

TIE-30501 - Identiteetin ja pääsynhallinta

Harjoitus 2

1 Alkuvalmistelut

Ennen työn suoritusta avattiin SSH-yhteys TTY:n linux-ssh-palvelimelle, jolle luotiin aiemmin avainpari pomppu-koneelle pääsyä varten. Pomppu-kone sijaitsee osoitteessa 130.230.112.15.

2.1 Tehtävä 2.1

Aluksi otettiin pomppu-koneelta yhteys user-koneelle SSH-avainparin luomista varten. Avainpari luotiin komennolla `ssh-keygen -t rsa -b 4096` ja sille asetettiin myöskin salasana *salasana*. Salasana suojaa avainparia siinä tapauksessa, että tiedosto syystä tai toisesta päättyy hyökkääjälle. Luotujen tiedostojen käyttöoikeudet olivat valmiiksi riittävät, mutta työhöjeen mukaisesti asetettiin kuitenkin julkiselle avaimelle uudet oikeudet niin, etteivät muut käyttäjät voi lukea sitä (`chmod 600 ~/.ssh/id_rsa.pub`).

Seuraavaksi lisättiin juuri luotu julkinen avain www-koneelle käyttäjälle *septaa*. Avain siirrettiin komennolla `ssh-copy-id -i ~/.ssh/id_rsa.pub septaa@www`, mutta sen olisi voinut vain kopioida tekstinä *septaa*:n `.ssh`-kansion `authorized_keys`-tiedostoon. Tämän jälkeen testattiin user-koneelta kirjautumista komennolla `ssh septaa@www.acme.iam` ja avainparin salasanan syöttämisen jälkeen kirjautuminen meni läpi. Julkisen avaimen sormenjälki on nyt `SHA256:p52BCiAaVdS6g3vrJ5L67fnhYXdGoG2bfpXWxpW5GSQ root@user (RSA)`.

2.2 Tehtävä 2.2

TODO

2.3 Tehtävä 2.3

CA-serverille kirjauduttiin root-käyttäjällä. Tiedostojärjestelmän juureen luotiin uusi hakemisto *super-ca*, jonka sisälle luotiin tehtävässä määritellyt hakemistot. Kansioon *ca* lisättiin tyhjä tiedosto `index.txt` ja `serial`, jonka sisällöksi tuli 1001. Tämän lisäksi `private`-kansion oikeuksiksi määriteltiin 600 `chmod`in avulla.

Kansioon *conf* luotiin tiedosto `openssl.cnf`, jonka sisältö kopioitiin Moodlen esimerkkiedostosta. Varmentajalla on enemmän oikeuksia verrattaen `server_certiin`, sillä nimenomaan varmentaja on se, joka allekirjoittaa varmenteita. `Server_certiä` ei käytetä varmenteiden allekirjoittamiseen, joten niihin ei oikeuksiakaan anneta. Päävarmentajaan luotetaan kaikkein eniten, välivarmentajaan toiseksi eniten ja muihin varmenteisiin vähiten.

2.4 Tehtävä 2.4

CA:lle luotiin RSA-avain ja sille annettiin salasana *master* ja oikeat tiedosto-oikeudet. Avaimen avulla luotiin varmenne, johon syötettiin tehtävässä annetut tiedot. Varmenteen tiedot saatiin tulostettua komennolla *openssl x509 -noout -text -in public/ca.cert.pem* ja ne näkyvät kuvassa 1.

```
root@ca:/super-ca/ca# openssl x509 -noout -text -in public/ca.cert.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 14278795783467340317 (0xc628791cddea921d)
    Signature Algorithm: sha256withRSAEncryption
    Issuer: C=Fi, ST=Pirkanmaa, L=Tampere, O=IAM TUT 2017, CN=IAM CA/emailAddress=lauri.haavisto@acme.iam
    Validity
      Not Before: Sep 20 16:27:22 2018 GMT
      Not After : Sep 15 16:27:22 2038 GMT
    Subject: C=Fi, ST=Pirkanmaa, L=Tampere, O=IAM TUT 2017, CN=IAM CA/emailAddress=lauri.haavisto@acme.iam
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (4096 bit)
```

Kuva 1 Päävarmenteen tiedot

Välivarmentajaa varten luotiin tarvittava hakemistorakenne, johon luotiin vastaavalla tavalla tiedostot *index.txt* ja *serial*. Konffitiedosto haettiin CA:n hakemistosta ja siihen tehtiin tarvittavat muutokset. Myöskin välivarmentajalle luotiin RSA-avain salasanalla *medium*. Varmenteen saamiseksi suoritettiin CSR, jossa annettiin tehtävässä mainitut tiedot. CSR toimitettiin varmentajalle komennolla *cp* ja tietojen varmentamisen jälkeen luotiin itse varmenne, jonka tiedot näkyvät kuvassa 2. Tämän jälkeen kopioitiin varmenne välivarmentajan tiedostoihin.

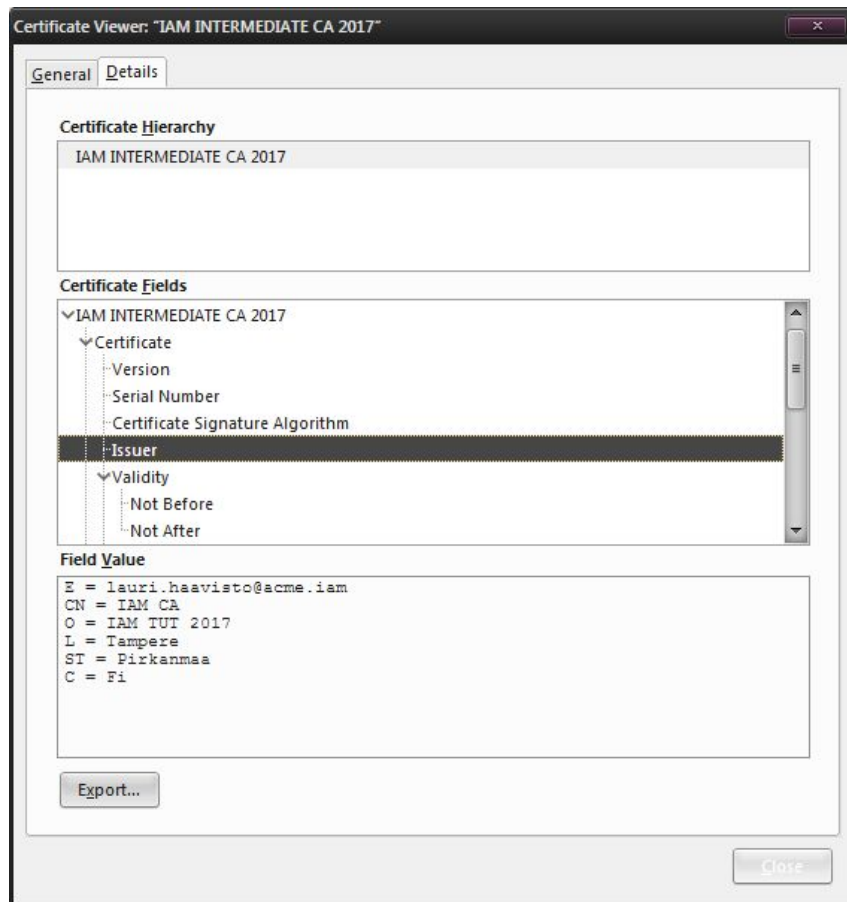
```
root@ca:/super-ca/intermediate# openssl x509 -noout -text -in public/intermediate.cert.pem
Certificate:
  Data:
    Version: 3 (0x2)
    Serial Number: 4097 (0x1001)
    Signature Algorithm: sha256withRSAEncryption
    Issuer: C=Fi, ST=Pirkanmaa, L=Tampere, O=IAM TUT 2017, CN=IAM CA/emailAddress=lauri.haavisto@acme.iam
    Validity
      Not Before: Sep 20 16:58:31 2018 GMT
      Not After : Sep 17 16:58:31 2028 GMT
    Subject: C=Fi, ST=Pirkanmaa, O=IAM TUT 2017, CN=IAM INTERMEDIATE CA 2017/emailAddress=lauri.haavisto@intermediate.iam
    Subject Public Key Info:
      Public Key Algorithm: rsaEncryption
      Public-Key: (4096 bit)
```

Kuva 2 Välivarmenteen tiedot

Lopuksi luotiin varmenneketjutiedosto yhdistämällä pää- ja välivarmentajan tiedot tiedostoon *cert-chain.cert.pem*.

2.5 Tehtävä 2.5

Julkisten varmenteiden jakamista varten pystytettiin apache2-palvelin ja Apachen konffitiedostoon asetettiin sertifikaateille tyyppi *application/x-x509-ca-cert*. Varmenteet kopioitiin www-hakemistoon tiedostonimillä tut-suprt-ca.crt.pem, tut-intermediate-ca.crt.pem ja tut-chain.crt.pem. Apache uudelleenkäynnistettiin ja index.html poistettiin. Välivarmentajan sertifikaatti Firefox-selaimelta tarkasteltuna on esillä kuvassa 3.



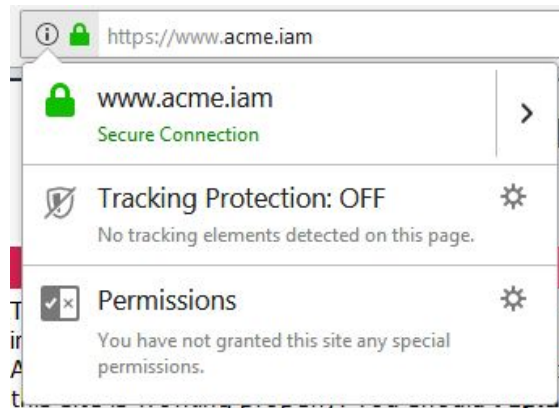
Kuva 3 Välivarmenteen Issuer-kohta Firefoxista tarkastellen

Pystytetyllä nettipalvelimella ei ole käytössä SSL-salausta, joten palvelimen identiteettiä ei nykyisellään pysty varmentamaan. Hyökkääjä pystyisi esim. DNS spoofaamalla tekeytymällä palvelimeksi ja huijaamaan muita käyttämään itse luomiaan varmenteita.

2.6 Tehtävä 2.6

WWW-sivun suojaamista varten otettiin SSH-yhteys www-koneeseen ja luotiin sille aluksi RSA-avain kansioon `/etc/ssl/`. Seuraavaksi luotiin CSR, jossa common name -kohtaan annettiin osoite www.acme.iam ja sähköpostiksi lauri.haavisto@www.acme.iam. Muut kohdat jätettiin oletukselle. CSR toimitettiin välivarmentajalle ca-koneelle suoraan kopiaimalla ja sen tarkistamisen jälkeen luotiin varsinainen varmenne. Laajennokseksi valittiin `server_cert`, koska varmenne tulee www-palvelimelle käyttöön eikä sillä allekirjoiteta uusia varmenteita.

Tämän jälkeen varmenne kopioitiin www-palvelimelle kansioon `/etc/ssl/`. Avaimen ja varmenteen lisäksi haettiin myöskin välivarmenne ca-palvelimelta komennolla `wget http://ca.acme.iam/tut-chain.crt.pem`, jolle annettiin nimi `acme-ca-chainfile.pem`. Nämä tiedostot lisättiin Apachen default-ssl-konfigitiedostoon ohjeen mukaisesti. SSL-sivun ja SSL-moduulin päälle laittamisen jälkeen (komento `a2ensite default-ssl ja a2enmod ssl`) käynnistettiin vielä www-palvelin uudelleen. Tämän jälkeen vierailtiin Firefoxilla osoitteessa <https://www.acme.iam>.



Kuva 4 Suojattu yhteys osoitteeseen www.acme.iam

Selain antoi aluksi varoituksen varmenteesta, mutta kun käytiin `ca.acme.iam`-sivulla hyväksymässä päävarmenne, niin varoitus katosi (kuva 4).

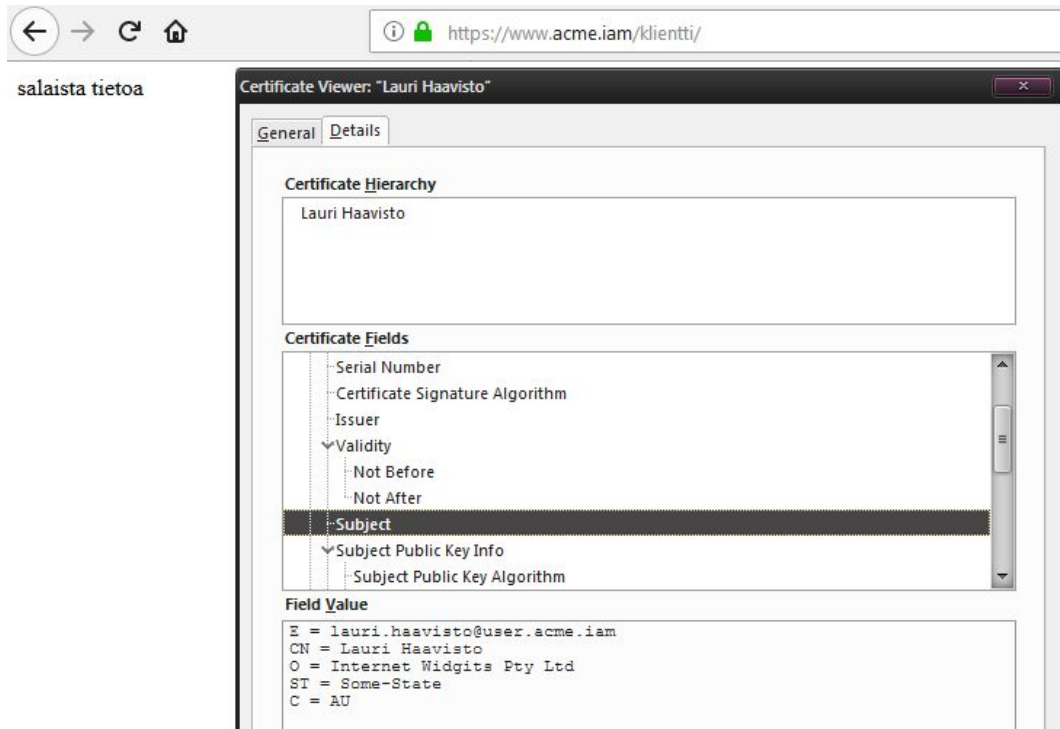
2.7 Tehtävä 2.7

Henkilövarmenteen luomiseksi otettiin SSH-yhteys user-koneeseen. Avain toimii toimii 2-factor-tunnistautumisena, kun voidaan varmistautua siitä, että se on myöskin salasanasuojattu. Edellisen tehtävän tapaan aluksi luotiin käyttäjälle avain, jota käytettiin CSR:n luomiseen. CSR siirrettiin välivarmentajalle, joka loi varmenteen käyttäjälle. Kun varmenne oli siirretty myöskin käyttäjän haltuun, luotiin siitä ja yksityisestä avaimesta pkcs12-paketti, jolle asetettiin salasana avaimen turvaamiseksi.

Paketin luomisen jälkeen konffattiin www-palvelimen default-ssl-tiedostolle uusi lokaatio kansiolle `/klientti`, johon määriteltiin ohjeen mukaiset asetukset. Lisäksi asetettiin

SSLCACertificateFile:n arvoksi *acme-ca-chainfile.pem*. Palvelimen uudelleenkäynnistämisen jälkeen luotiin vielä fyysisesti kansio */var/www/html/klientti/*, johon lisättiin *index.html*.

Firefoxilla sijaintiin */klientti* ei vielä ole pääsyä, koska käyttäjän avainmateriaali puuttuu. Selain antaa ilmoituksen "Secure Connection Failed" ja Apachen *error.log*ista löytyy rivi "peer did not return a certificate -- No CAs known to server for verification?". Pääsy onnistui kuitenkin heti sen jälkeen, kunhan vain lisättiin selaimeen .p12-tiedosto. Kuvasta 5 näkyy salattu yhteys luotuun hakemistoon käyttäen varmennetta.



Kuva 5 Sivun <https://www.acme.iam/klientti> ja henkilökohtainen varmenne

SSLUserName ja SSLRequire -asetukset näyttivät myöskin toimivan hyvin.