista määritellä, mutta kun määrittelijä viettää tarpeeksi pitkään dokumentaation syövereissä, löytää hän sieltä vastauksen lähes mihin tahansa ongelmaan.

Työstä valmistui laboratorioympäristö, jossa federoimaton Service Provider -kone juttelee suoraan Identity Providerin kanssa. Identity Providerista on suora yhteys Active Directoryyn. Service Provider suojaa Redis-tietokantaan muutoksia tekevää PHP-skriptiä.

Kappaleen 5.2 mukainen PHP-välityspalvelin ja testifederaatioon liitettävä Shibboleth-palvelin jäävät jatkokehitettäväksi. Skriptin ohjelmointi on kahden päivän työ, mutta federoidun Shibbolethin viimeistely ja palvelimen testaaminen voi vaatia kaksikin kuukautta aikaa.

7 LÄHTEET

Apache Software Foundation. 2016. Welcome - The Apache HTTP Server Project. Saatavissa https://httpd.apache.org [viitattu 11.4.2016]

Barth, A. 2011. RFC 6265 - Http State Management Mechanism. Saatavissa: https://tools.ietf.org/html/rfc6265#section-3 [viitattu 14.4.2016]

Cover, R. 2010. Cover Pages: Security Assertion Markup Language (SAML). Saatavissa: http://xml.coverpages.org/saml.html [viitattu 31.3.2016]

CSC. 2014. CSC - Haka-käyttäjätunnistusjärjestelmä - Kaikki palvelut. Saatavissa: https://www.csc.fi/fi/-/haka-kayttajatunnistusjarjestel-1?_82_languageId=fi_FI [viitattu 30.3.2016]

empoweredID. 2016. Service Providers, Identity Providers & Security Token Services. Saatavissa: https://www.empowerid.com/learningcenter/technologies/service-identity-providers [viitattu 31.3.2016]

Finlex. 2015. Rikoslaki luku 38 pykälä 7. Saatavissa: http://finlex.fi/fi/laki/ajantasa/1889/18890039001 [viitattu 15.4.2016]

Finlex. 2015. Rikoslaki luku 38 pykälä 8. Saatavissa: http://finlex.fi/fi/laki/ajantasa/1889/18890039001#L38 [viitattu 15.4.2016]

Killeen, E. 2012. What is federation? And how is it different from SSO? Saatavissa: http://blog.empowerid.com/blog-1/bid/164625/What-is-federation-And-how-is-it-different-from-SSO [viitattu 31.3.2016]

Laalo, K. 2015. Usein kysytyt kysymykset - Haka-käyttäjätunnistusjärjestelmä. Saatavissa: https://confluence.csc.fi/display/HAKA/Usein+kysytyt+kysymykset#Useinkysytytkysymykset-Olemmerekister%C3%B6im%C3%A4ss%C3%A4kehitysymp%C3%A4rist%C3%B6%C3%A4Hakaan [viitattu 4.4.2016]

Microsoft. 2016. Overview: The Official Microsoft IIS Site. Saatavissa: http://www.iis.net/overview [viitattu 11.4.2016]

Shibboleth Consortium. 2015. Shibboleth Consortium - Service Provider. Saatavissa: https://shibboleth.net/products/service-provider.html [viitattu 31.3.2016]

LIITE 1 - ATTRIBUTE-RESOLVER.XML

```
<?xml version="1.0" encoding="UTF-8"?>

<resolver:AttributeResolver
        xmlns:resolver="urn:mace:shibboleth:2.0:resolver"
        xmlns:pc="urn:mace:shibboleth:2.0:resolver:pc"
        xmlns:ad="urn:mace:shibboleth:2.0:resolver:ad"
        xmlns:dc="urn:mace:shibboleth:2.0:resolver:dc"
        xmlns:enc="urn:mace:shibboleth:2.0:attribute:encoder"
        xmlns:sec="urn:mace:shibboleth:2.0:security"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:mace:shibboleth:2.0:resolver
http://shibboleth.net/schema/idp/shibboleth-attribute-resolver.xsd
                            urn:mace:shibboleth:2.0:resolver:pc http://shibboleth.net/schema/idp/shibboleth-
attribute-resolver-pc.xsd
                            urn:mace:shibboleth:2.0:resolver:ad http://shibboleth.net/schema/idp/shibboleth-
attribute-resolver-ad.xsd
                            urn:mace:shibboleth:2.0:resolver:dc http://shibboleth.net/schema/idp/shibboleth-
attribute-resolver-dc.xsd
                            urn:mace:shibboleth:2.0:attribute:encoder
http://shibboleth.net/schema/idp/shibboleth-attribute-encoder.xsd
                                                    urn:mace:arcs.org.au:shibboleth:2.0:resolver:dc
classpath:/schema/arcs-shibext-dc.xsd
                            urn:mace:shibboleth:2.0:security http://shibboleth.net/schema/idp/shibboleth-
security.xsd">

    <!-- ========================================== -->
    <!--      Attribute Definitions                 -->
    <!-- ========================================== -->

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="eduPersonEntitlement"
sourceAttributeID="eduPersonEntitlement">
        <resolver:Dependency ref="staticAttributes" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
def:eduPersonEntitlement" encodeType="false" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.7"
friendlyName="eduPersonEntitlement" encodeType="false" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition xsi:type="ad:Simple" id="givenName" sourceAttributeID="givenName">
      <resolver:Dependency ref="myLDAP" />
      <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
def:givenName" encodeType="false" />
      <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.42"
friendlyName="givenName" encodeType="false" />
    </resolver:AttributeDefinition>


    <resolver:AttributeDefinition xsi:type="ad:Simple" id="surname" sourceAttributeID="sn">
      <resolver:Dependency ref="myLDAP" />
      <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:sn"
encodeType="false" />
      <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:2.5.4.4" friendlyName="sn"
encodeType="false" />
    </resolver:AttributeDefinition>


    <!--
    The uid is the closest thing to a "standard" LDAP attribute
    representing a local username, but you should generally *never*
    expose uid to federated services, as it is rarely globally unique.
    -->
```

```
    <resolver:AttributeDefinition id="uid" xsi:type="ad:Simple" sourceAttributeID="uid">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:uid"
encodeType="false" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.1"
friendlyName="uid" encodeType="false" />
    </resolver:AttributeDefinition>

    <!--
    In the rest of the world, the email address is the standard identifier,
    despite the problems with that practice. Consider making the EPPN value
    the same as your official email addresses whenever possible.
    -->
    <resolver:AttributeDefinition id="email" xsi:type="ad:Simple" sourceAttributeID="email">
        <resolver:Dependency ref="myLDAP" />
        <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-def:mail"
encodeType="false" />
        <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:0.9.2342.19200300.100.1.3"
friendlyName="email" encodeType="false" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition id="homeOrganization" xsi:type="ad:Simple"
sourceAttributeID="homeOrganization">
      <resolver:Dependency ref="staticAttributes" />
      <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:oid:1.3.6.1.4.1.25178.1.2.9" />
      <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.25178.1.2.9"
friendlyName="homeOrganization" />
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition id="homeOrganizationType" xsi:type="ad:Simple"
sourceAttributeID="homeOrganizationType">
      <resolver:Dependency ref="staticAttributes" />
      <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:oid:1.3.6.1.4.1.25178.1.2.10" />
      <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.25178.1.2.10"
friendlyName="homeOrganizationType" />
    </resolver:AttributeDefinition>

    <!-- prerequisite to scripted eduPersonAffiliation -->
    <resolver:AttributeDefinition id="isUnderGrad" xsi:type="ad:Simple"
sourceAttributeID="isUnderGrad">
      <resolver:Dependency ref="myLDAP" />
      <!-- no encoder needed -->
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition id="isPostGrad" xsi:type="ad:Simple" sourceAttributeID="isPostGrad">
      <resolver:Dependency ref="myLDAP" />
      <!-- no encoder needed -->
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition id="isStaff" xsi:type="ad:Simple" sourceAttributeID="isStaff">
      <resolver:Dependency ref="myLDAP" />
      <!-- no encoder needed -->
    </resolver:AttributeDefinition>

    <resolver:AttributeDefinition id="eduPersonAffiliation" xsi:type="ad:Script">
      <resolver:Dependency ref="isUnderGrad" />
      <resolver:Dependency ref="isPostGrad" />
      <resolver:Dependency ref="isStaff" />
      <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
def:eduPersonAffiliation" />
      <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.1"
friendlyName="eduPersonAffiliation" />
      <ad:Script>
                <![CDATA[
```

```
                                is_UnderGrad = true;
                                is_PostGrad = true;
                                is_Staff = true;

if (is_Staff) { eduPersonAffiliation.getValues().add("staff"); };
if (is_UnderGrad || is_PostGrad ) { eduPersonAffiliation.getValues().add("student"); };
if (is_UnderGrad || is_PostGrad || is_Staff ) { eduPersonAffiliation.getValues().add("member"); };
                ]]>
    </ad:Script>
  </resolver:AttributeDefinition>

  <resolver:AttributeDefinition id="eduPersonPrimaryAffiliation" xsi:type="ad:Script">
    <resolver:Dependency ref="isUnderGrad" />
    <resolver:Dependency ref="isPostGrad" />
    <resolver:Dependency ref="isStaff" />
    <resolver:AttributeEncoder xsi:type="enc:SAML1String" name="urn:mace:dir:attribute-
def:eduPersonPrimaryAffiliation" />
    <resolver:AttributeEncoder xsi:type="enc:SAML2String" name="urn:oid:1.3.6.1.4.1.5923.1.1.1.5"
friendlyName="eduPersonPrimaryAffiliation" />
    <ad:Script>
                <![CDATA[
                                eduPersonPrimaryAffiliation.getValues().add("staff");
                ]]>
    </ad:Script>
  </resolver:AttributeDefinition>


  <resolver:AttributeDefinition xsi:type="ad:Scoped" id="eduPersonScopedAffiliation"
scope="%{idp.scope}" sourceAttributeID="eduPersonAffiliation">
    <resolver:Dependency ref="eduPersonAffiliation" />
    <resolver:AttributeEncoder xsi:type="enc:SAML1ScopedString" name="urn:mace:dir:attribute-
def:eduPersonScopedAffiliation" encodeType="false" />
    <resolver:AttributeEncoder xsi:type="enc:SAML2ScopedString"
name="urn:oid:1.3.6.1.4.1.5923.1.1.1.9" friendlyName="eduPersonScopedAffiliation"
encodeType="false" />
  </resolver:AttributeDefinition>


  <resolver:DataConnector id="myLDAP" xsi:type="dc:LDAPDirectory"
    ldapURL="%{idp.attribute.resolver.LDAP.ldapURL}"
    baseDN="%{idp.attribute.resolver.LDAP.baseDN}"
    principal="%{idp.attribute.resolver.LDAP.bindDN}"
    principalCredential="%{idp.attribute.resolver.LDAP.bindDNCredential}"
    useStartTLS="%{idp.attribute.resolver.LDAP.useStartTLS:false}">
    <dc:FilterTemplate>
      <![CDATA[
        %{idp.attribute.resolver.LDAP.searchFilter}
      ]]>
    </dc:FilterTemplate>
    <dc:ReturnAttributes>%{idp.attribute.resolver.LDAP.returnAttributes}</dc:ReturnAttributes>

  </resolver:DataConnector>


  <!-- Static Connector -->
  <resolver:DataConnector id="staticAttributes" xsi:type="dc:Static">
    <dc:Attribute id="eduPersonEntitlement">
                <dc:Value>urn:mace:dir:entitlement:common-lib-terms</dc:Value>
    </dc:Attribute>
    <dc:Attribute id="o">
                <dc:Value>Ilponorganisaatio</dc:Value>
    </dc:Attribute>
    <dc:Attribute id="homeOrganization">
                <dc:Value>world.server</dc:Value>
```

```
    </dc:Attribute>
    <dc:Attribute id="homeOrganizationType">
                <dc:Value>urn:mace:terena.org:schac:homeOrganizationType:int:university</dc:Value>
    </dc:Attribute>
  </resolver:DataConnector>

</resolver:AttributeResolver>
```

LIITE 2- ATTRIBUTE-FILTER.XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Ilmeisesti tämä tiedosto määrittelee mitä attribute-resolver.xml:ssä määriteltyä juttua tuutataan
ulkomaailmaan -->

<AttributeFilterPolicyGroup id="ShibbolethFilterPolicy"
                                                    xmlns="urn:mace:shibboleth:2.0:afp"
                                                    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
                                                    xsi:schemaLocation="urn:mace:shibboleth:2.0:afp
http://shibboleth.net/schema/idp/shibboleth-afp.xsd">

 <!-- Release some attributes to an SP. -->
 <AttributeFilterPolicy id="example2">
  <PolicyRequirementRule xsi:type="OR">
   <Rule xsi:type="Requester" value="http://localhost/shibboleth" />
   <!-- <Rule xsi:type="Requester" value="https://another.example.org/shibboleth" /> -->
  </PolicyRequirementRule>

  <AttributeRule attributeID="displayName">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

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

  <AttributeRule attributeID="cn">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

  <AttributeRule attributeID="sn">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

  <AttributeRule attributeID="givenName">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

  <AttributeRule attributeID="uid">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

  <AttributeRule attributeID="mail">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>

  <AttributeRule attributeID="auEduPersonSharedToken">
   <PermitValueRule xsi:type="ANY"/>
  </AttributeRule>

  <AttributeRule attributeID="eduPersonScopedAffiliation">
   <PermitValueRule xsi:type="ANY" />
  </AttributeRule>
```

```
  </AttributeFilterPolicy>
</AttributeFilterPolicyGroup>
```