LAUREA
AMMATTIKORKEAKOULU
Yhdessä enemmän

# SSL-sertifikaattien päivittäminen ja hallinnointi palvelinympäristössä

Puumalainen, Leevi

2017 Laurea

Laurea-ammattikorkeakoulu

# SSL-sertifikaattien päivittäminen ja hallinnointi palvelinympäristössä

Leevi Puumalainen
Tietojenkäsittely
Opinnäytetyö
Tammikuu, 2017

**Laurea-ammattikorkeakoulu** **Tiivistelmä**
Leppävaara
Tietojenkäsittely

Leevi Puumalainen

**SSL-sertifikaattien päivittäminen ja hallinnointi palvelinympäristössä**

Vuosi 2017 Sivumäärä 32

Toiminnallisen opinnäytetyön tavoitteena on luoda ohjeistus ja kuvaus SSL-sertifikaattien päivittämistä ja hallinnointia varten palvelinympäristössä, jotta sertifikaattien päivitys on jatkossa helpompaa. Työn toimeksiantajana on S-Pankki Oy. Työssä tutkittiin SSL-sertifikaatteja ja avattiin kokonaisprosessia niiden hakemisprosessista päivittämiseen. Kyseessä on kahden tai kolmen vuoden välein tapahtuva operaatio, jonka tarkoitus on pitää sertifikaatit ajan tasalla.

Tietoperustana käytetään RFC-dokumentaatiota, joka määrittelee eri komponenttien teknisen taustan ja standardin. Tämän lisäksi on hyödynnetty sertifikaattien myöntäjien tekemiä ohjeita ja esimerkkejä.

Tutkimusmenetelmänä on käytetty toimintatutkimusta sekä tehtyjä havaintoja työkaluista ja prosessista. Ohjelmistoista on tarkasteltu tarkemmin Windows Server 2012 R2-käyttöjärjestelmää ja OpenSSL-sertifikaattien hallinnointityökalua. Näiden lisäksi Keytool on ollut hyödyllinen työkalu sertifikaattien viemiseen Javan luotettujen sertifikaattien kirjastoon.

Opinnäytetyön tuloksena luotiin pikaohje sertifikaattien päivittämistä varten. Lisäksi työssä on havainnollistettu esimerkin avulla tietoperustaa hyödyntäen päivittämisprosessin toteutus. Lopputukoksena sertifikaatit saatiin päivitettyä palvelimille onnistuneesti. Prosessin kehitysehdotuksena syntyi ajatus toiminnan tehostamisesta, sekä mahdollinen jatkotutkimus PKI (Public Key Infrastructure)-ympäristön luomiseksi omia ei-kaupallisia sertifikaatteja varten.

Asiasanat: SSL, TLS, palvelin, sertifikaatti, Windows

**Laurea University of Applied Sciences** **Abstract**
Leppävaara
Bachelor's degree Program in Business Information Technology

Leevi Puumalainen

**Updating and Managing SSL Certificates in Server Environment**

Year 2017 Pages 32

The target of this thesis is to update and manage SSL certificates in server environment. The client of the work is S-Pankki Oy. The aim was to study the background of the SSL certificate and to clarify the overall process from applying to updating those into system. The task was to create guidelines and description of the measures, in order to perform same kind of a process in the future with less effort. Every two to three years this process needs to be repeated to keep certificates up to date and that's why it's useful to gather important information regarding the process.

The theoretical background for the thesis comes from RFC documents, which define in detail the technical standards of different components and formats. In addition, the instructions and the examples made by the Certification Authorities have been utilized.

The research method of this thesis covers action research and thesis worker's own observations about tools and processes. The Windows Server 2012 R2 operating system and certificate managing tool OpenSSL play the most important software role. In addition to these, Keytool has been as a useful tool to import certificates into Java's trusted certificate library.

The result of the thesis process was a quick guide for updating certificates and also an applied example of the process that was supported with the theoretical background. The result was as desired. On the other hand, certificates were updated to servers successfully. As a result there was also an improvement idea to intensify process and possible further research about PKI (Public Key infrastructure) and how to create it for self-signed certificates.

Keywords: SSL, TLS, server, certificate, Windows

Sisällys

# 1 Johdanto

Opinnäytetyön aiheena on SSL-sertifikaattien päivittäminen palvelinympäristöön. Aihe on tärkeä tietoturvan ja käytännön toiminnallisuuden näkökulmasta. Työn tilaaja-asiakas on S-Pankki. Ympäristönä ovat SAS:n palvelimet, jotka toimivat Windows Server 2012 R2-käyttöjärjestelmän päällä. Tarve sertifikaattien päivittämiseen huomattiin käytönyhteydessä. Tässä yhteydessä tuli ilmi muun muassa vanhemman mallisten sertifikaattien suojauksen vanheneminen. Ensimmäisen kerran, kun lähdimme päivittämään sertifikaatteja, pyrimme vain pikaisesti saamaan voimassaolevat sertifikaatit järjestelmään. Tämän seurauksesta huomasimme Firefox- ja Chrome-selaimien virheilmoitukset liittyen sertifikaattien luotettavuuteen. Näistä havainnoista johtuen päätettiin tutkia päivitysprosessia tietoperustaan tukeutuen ja toteuttaa koko päivitysprosessi opinnäytetyönä.

Työn tavoitteena on kartoittaa ja dokumentoida sertifikaattien anomisprosessi ja avata aiheeseen liittyvää akronyymiviidakkoa eli moniosaisen ilmauksen alkukirjaimista muodostuvia koostelyhenteitä. Tavoitteen mukaisena lopputuloksena syntyy toimivan prosessin kuvaaminen dokumentoituna. Onnistuneeseen lopputulokseen päästään hyödyntämällä toimintatutkimuksellista tutkimusmenetelmää, jossa pyritään keskittymään käytännön läheiseen ratkaisuun, joka on pohjustettuna vankalla tietoperustalla.
Työn keskeisiä lähteitä on SAS:n julkinen dokumentaatio ja erinäiset RFC-dokumentit, joissa määritellään SSL:n toimintaperiaatteita. Lisäksi tulee ottaa huomioon S-Pankkia koskevat viranomaisten asettamat tietoturvavaatimukset, jotka erityisesti kohdistuvat pankkialaan.

# 2 Sertifikaattien tietoperusta

Seuraavan luvun aikana käydään läpi tärkeitä terminologisia sanoja, joita tarvitaan tämän opinnäytetyön läpikäynnissä. Näiden sanojen avaaminen on erittäin tärkeää niin ymmärryksen, kuin kokonaisuuden hahmottamisen vuoksi. Lisäksi teorialuku aloitetaan esittelemällä tutkimusmenetelmä.

## 2.1 Toimintatutkimus tutkimusmenetelmänä

Toimintatutkimus sisältää erilaisia alalajeja, joita on esimerkiksi kriittinen, tekninen ja praktinen toimintatutkimus (Eriksson & Kovalainen 2008, 196-198). Alalajeille on hyvin paljon yhtäläisyyksiä vuorovaikutteiseen ja tutkivaan toimintatapaan (Eriksson & Kovalainen 2008, 208). Tilannesidonnaisuus on hyvin tyypillistä tämän tyylisissä tutkimusmenetelmissä ja siinä halutaan päästä ratkaisemaan käytännön ongelmia. Havaittua ongelmaa pyritään ratkaisemaan toimintatutkimuksellisissa tapauksissa. Se on loistava valinta tapauksissa, joissa pyritään löytämään käytännönläheinen keino ratkaisu (Metsämuuronen 2006, 102-103). Liike-elämän ongelmanratkaisun ja tutkimuksen tarpeet perustuvat käytännön asioihin ja

yrityksen eri toimintoihin. Tapauskohtaisuus ja käytännönläheisyys ovat hyvin tärkeitä puhuttaessa liike-elämän tarpeista. (Eriksson & Kovalainen 2008, 193-194)

## 2.2 OSI-malli

OSI-malli (Open Systems Interconnection Reference Model) on poikkileikkaus eri tiedonsiirtoprotokollien muodostamasta seitsenkerroksisesta rakenteesta (taulukko 1).

| | |
|---|---|
| 7. Sovelluskerros (Application layer) | HTTP, FTP, SMTP |
| 6. Esitystapakerros (Presentation layer) | GIF (Graphic Interchange Format), JPEG (Joint Photographic Experts Group) |
| 5. Istuntokerros (Session layer) | VoIP, PPTP |
| 4. Kuljetuskerros (Transport layer) | TCP, UDP |
| 3. Verkkokerros (Network layer) | IP, reititin |
| 2. Siirtokerros (Data link layer) | Ethernet, kytkin |
| 1. Fyysinen kerros (Physical layer) | Ethernet, toistin |

Taulukko 1: OSI-malli, Sininen kuvaa ylempää kerrosta ja vihreä alempaa kerrosta

Sen ylin kerros kuvaa käyttäjälle näkyvää kerrosta eli Sovelluskerrosta. Sovelluskerroksen tyypillisimpiä protokollia on FTP (File Transfer Protocol), HTTP (Hypertext Tranfer Protocol) ja SMTP (Simple Mail Transfer Protocol). Seuraavassa kerroksessa on esitystapakerros, joka sisältää kuva-formaatteja. Istuntokerroksessa, joka on seuraavana laskeutuvassa tornissa, on tyypillisesti PPTP (Point to Point Tunneling Protocol), VoIP (Voice over Internet Protocol) ja muita vastaavia ratkaisuja. Kuljetuskerroksessa toimii TCP (Transmission Control Protocol) ja UDP (User Datagram Protocol). Sen tarkoituksena on huolehtia verkkopaketit perille. Seuraava kerros käsittää verkkokerroksen, jossa sijaitsee IP (Internet Protocol). Tähän kerrokseen tyypillinen verkkolaite on reititin, sillä se toimii kolmannessa kerroksessa, eli verkkokerroksessa. Kaksi alinta kerrosta, eli siirtokerros ja fyysinen kerros sisältävät Ethernet tekniikkaa ja niille tyypillisimmät laitteet ovat kytkin siirtokerrokselle ja fyysiselle kerrokselle toistin. (B. Vangie, 2017)

## 2.3 SSL ja TLS

SSL eli Secure Sockets Layer tunnetaan nykyisin nimellä TLS, Transport Layer Security.

Kyseessä on protokolla, jota käytetään Internetissä tapahtuvan tietoliikenteen suojaamiseen IP-verkossa. TLS on yksi käytetyimmistä salausprotokollista ja se on hyvin tavallinen www-sivujen istuntojen suojaamisessa, kun käytetään HTTPS-protokollaa. TLS version 1.2 on uusin versio ja sen määrittely tapahtuu RFC 5246-viitekehyksessä (Request for Comments). RFC:t ovat Internetiä koskevia standardeja ja niitä julkaisee IETF-organisaatio (Internet Engineering Task Force). SSL-sertifikaatit ovat varmenteita ja niiden tarkoituksena on osoittaa sivuston hyvää mainetta. (T. Dierks, E. Rescorla 2008,3-5)
CA vastaavat sertifikaattien allekirjoittamisesta ja tällä pyritään keskittämään luotettavuutta. Esimerkiksi Mozillalla on asetettu luotettavientahojen listalle yli 50 varmentajan myöntäjää. (I. Sani 2011)

### 2.3.1 SSL:n historia

Historiallisesti SSL on merkittävä salausprotokolla ja se oli alun perin kehitetty Netscape-selaimeen. Sen tarkoituksena oli suojata HTTP-yhteyksiä (Hypertext Transfer Protocol) Internetissä. Käytännössä se ei rajoittunut vain HTTP-protokollaan, vaan sitä pystytään hyödyntämään kaikissa TCP (Transmission Control Protocol)-yhteyksissä. Tämän lisäksi nykysyin SSL:n käyttö on laajentunut muihinkin yhteysmuotoihin, kuten SMTP (Simple Mail Transfer Protocol), POP (Post Office Protocol), IMAP (Internet Message Access Protocol), LDAP (Lightweight Directory Access Protocol) ja IRC (Internet Relay Chat). (Netscape 2017; A. Freier, P. Karlton, P. Kocher 2011)

### 2.3.2 PEM-formaatti

PEM eli Privacy Enchanced Mail on vuodelta 1993 oleva ehdotus suojattuun sähköpostitteluun. Se perustuu yhteen PKI ympäristössä olevaan juureen. PEM toimii sovelluskerroksessa, mutta sitä ei koskaan otettu laajasti käyttöön. Tästä huolimatta PEM:in sisältämä x.509 avaintenhallinta standardi. Se on hyvin yleisesti tunnettu ja se toimii kuljetuskerroksessa muun muassa TCP- ja UDP-protokollien kanssa. (J.Linn 1993, 9)

### 2.3.3 PKCS#7 ja PKCS#12

PKCS eli Public Key Cryptography Standards on standardi muoto kryptografiselle ryhmälle. Siihen kuuluvat muun muassa paljon käytetyt PKCS#7 ja PKCS#12. Nämä ovat eri versioita ja malleja, miten sertifikaatteja voidaan tallentaa eri muotoihin. PKCS#7 sisältää pelkän sertifikaatin toisinkuin PKCS#12, joka sisältää salaisen avaimen sertifikaatin lisäksi. PKCS#7 on määritelty RFC 2315 ja PKCS#12 taas RFC 7292. (K. Moriarty, M. Nystrom, S. Parkinson, A. Rusch, M. Scott 2014, 3-6; B. Kaliski 1998, 1-3)

## 2.4 CA

CA (Certificate Authority) on taho, jolla on oikeus myöntää digitaalisia sertifikaatteja. Näitä digitaalisia sertifikaatteja käytetään yleisesti SSL-sertifikaateissa. CA:n rooli SSL-sertifikaateissa on allekirjoittaa, eli vakuuttaa sertifikaatin omistajan olevan se henkilö kuka tämä väittää olevansa. X.509 standardi määrittelee tarkemmin CA:n taustan ja vaatimuksen.


1. Käyttäjä, joka haluaa sertifikaatin itselleen, täytyy luoda itselleen työkalulla
2. salaisen- ja julkisen avaimen, jotka muodostavat parin.
3. Tämän jälkeen käyttäjä generoi itselleen CSR:n (Certificate Signing Request). CSR sisältää käyttäjän julkisen avaimen ja henkilötietoja, kuten nimi, yritys, sähköpostiosoite jne.
6. Hakemuksen savuttua varmenteiden päämyöntäjä tai varmenteiden välittäjän tulee
7. Varmistavat hakijan tiedot ja henkilöllisyys.
8. Tiedot varmistettuaan CA myöntää sertifikaatin, joka sisältää varmistetut tiedot käyttäjästä. (Webtrust 2011, 6)

Kuvio 1: CA-prosessin kuvaus

Yleisimpiä CA-toimijoita on Comodo, Symantec, GoDaddy ja GlobalSign. Nämä yritykset ovat esimerkkejä niistä, jotka allekirjoittavat sertifikaatteja, joita jokainen voi käydä mm. tutkimassa eri HTTPS-sivuille. Nämä neljä yritystä hallitsee yli 90 % markkinoista, joista pelkästään n.70 % on hallussa Comodolla (41,0 %) ja Symantecilla(30,2 %) A W3Techs:in tutkimuksen mukaan vuonna 2015 toukokuussa. (Webtrust 2011, 6; W3techs 2015)

## 2.5 HTTPS

HTTPS eli Hypertext Tranfer Protocol Secure on paljon käytetty sovellus kerroksen protokolla, jossa on yhdistetty HTTP- ja TLS-protokollia. Lähes poikkeuksetta HTTPS-yhteyttä käytetään verkkoliikenteessä, joka halutaan suojata hyvin. Tämmöisiä erilaisia syitä verkkoliikenteensuojaamiseksi on esimerkiksi arkaluontoinen informaatio, joko käyttäjästä tai palveluntarjoajasta. Pankit ovat erittäin tarkkoja tietoturvastaan ja kaikki verkkopankkiyhteydet vaativat suojatun HTTPS-yhteyden. (E. Rescorla 2000, 2; Centero 2012)

## 2.6 PKI

PKI eli Public-Key Infrastructure tunnetaan paremmin suomeksi julkisen avaimen infrastruktuurina. Tätä rakennetta käytetään paljon erityisesti varmentamisessa. PKI koostuu kahdesta avaimesta: public key (julkinen avain), sekä private key (salainen avain). Julkinen avain on jaettavissa laajasti ja se eroaa salaisista avaimista siten, että salainen avain löytyy vain omistajalta. (Webtrust 2011, 7)

## 2.7 CSR

CSR eli sertifikaattien allekirjoitus hakulomake (taulukko 2) on hyvin tärkeä osa sertifikaattien haku tai uusimisprosessia. Ilman CSR:ää ei voida myöntää CA:n myöntämiä tai itse allekirjoitettuja sertifikaatteja. Hakulomake sisältää erilaisia tietoja käyttäjästä/yrityksestä, jonka perusteella luodaan koodin pätkä, joka lähetetään sertifikaattien myöntäjälle (esimerkiksi Symantec:ille). Tarkempi kuvaus prosessin etenemisestä esitetään toteutusvaiheessa. CSR:ssä täytetään osoite tieto, joka on sivuston domain. Mikäli saman osoitteen alla on esimerkiksi: https://julkinen.esimerkki.fi voidaan hakemukseen lisätä SAN (Subject Alternative Name) eli vaihtoehtoinen osoite. Tämä kohta tulee sisällyttää alkuperäiseen kohtaan, mikäli haluaa käyttää sertifikaattia kyseisessä osoitteessa. Jokainen osoite vaati oman kappaleensa, joten tällä tavoin ei voida säästää taloudellisesti, ainoastaan helpottaa anomisprosessia. (M. Nystrom, B. Kaliski 2000, 2-4)

| | |
|---|---|
| CN (Common name) | https://esimerkki.fi |
| O (Organization) | Yritys |
| OU (Organization Unit) | IT |
| L (Locality) | Helsinki |
| S (State) | Finland |
| C (Country) | FI |
| E (Email) | esimerkki@yritys.fi |
| K (Key lenght) | Minimissään 2048 tavua |

Taulukko 2: CSR-tietolomake

Lomakkeen täytön voi suorittaa monella eri tapaa. Yksi näistä on vapaassa jakelussa oleva OpenSSL, joka soveltuu myös omien sertifikaattien allekirjoittamiseen. Toinen vaihtoehto on ladata sertifikaatteja myöntävältä taholta oleva applikaatio, johon tiedot syötetään. Lopputuloksena on julkinen avain, sekä salainen avain, joka tallentuu koneelle sertifikaattikirjastoon tai erilliseen tiedostoon, joka on salasanasuojattu. On tärkeää muistaa, ettei salaista avainta koskaan luovuteta kenellekään ulkopuolisella tai tarpeettomalle henkilölle. Mikäli salainen avain vuotaa on sertifikaatin allekirjoitus yhtä tyhjän kanssa ja sen tuoma tietoturva on olematon. Näin ollen kyseisen avaimen suojeleminen on erittäin tärkeää ja tämän vuoksi yleensä piilotettu palvelimelle. (M. Nystrom, B. Kaliski 2000, 2-4)

```
-----BEGIN CERTIFICATE REQUEST-----
MIICrDCCAZQCADBoMQswCQYDVQQGEwJGSTEWMBQGA1UEAxMNZXNpbWVya2tpLmNv
bTERMA8GA1UEBxMISGVsc2lua2kxDzANBgNVBAoTBllyaXR5czEQMA4GA1UECBMH
RmlubGFuZDELMAkGA1UECxMCSVQwggEiMA0GCSqGSIb3DQEBAQUAA4IBDwAwggEK
AoIBAQDRJVJpUO9t1gOElWwEDROFYwGUVC4EkvGBUT5MffTX8pNlk0CsV7oiHKey
tXMTtPDDxvXLBrbG/CzelPxIfS9y/VM8N0Jp8TS4gHEiESHL+0HEBzMUujcF6Dkp
oPvyAKYdJAAM8f+dGYTD4VUVyaJiX3pOwOYWlnQFO1C9Os7labKpajpw0K/UvKxs
bqMVINhjPWiRBvOS8zfb8TaKMUkybDZE6TcKqj+BzPK69xiFIlAhHFYXIVZMOMYC
V3KAtQG+YEmr29SgVtuaWmPDMONZguMGVM547e/z9koPRHYBfen+w6eeZ69bsjlN
H0ZWyPORprFJYEYN6M/nyBDuOIJhAgMBAAGgADANBgkqhkiG9w0BAQsFAAOCAQEA
jkq3vSH1BeV29PPZ0Kl0cDfiqF8Uv12UGV4OSk+vtr8eD3O+m+qCMKL3fn4rRkGs
LZBlpVxzw+8kMnWN8EzVY2THLP01bVGIlpc8IG9PXeG9uQXA19f2m3x34eYRNZyd
Ihu4y46T4EpBINjAVItQKvEBj+El77p3Hb41qCG87WAA/whJX86Cyb8co8t5dTTU
4wXWfjjh9mADJKmcFul7gRXj0puJ4oPGbHKuJWzimhLCVRWMv61/wYs8rFzrQxf7
lQme7ZrE7PqxMfIKTHh+q8xh3ltl/E0NJ/CvwiCleHijG4GTNdIbUqg6VxMrS6Gt
HlN1ZD4rpJN8D769aDkYoQ==
-----END CERTIFICATE REQUEST-----

-----BEGIN PRIVATE KEY-----
MIIEvgIBADANBgkqhkiG9w0BAQEFAASCBKgwggSkAgEAAoIBAQDRJVJpUO9t1gOE
lWwEDROFYwGUVC4EkvGBUT5MffTX8pNlk0CsV7oiHKeytXMTtPDDxvXLBrbG/Cze
lPxIfS9y/VM8N0Jp8TS4gHEiESHL+0HEBzMUujcF6DkpoPvyAKYdJAAM8f+dGYTD
4VUVyaJiX3pOwOYWlnQFO1C9Os7labKpajpw0K/UvKxsbqMVINhjPWiRBvOS8zfb
8TaKMUkybDZE6TcKqj+BzPK69xiFIlAhHFYXIVZMOMYCV3KAtQG+YEmr29SgVtua
WmPDMONZguMGVM547e/z9koPRHYBfen+w6eeZ69bsjlNH0ZWyPORprFJYEYN6M/n
yBDuOIJhAgMBAAECggEAckAaIfAVq4Gi0FWz+Q6PoGkyXq4RqWwAQ+l2w+B2lVTt
WuhDByKxSpsuvQBESfqpH7nbYbpcNJ1jgaMF4i+IsR9Tzew9+pPKghOgOcT0GnM6
PPTV4bfMAQ0XlPbdSxkxpW+aG3hDc7WSAlNg4JMSgvy3WrgjuSkdXhJGC+lJRqaX
JJ6rclwAEN9Jzs6Y7+eD4LrF3WmrPZ+IRiHd4F8ANBn07qnAgvIYXy6xMA232Aoq
kKluaKKUNXR4KG9nJVPp7SBmDfPrMfD318zx4v0HRwA/J6C3LwwOrmv5nwh+ubiv
mOKTFz20X7fU0vaoWrSyBiDGHBcl7Uj7s5BGF+jGdQKBgQDoU67i5m3EIt5JBhTL
+TILdCDL4eX9lNRnjXsw4jvIYtY39mXzluPqfI2t2ZtD5bALfgSuQf37ME23H97A
ENqWCwbNPdjU6zwyvWOZCk2bk5pbFSKXjbmJa8btouyMsrcrXIZt604uIu5xDgul
7Q9wuB7e5vblIfXluUVPBJRngwKBgQDmdPPsqruSPcENLxLX6d4yw7eP55JzP6Jk
sRiOWJLQJlSLGSvWqiL+tda+25MTvkXO6tA2hBQcojlJ9O4vXEnj42KduklUJKcm
VkiEzcLZroAjQxfk9bi3xitQPuD5/RWiqAeq0vMlbdmCVgivEbbYilNR8+kl3yn4
MGJpEmLlSwKBgGNvvcCbw4tZKS60DZ6nz8WbNJZO33Ne7nJL2dg4XEIOG4XkTgqO
IIqyOKmaI5xR6KFilHpJAH3MVNb2Kw6lqAjkO0sJF486B4/oA4LzJ/hYmo41Y5Le
M/UfTgtT2k72BvPmLBtLxTBl/vwgeNfRuPQTuhLJZBXTLUS4wXNqkGIlAoGBALvr
IfwaPFlDgef+Op9VVJqQfV4atpDDkugIgL2R/CU/7PB/1f91PJP2MuXupj+zJ7lS
P/Y0PTIcFhr4XkDAQIBTKlbzU5sJSEM6mGeyYPCgGlymmRVaSQWR0j/dBhbTCJFZ
8uaYnTk0GejtvqcDCtYIxihNfBAlqVN4IMcM9xeBAoGBAJSTt2tU3j9z33KDpNxj
74BB5eY+mMMXeR5kpj/ZONlwaFt3fcGqwjWjcIw0DBdgGa9bhP/EJ3ReN5KMBFfn
J2rAXABHwcSq4xpgS9Gj7syKZyEsL9jQ78ZB0b8dKSZbgMah/9eEoCP6qVLFnfln
OA9i+JmJlSTVMVBhrIXMDEuQ
-----END PRIVATE KEY-----
```

Kuvio 2: Julkinen- ja salainen avain

Yllä olevassa kuvassa (kuvio 2) on havainnollistava esimerkki miltä näyttää generoitu julkinen- ja salainen avain.

## 2.8 OpenSSL

OpenSSL on avoimeen lähdekoodiin perustuva sertifikaattien luonti- ja hallinnointityökalu. Ohjelma sisältää kirjaston, joka on kirjoitettu C-ohjelmointikielellä ja se sisältää erilaisia kryptografisia funktioita. Projekti on aloitettu vuonna 1998 ja sitä käytti liki kaksi kolmasosaa palvelimista vuonna 2014. Ohjelmisto on alustariippumaton ja sitä saa tarvittaessa laajennettua lisäosilla, jolloin kasvaa mahdollisuus hyödyntää kirjastoa muillakin ohjelmointikielillä. (OpenSSL 2016)

## 2.9 Java truststore-kirjasto

Javassa on truststore, joka tunnetaan nimellä cacert johtuen sen tiedostonimestä. Kirjastoa käytetään sertifikaattien säilömiseen, kun niitä halutaan käyttää Javaa sisältävissä ratkaisuissa. Sinne tallennetut sertifikaatit ovat yleensä PEM-formaatissa, jolloin ne ovat suojaamattomia. Tästä syystä kirjasto on suojattu salasanalla. Kirjaston oletusarvoinen salasana on ”changeit”.

Truststore on mahdollisuus laajentaa uusilla sertifikaatti ketjuilla. Tähän tarvittava esimerkkikoodi on:” keytool -import -file C:\cascerts\firstCA.cert -alias firstCA -keystore myTrustStore”. Tämä tulee toistaa yhteensä kolme kertaa, sillä muutokselle, että ”firstCa” korvataan ”secondCA” ja ”thirdCA”. Näiden sertifikaattien funktio on linkittyä toisiinsa ja näin ollen muodostaa sertifikaattipolun. Tämä on välttämätöntä, mikäli kyseessä on itse allekirjoitetut sertifikaatit. (Oracle 2016)

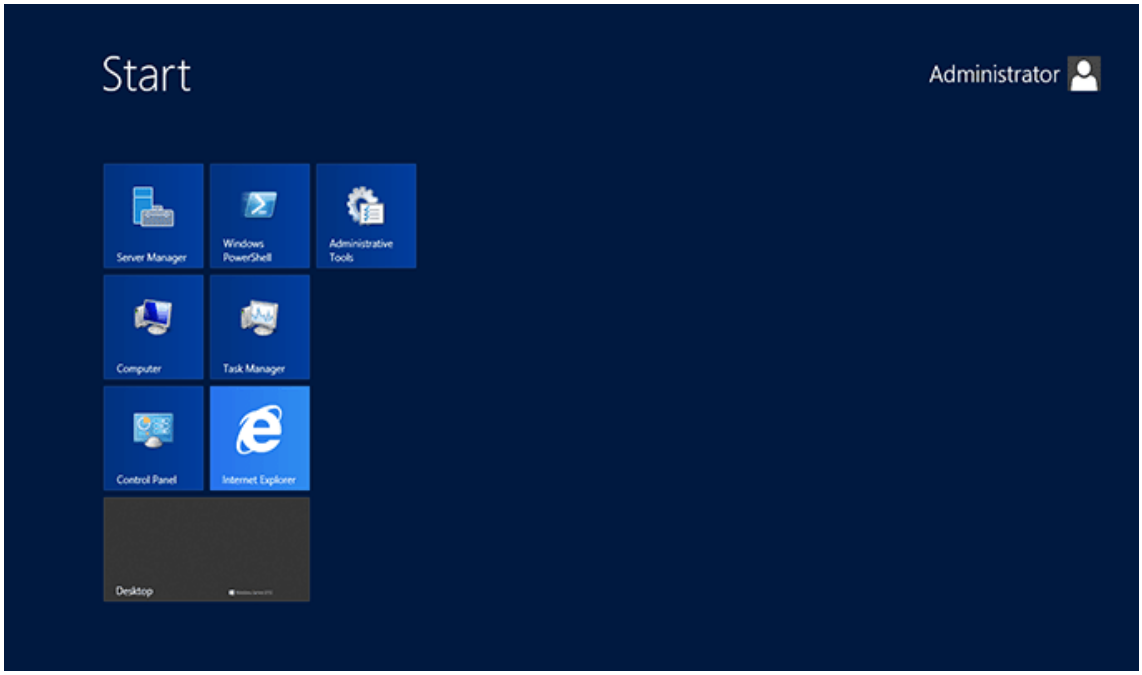
Keytool on Javan työkalu, jolla siirretään sertifikaatteja cacert kirjastoon. Se on erittäin keskeisessä roolissa, sillä monet sovellukset käyttävät sitä esimerkiksi istuntojen varmentamiseksi. Keytool-työkalua voidaan käyttää myös sertifikaattien poistamiseen kirjastosta. Toinen vaihtoehto on lisätä sertifikaatti vanhentuneiden listalle, jolloin sitä ei enää käytetä pyydettäessä. (Oracle 2016)

# 3 Ympäristö

Kohdeympäristö, jossa on tarkoitus toteuttaa sertifikaattien päivittäminen, koostuu useammasta Windows palvelimesta ja jokaisella on oma funktionsa. Näitä funktioita on mm. yhteyspalvelin ja tukevat palvelimet, joissa pyörii applikaatioita ja tietokantoja. Tietoturvallisuuden vuoksi näistä palvelimista käytetään nimitystä palvelin 1, palvelin 2 jne. Windows versiona toimii Windows Server 2012 R2.

## 3.1 Windows Server 2012 R2

Windows Server 2012 R2 on Microsoftin toiseksi uusin palvelinversio ja se on hyvin laajalti käytetty. Se julkaistiin 18.10.2013. Eroja Windows Server 2008:aan on huomattavasti ja eritoten käyttöliittymässä, joka muistuttaa kuluttajille tuttua Windows 8: aa.

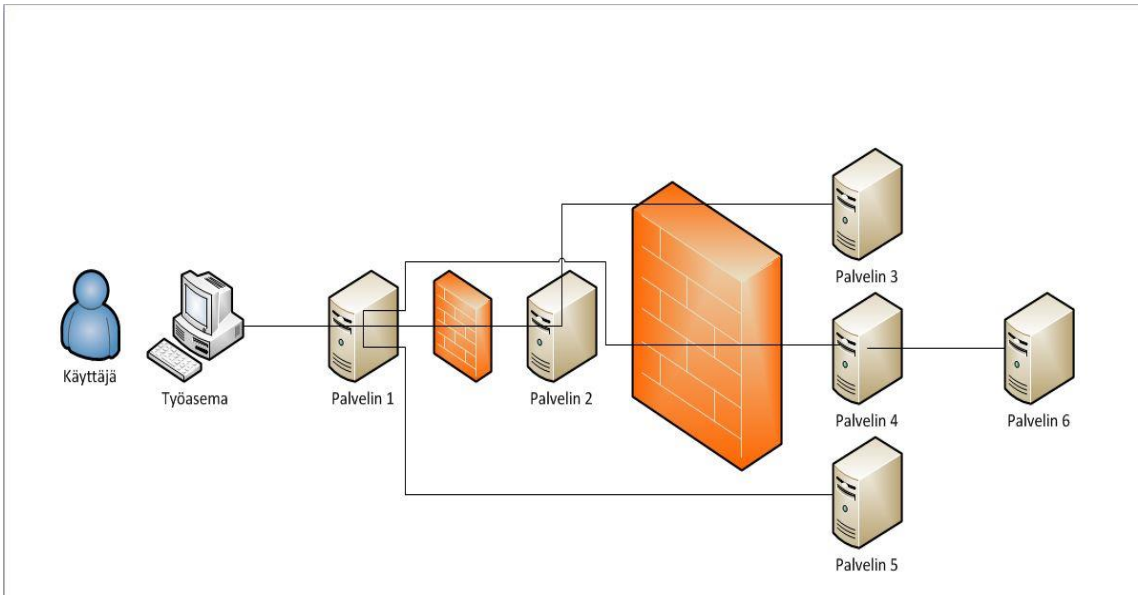



Kuvio 3: Windows Server 2012 R2 Start-valikko

Tätä käyttöliittymää kutsutaan retro-käyttöliittymäksi (kuvio 3), sillä se koostuu erivärisistä palikoista, joilla on korvattu perinteinen käynnistä-valikko. Muita lisättyjä ominaisuuksia on päivitetty Hyper-V, joka mahdollistaa paremman virtuaalisoinnin palvelimella. Lisäksi Task Manager on kokenut muodonmuutoksen ja siihen on päivittynyt graafinen ilme, sekä mahdollisuus tutkia prosesseja entistä helpommin ja tarkemmin. R2 version myötä Microsoft paransi TLS-tukea tukemaan RFC 5077. Tämä toi TLS-istuntojen jatkamisen ilman palvelinpuolen tilaa. Toisin sanoen se paransi pitkien TLS-suojattujen yhteyksien suorituskykyä, joiden täytyi uudelleen yhdistää istunnon vanhenemisen vuoksi. (Microsoft 2013)

## 3.2 Palvelinympäristön kuvaus

Kuvio 4 havainnoi ympäristön rakennetta ja niiden välisi yhteyksiä. Seuraavissa kappaleissa käyn läpi yleisluontoisesti niiden sisältöä niiltä osin, kuin tietoturvallisesti on mahdollista.




Kuvio 4: Palvelinympäristö

Ympäristö koostuu kuudesta palvelimesta ja niitä hallinnoidaan etäyhteyksillä. Palvelimet on erotettu palomuurilla siten, että palvelimet 3-6 ovat saman palomuurin takana.

### 3.2.1 Palvelin 1

Palvelin 1 toimii eräänlaisena hyppypalvelimena, johon otetaan yhteyttä käyttäjän työasemalta. Hyppypalvelimella käytetään erilaisia sovelluksia ja käyttöliittymiä. Varsinainen laskennallinen prosessi suoritetaan palvelimella 4.

### 3.2.2 Palvelin 2

Palvelimen 2 toiminnallisuus perustuu edustajana olemiseen. Tällä tarkoitetaan sitä, että palvelin hoitaa liikenteen ohjaamisen HTTP/HTTPS- liikenteessä. Palvelimelle 2 on määritelty konfiguroinnit, esimerkiksi mistä löytyy sertifikaatti ja mitä protokollaa tulee noudattaa.

Palomuuri on sijoiteltu kummallekin puolelle palvelinta ja näin ollen palvelin on DMZ-alueella eli demilitarisoitu alueella eristettynä muista verkoista segmenteistä.

### 3.2.3 Palvelin 3

Palvelin 3 huolehtii selaimen kautta käytettävistä palveluista. Palvelin 2 toimii välipalvelimena käytettäessä palvelimen 3 palveluja selaimella. Palveluiden uudelleen käynnistys tapahtuu Windowsin Service management-työkalun avulla. Sovellusten konfigurointi hoidetaan sovelluksen toimittajan toimesta. Muokkauksia voidaan tehdä suoraan config-tiedostoihin tai sovelluksen toimittajan toimittamalla työkalulla. Työkalua käytetään palvelimella 1 selaimen tai graafisen käyttöliittymän kautta.

### 3.2.4 Palvelin 4

Palvelin 4 on ympäristön aivot eli se hoitaa laskennalliset toimenpiteet ja jakaa prosessejaan eri prosesseille. Prosesseja voidaan ajastaa ja automatisoida ja niiden lokit löytyvät määrätyistä sijainneista. Palvelimella sijaitsee aineisto ja päävaranto, eikä sinne ole pääsyä kuin ylläpitäjillä ja joillakin kehittäjillä.

### 3.2.5 Palvelin 5

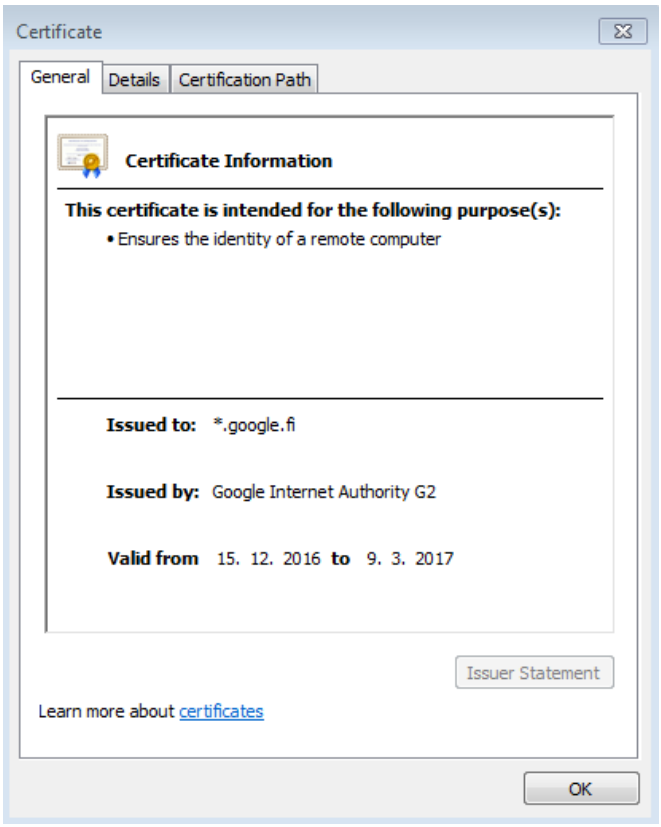
Palvelimen 5 funktio on hoitaa sovelluksen omaa metadataa ja siihen liittyvää käyttäjien varmentamista. Palvelin 5 on asetettu synkronoimaan tarvittavat AD eli Active Directory-tunnukset metadataan, jolloin käyttäjien lisääminen täytyy tehdä vain kertaalleen AD:n päässä.

### 3.2.6 Palvelin 6

Palvelin 6 toimii SQL-palvelimena ja sen tehtävä on ylläpitää SQL-kantaa. Palvelinta 6 käyttää pääasiallisesti palvelin 3 ja 4.

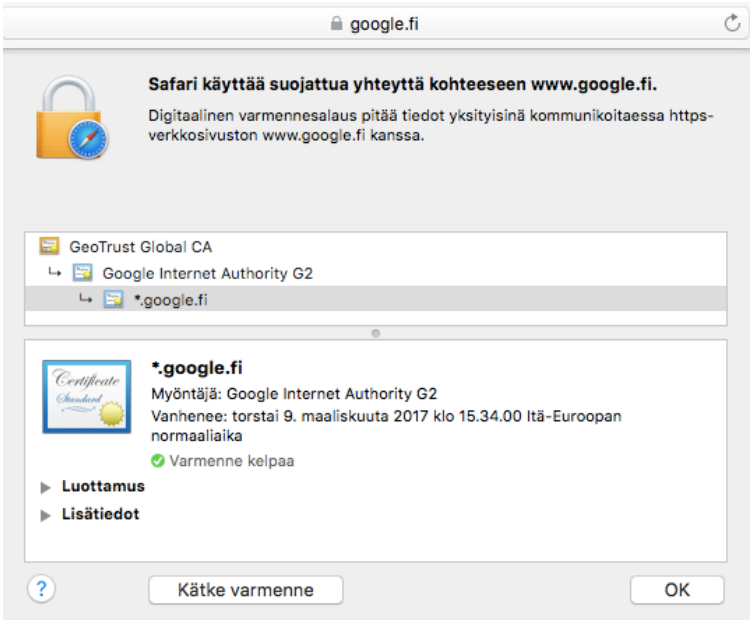
# 4 Toteutus

Koko päivitysprojektin tarve syntyi aikamääreistä ja aikamääreen saa hyvin vastaavaan tehtävään kahdesta paikkaa. Esimerkiksi tarve sertifikaatille, eli sellaista ei ollut ennestään, mutta tarkoitus on varmentaa sivuston liikenne. Tässä tapauksessa voidaan siirtyä suoraan pohtimaan kaupallisten sertifikaattien ja itse myöntämien sertifikaattien eroa ja hyötyjä. Pääasiassa itse myöntämiä sertifikaatteja käytetään yritysten sisäisissä verkoissa. Tämä kuitenkin vaatii huomattavasti enemmän työtä, sillä koko PKI-ympäristö täytyisi luoda varmenteiden ympärille, jotta ne toimisivat, eivätkä käyttäjät saisi turhia virheilmoituksia epäluotettavasta sertifikaatista.




Kuvio 5: Internet Explorerissa näkyvä esimerkkivarmenne

Toinen mahdollinen syy on sertifikaattien vanheneminen. Sertifikaatin umpeutumispäivämäärän saa selville avaamalla sertifikaatin esimerkiksi sellaisella HTTPS-sivulla, missä se on käytössä.




Kuvio 6: Safarissa näkyvä esimerkkivarmenne

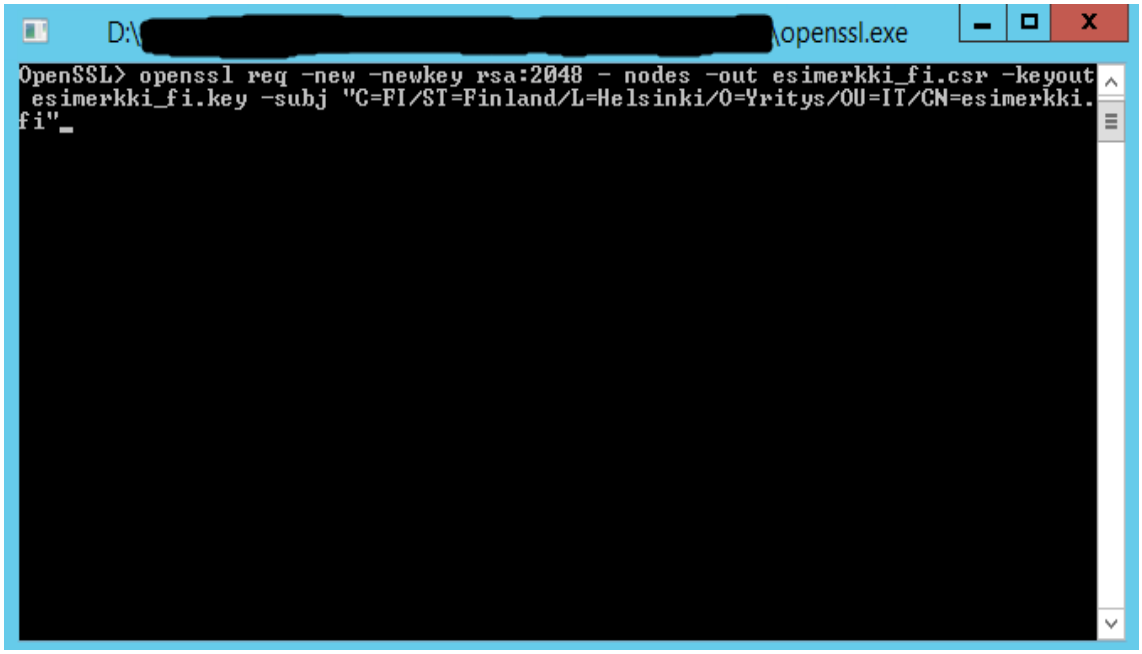
Keskitymme tässä prosessissa kaupallisten sertifikaattien hakemiseen ja niiden päivittämiseen ympäristössä. Prosessissa käytetään Symantecin sertifikaatteja.

## 4.1 Kaupallisen sertifikaatin hakeminen

Jotta voimme lähettää hakemuksen kaupallista sertifikaattia varten, tulee luoda ensin CSR-tietolomake (taulukko 2). CSR:n luomiseen voimme käyttää myöntäjän luomaa työkalua tai OpenSSL-ohjelmistoa. On suositeltavaa, että CSR luodaan palvelimella, missä sertifikaatti halutaan ottaa käyttöön. Tämä koskee erityisesti niitä tapauksia, joissa käytetään myöntäjän luomaa työkalua.
Tässä kappaleessa käydään läpi millä tavalla OpenSSL-ohjelmistolla saadaan luotua CSR. OpenSSL tulee ensin avata, jolloin avautuu Windows ympäristössä cmd-ikkuna (kuvio 7).


Kuvio 7: OpenSSL-aloitusnäkymä

Näkymä muistuttaa huomattavasti perinteistä CMD-ikkunaa. Ikkunan avauduttua voidaan ryhtyä kirjoittamaan komentoa CSR:n luomiseksi.




Kuvio 8: CSR:n luontikoodi sivustolle esimerkki.fi

Kuviossa 8 näkyvä koodi: ”openssl req -new -newkey rsa:2048 -nodes -out esimerkki_fi.csr -keyout esimerkki_fi.key -subj "/C=FI/ST=Finland/L=Helsinki/O=Yritys/OU=IT/CN=esimerkki.fi" ”. Koodin suoritettua saadaan esimerkki_fi.csr ja esimerkki_fi.key tiedostot. Näistä key päätteinen tiedosto tulee laittaa huolellisesti säilöön, sillä se sisältää salaisen avaimen, jonka avulla saadaan suojaus ja sertifikaatti murrettua. Näin ollen tämän tiedoston kadotessa tulee luoda uusi salainen avain ja tätä kautta tulee uusia myös julkinen avain. Csr-päätteinen tiedosto on tässä tapauksessa tiedosto, joka välitetään eteenpäin kaupallista sertifikaattia varten. Tätä avainta vastaan saamme sertifikaatin, joka sisältää julkisen avaimen ja sen vasta parina toimii aikaisemmin luotu salainen avain. (Globalsign 2016)

## 4.2 Sertifikaatin vastaanottaminen

Sertifikaattia anottaessa on annettu yhteystiedot, johon lähetetään suojattuna sertifikaatti ja siihen liittyvä välittäjän sertifikaatti. Tätä jälkimmäistä sertifikaattia tarvitaan harvemmin, sillä se sisältyy useimmiten valmiiksi eri järjestelmien luotettavien sertifikaattien listassa. Saatuamme haltuun lähetetyn sertifikaatin voi olla, että joudumme kääntämään sertifikaatin oikeaan muotoon tai tarkistaa sertifikaatin sopivuuden salaisen avaimen kanssa. Sertifikaatin kääntäminen PKCS#7 muodosta PEM-muotoon on hyvin yleinen operaatio, vaikka-kin kyseessä on suoraan myöntäjältä saatu sertifikaatti. Syy kääntämiselle on yksinkertainen: Apache:a varten tarvitaan sertifikaatti luettavaan muotoon. Yleensä salainen avain on jo it-sellään, mikäli CSR on luotu OpenSSL:n avulla. OpenSSL on hyvin käyttökelpoinen työkalu ser-tifikaattien päivittämisoperaatiossa. Haluamme ajaa seuraavaksi koodin, jolla käännämme PKCS#7 tiedoston muotoon PEM: ”openssl pkcs7 -in esimerkki_fi.p7b -print_certs -out esi-merkki_fi.pem”. Toinen vastaava konversio saattaa tulla vastaan, kun CSR on luotu muulla työkalulla, eikä salaista avainta erikseen eriytetä operaatiossa. Tässä tapauksessa salainen avain paketoidaan ulos saadun sertifikaatin avulla. Tässä tapauksessa saadaan palautettua paketti PKCS#12, joka sisältää sertifikaatin ja salaisen avaimen. Tämän jälkeen prosessi on samantyylinen, kuin PKCS#7 kanssa, mutta nyt haluamme salaisen avaimenkin irti tiedostosta. Seuraavilla koodeilla saamme irrotettua PKCS#12 tiedostosta sertifikaatin ja salaisen avaimen:

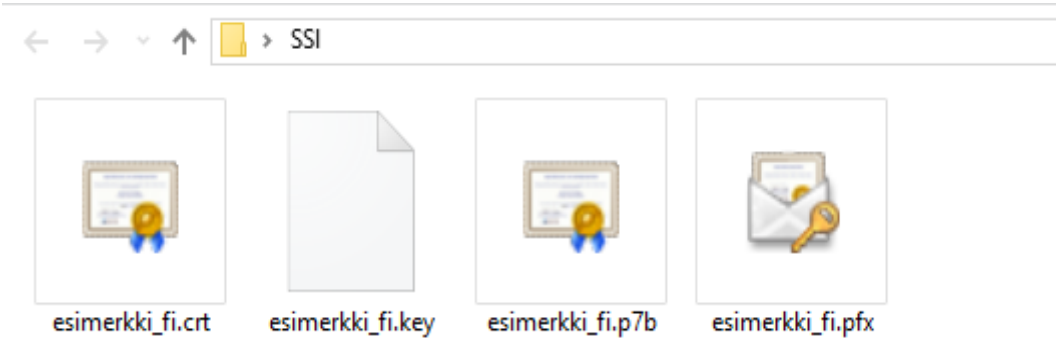
```
”   openssl pkcs12 -in esimerkki_fi.pfx -nocerts -out esimerkki_fi.key

    openssl pkcs12 -in esimerkki_fi.pfx -clcerts -nokeys -out esimerkki_fi.crt
“
```

Ylemmässä koodissa haluamme eriyttää salaisen avaimen ja määritämme sille nimen. Alemmassa koodissa eriytämme paketista sertifikaatin. Näillä toimenpiteillä olemme valmiita jatkamaan päivitystä. (OpenSSL 2016)

## 4.3 Tiedostot ja niiden nimeäminen

Seuraavaksi haluamme siirtää tiedostot oikeisiin kansioihin. Tätä varten meidän tulee selvittää missä edelliset sertifikaatit sijaitsevat. Nämä löydettyämme siirrämme luodut

tiedosto samaan polkuun. Emme halua yli kirjoittaa/tuhota edellisiä tiedostoja, vaan luomme niitä varten old_ssl-kansion, johon ne siirretään turvaan.




Kuvio 9: Tarvittavat tiedostossa oikeassa sijainnissa

Tiedostot tulee sijoittaa jokaiselle kuudelle palvelimelle identtiseen kansiosijaintiin D:\resources\ssl. Tämän lisäksi esimerkki_fi.crt ja esimerkki_fi.key tiedostot tulee sijoittaa palvelimilla 2 ja 3 sijaintiin D:\config\web\ssl. Config-tiedostossa on määritelty aikaisemmin polku sertifikaatille ja salaiselle avaimelle. Jotta näitä nimiä ei jouduttaisi muuttelemaan config-tiedostosta, nimetään tiedosto vastaavalla samalla nimellä, kuin vanhatkin tiedostot olivat. Tässäkin sijainnissa haluamme säilyttää edelliset tiedostot ja varastoida ne kansioon old_ssl.

## 4.4 Sertifikaattien päivittäminen sisäiseen kirjastoon

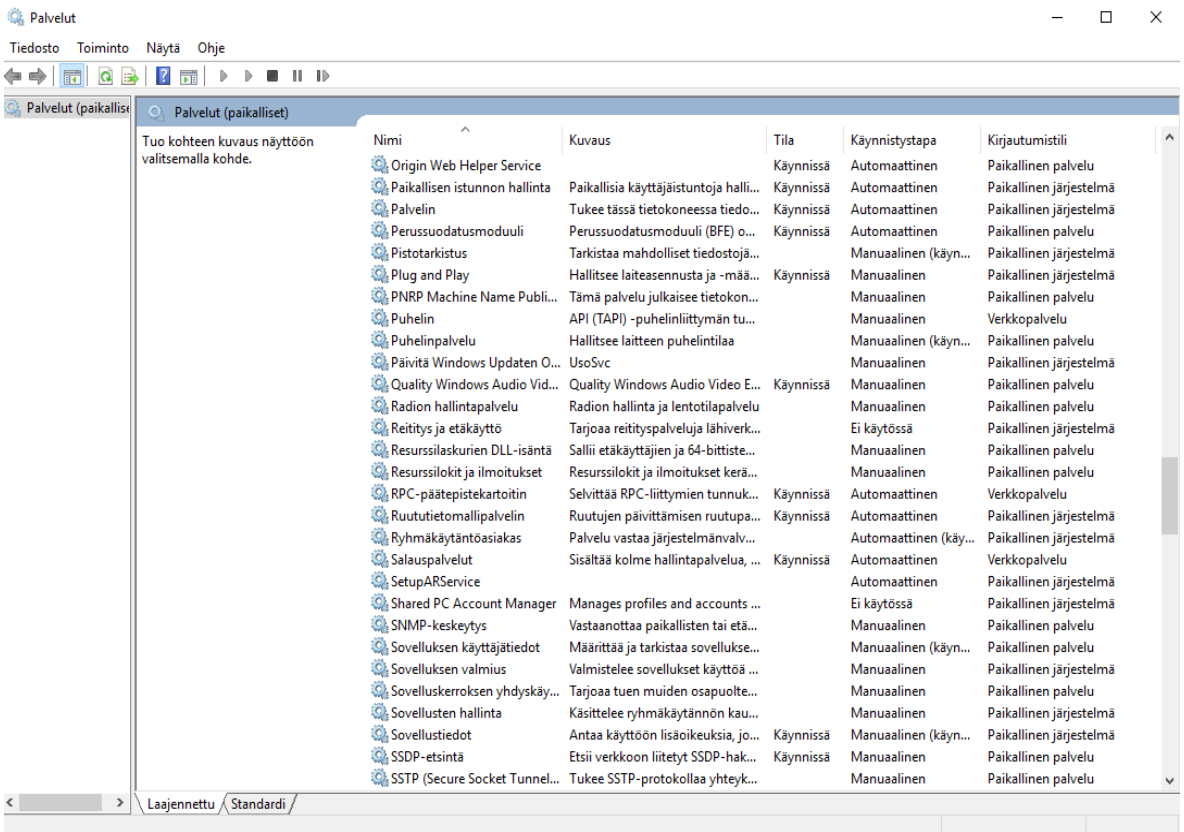
Sertifikaattien asettamisen jälkeen on vuorossa sertifikaatin lisääminen Java:n trustedstore:en. Tämä hoituu helpoiten käyttämällä Keytool-työkalua, jolla sertifikaatin lisääminen on yksinkertaista. Alussa on hyvä tarkastella kirjaston sisältämiä sertifikaatteja. Sitä varten voidaan käyttää polussa jdk\jre\lib\security komentoa: ”keytool -list -keystore cacerts”. Edellä oleva komento listaa asennetut sertifikaatit. Tarkasteltuamme sertifikaatti kirjaston nykytilan voimme lisätä uudet sertifikaattimme. Seuraavalla komennolla: ”keytool -keystore cacerts -importcert -alias esimerkki_fi -file esimerkki_fi.crt”. Tämän jälkeen tulee kysymys ”luotatko tähän sertifikaattiin?”. Tähän on mahdollisuus vastata y(yes)/n(no). Nyt, kun haluamme lisätä sertifikaatin ja olemme varmoja sen oikeellisuudesta, painamme y. Tässä vaiheessa viimeistään kysytään salasanaa cacert-kirjastoon. Oletusarvoinen salasana on ”changeit”. Näiden toimenpiteiden jälkeen voimme varmistaa sertifikaatin sisään viennin tarkistamalla kirjaston samalla komennolla, aikaisemmin tarkastelimme kirjaston sisältämiä sertifikaatteja. Tämän kappaleen toimenpiteet toistetaan jokaisella palvelimella ja niiden onnistumisen varmistaminen on tärkeää. (Oracle 2016)

## 4.5 Sertifikaattien asentaminen Windowsin työkalulla

Viimeistelläksemme päivitysprosessin tulee meidän asentaa sertifikaatti Windowsin omalla työkalulla jokaiselle palvelimelle. Tämä operaatio voidaan suorittaa avaamalla sertifikaatti sijainnista \resources\ssl. Valitsemme tiedoston päätteellä .pfx/.cer. Avattuamme sertifikaatin ilmestyy ikkuna, josta valitsemme ”Install certificate”. Aluksi ohjelma kysyy mihin kirjastoon haluamme asentaa sertifikaatit. Tässä tapauksessa käytämme vakio Windows kirjastoa. Tästä etenemme valitsemalla ”next”, kunnes asennusohjelma kysyy salasanaa. Ohjelman ilmoittaessaan ”Install Completed”, voidaan todeta asennuksen onnistuneen. (Microsoft 2016)

## 4.6 Palveluiden uudelleenkäynnistely

Palveluiden uudelleenkäynnistys on tarpeellista, jotta voimme käyttöön ottaa uudet sertifikaatit. Kyseiset palvelut koskettavat Apache-serveriä, sekä muita web-palveluita. Nämä palvelut löytyvät palvelimelta 2 ja 3. Apache:n palvelut ovat palvelimella 2 ja muut web-palvelut löytyvät palvelimelta 3. Palveluiden hallinta tapahtuu Windowsin Services-työkalulla. Työkalu sijaitsee Windowsin valvontatyökaluissa. (SAS 2016)




Kuvio 10: Havainnekuva suomenkielisestä Services-työkalusta

Paikallistettuamme halutut palvelut suljetaan ne tietyssä järjestyksessä alas. Järjestys on tärkeä erityisesti käynnistysvaiheessa, sillä palvelut rakentuvat toistensa päälle. Palveluiden uudelleenkäynnistys suoritetaan pysäyttämällä palvelu valitsemalla palvelu hiiren oikealla-korvalla ja valitsemme sieltä ”stop”. Tämän jälkeen työkalu ilmoittaa palvelun pysähtyneen. Meidän pitää pysäyttää kaikki palvelut järjestyksessä, ennen kuin käynnistämme seuraavat palvelut samaisesta valikosta, mutta tällä kertaa valitsemme ”start”. Käynnistymistä voimme seurata web-applikaatioiden osalta polusta config\web\webapps\ ja sieltä valitsemalla haluttu web-applikaation numero ja sen alta valitaan \logs ja avataan tiedosto server.txt. Tekstitiedoston loppuun ilmestyy ”service has been initialized in 123345ms”. Nyt kun tiedämme kuinka itse sammutus/käynnistys prosessi suoritetaan, voidaan ryhtyä tuumasta toimeen. (SAS 2016)

| Palvelu | Sammutus/Käynnistys |
|---|---|
| 1. Web app 4 | |
| 2. Web app 3 | |
| 3. Web app 2 | |
| 4. Web app 1 | |
| 5. Web server | |
| 6. Apache server | |

Taulukko 3: Tarkistustaulukko palveluiden käynnistämiseen

Palvelut sammutetaan aloittamalla taulukon kohdasta 1 ja edetään numerojärjestyksessä niin, että viimeisenä numero 6. Taulukkoa voidaan käyttää tarkistuslistana, jolloin siihen merkitään kunkin palvelun kohtaan ”x”/, kun palvelu on sammutettu. Vastaavasti käynnistäessä merkitään x/”x”, jolloin se kuvastaa palvelun olevan käynnissä. Palveluiden käynnistys tapahtuu käänteisessä järjestyksessä, eli kohdasta 6 kohtaan 1 edeten. (SAS 2016)

## 4.7 Tarkistaminen

Palveluiden käynnistyttyä voidaan tarkistaa sertifikaattien toimivuus. Tämä tapahtuu yksinkertaisesti ottamalla yhteyttä sivustoon, johon asetimme sertifikaatit. Avaamme selaimen ja otamme yhteyden osoitteeseen https://esimerkki.fi. Tämän jälkeen osoiterivillä pitäisi näkyä selaimesta riippuen lukon kuva ilman minkäänlaista viitettä sen virheellisyyteen. Hiirelle valitessamme lukon ikonia voimme tarkastella sertifikaattia. Nyt sertifikaatin tulisi

näyttää päivittyneen. Päivittymistä voidaan varmentaa vertailemalla myöntämispäivämäärää ja päättymispäivämäärää asennettuihin sertifikaatteihin. Mikäli ne täsmäävät on kyseessä samat sertifikaatit. Lopuksi varmistetaan, että sertifikaatissa lukee luotettu tai muu vastaava fraasi, joka indikoi sertifikaatin olevan validi ja kättelyn onnistuneen käyttäjän tietokoneen ja palvelimen välillä.

# 5 Yhteenveto

Sertifikaatit saatiin onnistuneesti päivitettyä kaikille palvelimille ja kaikki palvelut toimivat niin kuin ne toimivat ennen päivittämisoperaatiota. Sertifikaattien päivittyneisyyden voi helposti tarkistaa selaimella ottamalla yhteyttä sivustoon. Päivittämisprosessiin meni kokonaisuudessaan puolitoista tuntia sen jälkeen, kun saatiin sertifikaatit varmenteiden myöntäjältä. Tämä tarkoittaa noin 15 minuuttia palvelinta kohden, joka on kohtuullista, kun ottaa huomioon kaikki varmistelut ja monivaiheisen etenemisen. Prosessia voidaan tehostaa paljon, kunhan toimenpide tehtäisiin useammin mahdollisimman vakiossa ympäristössä. Tällöin käytetty aika toiminnalliseen osuuteen puolittuisi.

Hakemusvaiheeseen ja CSR:n luomiseen käytetty aika on noin 15 minuuttia, mikäli kaikki tarvittavat tiedot ovat valmiina käytettävissä. Prosessin aikana eteen tulevat ongelmanratkaisutilanteet voivat vaikeuttaa projektin kulkua ja täten pidentää prosessin kestoa. Tällaisia kohtia olivat muun muassa CSR:n luominen ja itse päivittämisprosessi. CSR:n luomisessa täytyi olla erityisen tarkka jo pelkästään sen luomiseen käytetyn sovelluksen suhteen, sillä se vaikutti omalta osaltaan itse prosessin kulkuun. Lisäksi käytetyt tiedot oli hyvä varmistaa moneen otteeseen ennen CSR:n luomista.

Virheellisen data koodissa voi aiheuttaa mahdollisesti suuren viivästyksen, koska siinä tapauksessa on ainoa vaihtoehto anoa uutta sertifikaattia uudella CSR:llä ja oikeilla tiedoilla täytettynä. Virhetilanne ilmenee toimimattomana sertifikaattina esimerkiksi siten, että sivusto ei avaudu, koska kättelyavaimien vaihto ei onnistu. Virhe olisi ilmennyt toimimattomana sertifikaattina esimerkiksi siten, että sivusto ei avaudu, koska kättelyavaimien vaihto ei onnistu. Päivitysprosessin aikana tapahtuneet ja havaitut virheet on helpompi korjata, mutta joka tapauksessa kasvattavat tarvittavan ajan määrää. Tällaisia virheitä olisi voinut olla esimerkiksi virheellinen konversio tai salaisen avaimen hävittäminen. Prosessissa on monia mahdollisia ongelmanratkaisua vaativia tilanteita, jotka jokainen omalta osaltaan voivat viivästyttää prosessin valmistumista. Huolellinen taustatyö ja suunnittelu omalta osaltaan tukivat prosessin onnistumista ja mahdollistivat onnistuneen prosessin luomisen.

SSL-sertifikaattien päivittäminen ja hallinnointi palvelinympäristöön on opinnäytetyönä toteutettu toimintatutkimuksena. Opinnäytetyön tulos toi uutta tietoa kokonaisprosessista ja pikaohjeen päivittämistä varten.

Mahdollisia jatkokehityksen kohteita voi olla perehtyminen itse allekirjoitettuihin sertifikaatteihin, joita varten luodaan PKI-ympäristö. Tällä tarkoitetaan ympäristöä, joka hoitaisi sisäisessä verkossa varmentamisia. Esimerkiksi sivusto ei ole liitetty www-sivujen joukkoon ja sovellukset toimivat verkon sisällä. Tällöin voidaan pohtia tarpeen mukaan kaupallisten sertifikaattien tarvetta. Jotkin sovellukset ja laitteet eivät kuitenkaan toimi aina itse allekirjoitettujen sertifikaattien kanssa, eivätkä tämän vuoksi muodosta yhteyttä.

Lähteet

A. Freier, P. Karlton, P. Kocher 2011. RFC 6101 8/2011
B. Kaliski 1998. RFC 2315 3/1998, 1-3.
E. Rescorla 2000. RFC 2818 5/2000, 2.
Eriksson, P., Kovalainen, A. 2008. Qualitative Methods in Business Research. SAGE.
J.Linn 1993. RFC 1421 2/1993, 9.
K. Moriarty, M. Nystrom, S. Parkinson, A. Rusch, M. Scott 2014. RFC 7292 7/2014, 3-6.
M. Nystrom, B. Kaliski 2000. RFC 2986 11/2000, 2-4.
T. Dierks, E. Rescorla 2008. RFC 5256 6/2008, 3-5.

Sähköiset lähteet:

B. Vangie. 2017. OSI layers. 6.1.2017. http://www.webopedia.com/quick_ref/OSI_Layers.asp
Globalsign. 2016. Install PFX/PKCS#12 in Apache web server with OpenSSL. Viitattu 20.12.2016. https://www.globalsign.com/en/support/install/install_pfx_apache.php
I. Sani 2011. Rehellisen Ahmedin autokauppa ei kelvannut Mozillalle. 20.4.2011. http://www.tivi.fi/Arkisto/2011-04-20/Rehellisen-Ahmedin-autokauppa-ei-kelvannut-Mozillalle-3184215.html
Microsoft. 2013. Windows Server 2012 R2 and Windows Server 2012. 1.11.2013. https://technet.microsoft.com/en-us/library/hh801901.aspx
Microsoft. 2016. Import a Certificate. Viitattu 15.12.2016. https://technet.microsoft.com/en-us/library/cc754489(v=ws.11).aspx
Netscape 2017. The SSL Protocol. Viitattu 9.1.2017. https://web.archive.org/web/19970614020952/http://home.netscape.com/newsref/std/SSL.html
OpenSSL. 2016. FAQ. Viitattu 12.12.2016. https://www.openssl.org/community/OpenSSL
Oracle. 2016. Configuring Java CAPS for SSL Support. Viitattu 2.12.2016. https://docs.oracle.com/cd/E19509-01/820-3503/6nf1il6er/index.html
SAS. 2016. Running Servers as Windows Services. Viitattu 2.10.2016. http://support.sas.com/documentation/cdl/en/bisag/68240/HTML/default/viewer.htm#n11hox4i2tgybvn1omal5elwlv5k.html
Symantec. 2013. Beginner's Guide to SSL Certificates. Viitattu 27.12.2016, 3-5. https://www.symantec.com/content/en/us/enterprise/white_papers/b-beginners-guide-to-ssl-certificates_WP.pdf
W3techs 2015. Usage of SSL certificate authorities for websites. 5/2015. https://w3techs.com/technologies/overview/ssl_certificate/all
Webtrust. 2011. Trust Service Principles and Criteria for Certification Authorities. Viitattu 26.12.2016, 6-7.

http://www.google.fi/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&ved=0ahUKEwi-qZL3-LnRAhXCCiwKHVjuAwwQFgggMAE&url=http%3A%2F%2Fwww.webtrust.org%2Fhomepage-documents%2Fitem54279.pdf&usg=AFQjCNF-xl9mnHTGa3ZEWY_zeBv9xwGEjg
Wikipedia 2016. Transport Layer Security. Viitattu 12.12.2016.
https://en.wikipedia.org/wiki/Transport_Layer_Security

Kuviot

Taulukot

Liitteet

Liite 1: Sertifikaattien päivittämisen ohje

Sertifikaattien päivittämisen ohje
8.1.2017

1. Sertifikaatin saamista varten tulee luoda CSR, joka lähetetään sertifikaatteja myöntävälle taholle. CSR:n voit luoda käyttämällä OpenSSL-työkalua tai sertifikaattien myöntäviltä tahoilta löytyvän ohjelman avulla. Mikäli käytät OpenSSL-työkalua, muista ottaa salainen avain talteen (.key päätteinen) ja älä lähetä sitä missään tapauksessa minnekään.

1. Saatuasi sertifikaatin myöntäjältä sertifikaatit tarkastele niiden tiedostomuotoja. Tarvitset sertifikaatit formaatissa PEM, jotta Apachen serverit pystyvät käsittelemään niitä. Tilanteesta riippuen löydät Googlen avulla paljon erilaisia ohjeita sertifikaattien kanssa toimimiseen erilaisten tiedostopäätteiden kanssa. Voit käyttää hakusanoina seuraavia:" manage certificates" tai "extract certificates from pfx to apache". Konvertointia varten voit käyttää työkalua OpenSSL.

2. Saatuasi kaikki tarvittavat tiedostot (.pfx , .key ja .cer/crt) voi sertifikaatit siirtää sijaintiin \config\web\ssl palvelimilla 2, sekä 3 ja kaikilla palvelimilla sijaintiin resources\ssl . Tätä ennen tulisi vanhat sertifikaatit siirtää pois alta kansioon \old_ssl. Uusien sertifikaattien ollessa kansiossa, kopioi edellisten sertifikaattien nimi uusiin. Näin ollen ei tarvitse vaihtaa config-tiedostoissa olevia määrityksiä. Vanhoihin tiedostoihin voi selkeytyksen vuoksi lisätä nimeen päivämäärän malliin "esimerkki_fi20161224.cer", joka kuvastaa päivämäärää jolloin sertifikaatit on päivitetty.

3. Sertifikaattien ollessa oikeissa paikoissa kaikilla palvelimilla voidaan ryhtyä viemään niitä Javan cacerts luotettujen sertifikaattien kirjastoon. Tätä varten tarvitsemme Keytool työkalua, joka vie sertifikaatit kyseiseen kirjastoon. Mikäli haluaa tarkastella kirjaston sisältöä ennen sertifikaattien vientiä, se onnistuu komennolla " keytool -list -keystore cacerts". Keytool työkalua varten täytyy määrittää sen sijainti levyllä ja sen oletus polku on jdk\jre\lib\security. Mikäli kirjastoon ei ole aikaisemmin viety myöntäjän sertifikaattia (Intermediate) tulee se sijoittaa samaiseen kirjastoon myönnetyn oman sertifikaatin kanssa. Sertifikaattien vienti onnistuu komennolla "keytool -keystore cacerts -importcert -alias esimerkki_fi -file esimerkki_fi.crt". Komennossa määritettään kirjasto mihin viedään, mistä viedään ja mikä alias sille luodaan. Alias tulisi olla mahdollisimman kuvaava ja sisältävän esimerkiksi vuoden. Näin ollen sen löytäminen kirjastosta on myöhemmin helppoa ja näin ollen homma nopeutuu. Suori-

tettua komennon se kysyy, oletko varma, että haluat lisätä/viedä sertifikaatin luotettujen kirjastoon. Tähän vastataan y(yes) tai n(no).

4. Kun kaikille palvelimille on suoritettu kohdan 4 mukaiset toimenpiteet voidaan asentaa Windows Serverin omaan sertifikaattien kantaan saatu sertifikaatti. Tämä tapahtuu yksinkertaisesti avaamalla sertifikaatti tiedosto (.pfx) ja valitsemalla aukeavasta ikkunasta ”Install Certificate”. Tämän jälkeen valitaan sijainniksi oletus sijainti ja jatketaan eteenpäin, kunnes ohjelma ilmoittaa asennuksen onnistuneen. Asennuksen yhteydessä saatetaan kysyä salasanaa, mikäli sertifikaatti on suojattu.

5. Seuraavaksi tulee käynnistää web-palvelut uudestaan. Käynnissä olevat palvelut löytyvät administrator tools-kansion alta nimellä ”palvelut/services”. Uudelleen käynnistys suoritetaan sulkemalla ensin web-applikaatiot. Web applikaatio- ja web server-palvelut löytyvät palvelimelta 3. Palveluiden pysäytys tulee aloittaa web app 4 ja edetä järjestyksessä alaspäin niin, että viimeiseksi web server. Tämän jälkeen mennään palvelimen 2 palveluihin ja sieltä painetaan restart apachen palveluille. Näiden käynnistyttyä voidaan palata palvelimelle 3, jossa voidaan ryhtyä käynnistämään palveluita käänteisessä järjestyksessä sammutukseen nähden. Palveluiden (web app) käynnistymistä tulee seurata niiden lokeista sijainnista config\web\webapp\webappx\logs\server.txt. Palvelun käynnistymisestä syntyy lokin loppuun teksti ”server initialized in time 123456ms”. Tämän jälkeen voidaan siirtyä seuraavaan palvelun käynnistämiseen.

6. Palveluiden ollessa käynnissä voidaan testata niiden toimivuus. Tämä tapahtuu yksinkertaisesti ottamalla selaimella yhteyttä sivustoon, johon sertifikaatit on asetettu. Navigoitua sivustolle voidaan osoiterivillä olevasta lukosta klikata ja tarkistaa sertifikaatin tiedot. Mikäli sertifikaatti vastaa asettamaa ja sivuston auetessa ei ilmene virheitä tai muita viitteitä ongelmiin liittyen sertifikaatteihin, voidaan todeta päivityksen onnistuneen. Mikäli ongelmia tulee, niin niitä tulee tarkistella virhesanomien avulla ja tehdä tarvittavat toimenpiteet.