

**Suojatut yhteydet ja varmenteet**Tatu Erkinjuntti, Jussi Isosomppi, Eino Kupias, Saku Kähäri

Raportti

Tietojenkäsittelyn koulutusohjelma

2018

Sisällys

[1 Esipuhe 1](#_Toc532423699)

[2 Projektin työsuunnitelma 2](#_Toc532423700)

[2.1 Aikataulu sekä suunnitellut tehtävät 2](#_Toc532423701)

[2.1.1 Projektin yleinen aikataulu 3](#_Toc532423702)

[2.1.2 Projektin suunnitellut työvaiheet 3](#_Toc532423703)

[2.2 Dokumentointi ja versionhallinta 3](#_Toc532423704)

[3 Varmenteiden tutkimus 5](#_Toc532423705)

[3.1 PKI ja julkisen avaimen salausmenetelmä 5](#_Toc532423706)

[3.2 Julisten avainten hallintajärjestelmä 5](#_Toc532423707)

[3.3 Hyödyt ja heikkoudet 5](#_Toc532423708)

[3.4 CA 6](#_Toc532423709)

[3.5 WoT 9](#_Toc532423710)

[3.6 Vertailukelpoisuus 9](#_Toc532423711)

[4 Testaus 10](#_Toc532423712)

[4.1 OCSP-viestiliikenteen häirintä 11](#_Toc532423713)

[4.2 Välimieshyökkäys vihamielistä root-sertifikaattia käyttäen 11](#_Toc532423714)

[4.3 CA / OCSP-palvelimeen kohdistuva palvelunestohyökkäys 11](#_Toc532423715)

[4.4 SSLstrip 12](#_Toc532423716)

[4.5 Ettercap / Bettercap 12](#_Toc532423717)

[4.6 Tulosten yhteenveto 13](#_Toc532423718)

[5 Ohjeistus 14](#_Toc532423719)

[5.1 Työn toteutus 14](#_Toc532423720)

[5.2 Tulokset 15](#_Toc532423721)

[6 Yhteenveto 16](#_Toc532423722)

[7 Liiteet 17](#_Toc532423723)

[7.1 Liite 1. Sertifikaattien salattu maailma 17](#_Toc532423724)

[7.2 Liite 2. Verkkovarmenteiden testausraportti 17](#_Toc532423725)

# Esipuhe

Verkkovarmenteet ovat olleet jo vuosikymmenten ajan keskeisessä osassa suojaamassa käyttäjiä verkkoliikenteessä. Ne ovat laajimmin hyväksytty ratkaisu tiedon suojaamiseksi kahden tahon välillä. Niihin sisältyy omat haasteensa ja puutteensa, mutta ne ovat tällä hetkellä ainoa helposti skaa-lautuva ratkaisu. Niiden avulla on mahdollista tunnistaa käyttäjiä ja suojata tärkeitä yhteyksiä kuten sähköposti, verkkokauppa ja -pankki liikennettä.

Tavalliselle käyttäjälle ne saattavat olla tuttuja vain käsitteen tasolla, tai mahdollisesti tämä ei tiedosta niitä millään tavalla. Selaimessa varmenteen olemassaolosta ilmoittaa selaimen osoite rivillä https-alku verkko osoitteessa ja lukon kuva vieressä. Tämä saattaa tulla ilmi vasta, kun selain ilmoittaa puutteellisesta sertifikaatista ja voi potentiaalisesti aiheuttaa verkkokaupoille isot tappiot, kun palvelu ei ole asiakkaiden tavoitettavissa.

Projektin tarkoituksena oli syventää ryhmän jäsenten ymmärrystä verkkovarmenteista, niiden toi-minnasta ja niihin liittyvistä ongelmista. Osana projektia testasimme varmenteita käytännössä eriste-tyssä verkossa, toisena osana laadimme ohjeistusta verkkovarmenteista tietämättömälle henkilölle.

# Projektin työsuunnitelma

Projektin työsuunnitelmassa ei huomioida mahdollisia töitä tai esitutkimuksia, joita suoritettiin ennen projektin virallista aloitusta.

Projektin työsuunnitelma muodostui alun suunnittelun perusteella kaksivaiheiseksi:

Ensin projektiryhmä tutki varmenteita teoriatasolla ja pyrki löytämään niistä mahdollisia heikkouksia. Koska varmenteiden käsite on laaja, varatiin tähän tutkimusvaiheeseen lähes puolet projektin työaikabudjetista. Tutkimusvaiheessa pyrittiin myös hahmottamaan kahden eri avainhallintajärjestelmän (Chain of trust ja Web of trust) eroja sekä kummankin tavan hyötyjä käyttäjän kannalta.

Projektin toisessa vaiheessa projektiryhmä jakautui kahteen eri työryhmään. Toisen ryhmän tehtävä oli valmistella peruskäyttäjille suunnattu ohjeistus verkkovarmenteiden käytöstä, toisen ryhmän taas pyrkiä testaamaan varmenteista löytyviä heikkouksia sekä käyttää niitä vihamielisessä tarkoituksessa.

Suunnitelmassa huomioitiin se, että toisessa vaiheessa suoritettavat testit sekä ohjeistuksen rakenne hahmottuisivat vasta ensimmäisen vaiheen tutkimuksen myötä.

Projektityö oli tarkoitus suorittaa pääsääntöisesti Haaga-Helian Pasilan kampuksen tiloissa, mutta projektiryhmän jäsenet luonnollisesti varautuivat myös mahdollisiin etänä suoritettaviin tutkimus- ja työtehtäviin.

## Aikataulu sekä suunnitellut tehtävät

Työn aikataulu pyrittiin pitämään realistisena, huomioiden kuitenkin mahdollisuus sen joustamiseen. Alkuperäiseen aikatauluun päädytiin sillä oletuksella, että projektin aikana esiin tulleisiin havaintoihin tai ongelmiin pystyttäisiin reagoimaan ilman merkittävää vaikutusta projektin kokonaisaikatauluun. Tarkoitusena oli, että projektin toisessa vaiheessa ryhmät voisivat tukea toistensa työn etenemistä oman työryhmänsä tuloksilla.

### Projektin yleinen aikataulu

|  |  |
| --- | --- |
| **Viikko** | **Työtehtävät** |
| vko 43 | Teorian tutkiminen |
| vko 44 | Teorian tutkiminen |
| vko 45 | Teorian tutkiminen |
| vko 46 | Testaus |
| vko 47 | Testaus |
| vko 48 | Testaus |
| vko 49 | Ohjeistuksen sekä projektin loppuraportin valmistelua. |
| vko 50 | Projektin loppuraportin esitteleminen. |

### Projektin suunnitellut työvaiheet

|  |  |  |
| --- | --- | --- |
| **Tehtävät, vaihe 1** | **Työmäärät** | **Lopputulokset** |
| Public key infrastructuren tutkiminen | 10 tuntia | Parempi tietämys PKI:staa |
| Yleisimpien salakirjoitusmenetelmien tutkiminen (PGP, X.509) | 10 tuntia | Parempi ymmärrys käytettävistä salakirjoitusmenetelmistä |
| Web of Trustin tutkiminen | 10 tuntia | Parempi tietämys WoT:sta |
| Certificate Authorityn tutkiminen | 10 tuntia | Parempi tietämys CA:sta |
| Kahden varmennemenetelmän vertailu, haitat sekä hyödyt | 15 tuntia | Varmennemenetelmien eroavaisuudet ja mahdolliset erikoisuudet |
| Ongelmakohtien tunnistaminen varmennemenetelmistä | 20 tuntia | Varmenteiden ongelmakohdat tunnistettu |

|  |  |  |
| --- | --- | --- |
| **Tehtävät, vaihe 2** | **Työmäärät** | **Lopputulokset** |
| Suoritettavien testien suunittelu | 25 tuntia | Valmis testisuunnitelma |
| Testien toteutus | 50 tuntia | Valitut testit suoritettu |
| Ohjeistuksen laatiminen | 12.5 tuntia | Valmis ohjeistus |
| Loppuraportin laatiminen | 12.5 tuntia | Valmis loppuraportti |

## Dokumentointi ja versionhallinta

Projektiryhmän jäsenille Git:n käyttäminen oli entuudestaan jo hyvin tuttua. Tämä nähtiin luontevana sekä versionhallinnan kannalta parhaimpana ratkaisuna projektidokumentoinnin yhteiseen työstämiseen. Projektia varten luotiin yksityinen Git-varasto, joihin kaikilla projektiryhmän jäsenillä oli käyttöoikeudet.

# Varmenteiden tutkimus

Koska varmenteiden teoriasta sekä niiden käytöstä on runsaasti tietoa, koettiin tutkimuksen alkuvaiheessa parhaimmaksi lähestymistavaksi projektiryhmän jäsenten itsenäinen lähdemateriaalin etsiminen. Perusajatuksena oli, että projektiryhmän jäsen arvioi lähteen sekä sen sisältämän tiedon hyödyllisyyden projektin kannalta, ja tämän jälkeen esitteli hyödylliseksi kokemansa sisällöt muille projektiryhmän jäsenille. Projektiryhmä päätti yhdessä kunkin lähteen käytöstä, ja hyväksyttyjen lähteiden työstämistä jatkettiin koko ryhmän kesken. Tällä hieman luovalla lähestymistavalla pyrittiin tehostamaan tutkimustyötä sekä käytettävien lähteiden määrää.

## PKI ja julkisen avaimen salausmenetelmä

Vaikka projektiryhmällä oli jo entuudestaan jonkin verran tietämystä varmenteista, aloitettiin tutkimus PKI:n (Public key infrastructure, julkisten avainten hallintajärjestelmä) sekä julkisen avaimen salauksen peruskäsitteistä. Nämä koettiin johdonmukaisiksi aloitusaiheiksi, joiden kautta projektiryhmä pystyi etenemään tutkimuksissaan projektin varsinaiseen aiheeseen, verkkovarmenteisiin. Tärkeäksi koettiin myös lisätiedon hankkiminen X.509 -sertifikaattien salauksista, muista PGP-salatuista avaimista sekä missä osassa ne ovat liittyen CA:n (Certificate Authority) tai WoT:n (Web of Trust) käyttöön.

## Julisten avainten hallintajärjestelmä

Johdonmukainen lähestymistapa verkkovarmenteiden tutkimuksessa oli siirtyä kahden yleisimmin käytössä olevan julkisten avainten hallintajärjestelmän (CA, WoT) tutkimukseen. Koska verkkovarmenteiden julkinen käyttö (verkkosivut) perustuu lähes yksinomaan CA:n käyttöön, keskittyi projektiryhmä ensin tähän menetelmään. WoT:n käytön tutkiminen verkkovarmenteissa muodostui haastavaksi pääasiassa siksi, että WoT:n julkinen käyttö on hyvin rajallista.

## Hyödyt ja heikkoudet

Koska tutkittujen hallintajärjestelmien käyttö poikkeaa merkittävästi toisistaan, tulee näiden osalta käsitellä niiden hyötyjä sekä haittoja erillään toisistaan, sillä ne eivät tässä yhteydessä ole suoraan vertailukelpoisia keskenään.

## CA

CA-varmenteiden käytön hyötyjä verkkopohjaisissa ratkaisuissa ei voi kiistää, vaan vakiintuneen hierarkiansa ja laajan suosionsa vuoksi niitä voidaan pitää turvallisen internet-käytön kulmakivinä. CA-varmenteet vahvistavat yksityisyyden suojaa sekä luottamusta alati vaarallisemmaksi muuttuvassa internetissä. Ne ovat olleet suuressa osassa mm. verkkokauppa- ja verkkopankkitoiminnan laajentumista. Nykyään peruskäyttäjä usein ymmärtää verkko-osoitteessa olevan pienen s-kirjaimen tai osoitepalkin ohessa olevan lukkoikonin merkityksen, ja ideaalitapauksisa osaa olla luottamasta sivustoihin, joiden varmenne on viallinen tai puuttuu kokonaan. Näin ollen voidaan sanoa, että X.509-varmenteen käyttö on tuonut internetin tietoturvallisen käytön entistä lähemmäksi peruskäyttäjän taitotasoa.

X.509-varmenteiden yleinen käyttö SSL/TLS yhteyksissä kohdistaa niihin omalta osaltaan myös uhkia. Koska digitaalinen kaupankäynti on hyvin riippuvainen varmenteiden käytöstä, ovat varmenteet luonnollisesti houkutteleva kohde häirinnälle, haavoittuvuuksien etsimiselle ja hyväksikäytölle, varsinkin rikollisuuden näkökannasta.

Tutkimuksemme aikana löysimme X.509-varmenteiden ja CA:n rakenteesta joitain ongelmakohtia, jotka tulisi niitä käyttäessä ottaa huomioon.

**Heikko salaus**

X.509-varmenne on mahdollista allekirjoittaa usealla eri algoritmillä, myös nykystandardilla arvoiden heikoilla salauksilla (SHA1, MD5). Suosituimmat selaimet (Chrome, Firefox, Safari, Edge) ovat onneksi tahollaan reagoineet tähän ongelmaan, eivätkä enää suoraan hyväksy kyseisiä varmenteita vaan antavat tästä käyttäjälle virheilmoituksen. Heikon salauksen käyttömahdollisuutta ei kuitenkaan ole kokonaan poistettu X.509-varmenteista, eli heikkoja sertifikaatteja voidaan näin edelleen luoda.

**Varmenteen tarkistus**

CA-varmenteen aitouden ja kelpoisuuden varmistus tapahtuu joko CRL:ää (Certificate Revocation List) tai OCSP-protokollaa (Online Certificate Status Protocol) käyttäen. CRL on lista ennenaikaisesti suljetuista varmenteista, jota päivitetään ajoittain ja joka päivittyy selaimeen yleensä kerran päivässä tai harvemmin. OCSP-protokolla tarkistaa varmenteen aitouden reaaliaikaisesti verkon yli sertifikaatin myöntäjän OCSP-palvelimelta. OCSP on suunniteltu korvaamaan vanhempi, hitaasti päivittyvä CRL-käytäntö (ollut vakiona käytössä mm. Firefoxin maaliskuussa 2014 julkaistusta versiosta 28 eteenpäin). Molemmissa menetelmissä on mukana digitaalinen allekirjoitus, jonka tarkoitus on estää varmenteen voimassaolotiedon peukalointi. Tämä tarkistus jättää kuitenkin auki mahdollisuuden teoreettiselle välimieshyökkäykselle, jossa käyttäjän selaimen ja OCSP-palvelimen välillä kaikki OCSP-viestit pudotetaan. Tämä estää selainta tarkistamatta varmennetta oikein, mikä puolestaan estää SSL-yhteyden muodostamisen. Toinen teoreettinen lähetymistapa tähän on toteuttaa DoS- hyökkäys (Denial of Service) OCSP-palvelinta vastaan.

**Kaikki varmenteet eivät ole saman arvoisia**

Koska CA-varmenteiden käyttö on erittäin yleistä, on niiden myyminen/myöntäminen myös kannattavaa liiketoimintaa palveluntarjoajille. Kokematon käyttäjä ei välttämättä ole tietoinen siitä, onko varmenteessa heikolla salauksella luotu allekirjoitus, tai siitä, informoiko varmenteen myöntäjä käyttäjää omassa palvelussaan tapahtuvista virhe- tai uhkatilanteista. Näitä tilanteita voivat olla esimerkiksi varmenteen allekirjoittamiseen käytetyn salausavaimen joutuminen vääriin käsiin tai varmenteen vanheneminen. Käyttäjällä ei myöskään ole usein tietoa varmenteen myöntäjän luotettavuudesta, tai historiasta tai vakavaraisuudesta. Tämä voikin käytännössä johtaa siihen, että halvalla hankitun sertifikaatin myöntäjä syystä tai toisesta lopettaa toimintansa, jossa tapauksessa asiakas jää ilman voimassa olevaa, hyväksyttyä sertifikaattia. Tämä mielessä pitäen onkin kannattavaa valita omalle toiminnalleen luotettava sertifikaatin myöntäjä.

**Välimieshyökkäys**

Yksi SSL-salauksen päätarkoituksista on estää välimieshyökkäyksen toteuttaminen verkossa. Suojattu yhteys usein ilmenee käyttäjälle vain ”https://” -merkinnästä osoitteen edessä. Tietoturva-asiantuntija Matthew Rosenfield, tunnetummin Moxie Marlinspike, esitteli vuonna 2009 ohjelman nimelta SSLstrip.

Ohjelma käyttää hyväkseen hyökkäystä nimeltä HTTPS stripping attack, joka toteutetaan välimieshyökkäyksen kautta, kaappaamalla käyttäjän selaimen ja salausta käyttävän palvelimen välinen liikenne ja kiertämällä sen salaus. Tietoliikenne hyökkääjän ja palvelimen välillä tapahtuu suojattua HTTPS-yhteyttä käyttäen, kun taas käyttäjän ja hyökkääjän välinen liikenne on HTTPS-liikenteeksi naamioitua, salaamatonta HTTP-liikennettä. Kuvaillussa skenaariossa käyttäjä on täysin tietämätön välimieshyökkäyksestä, sillä hänen silmiinsä liikenne näyttää täysin asiallisesti salatulta ongelmasta. Palvelimelle liikenne taas tulee salattuna, joten mitään outoa ei näy myöskään sen silmissä.

Tässä yhteydessä on huomion arvoista se, että välimieshyökkäyksessä hyökkääjä käyttää itse allekirjoitettua varmennetta, josta seuraa välittömästi virheilmoitus, jos vieraillaan Certificate Pinning-tekniikkaa käyttävillä verkkosivuilla.

**Certificate Pinning**

Cetificate Pinningiä käytettässä sivuston varmenne ja näin ollen myös sen julkinen avain on tallennettuna asiakaslaitteeseen tai sovellukseen valmiiksi, eli esimerkiksi kirjattu selaimen luotettuihin sertifikaatteihin. Kun käyttäjä vierailee listatulla verkkosivulla, verrataan tallennettua varmennetta ja avainta palvelun lähettämiin. Jos avaimet eivät täsmää, on syytä epäillä välimieshyökkäystä. Tämä suojaa palvelua myös tilaneessa, jossa varmenteen myöntäjän luotettavuus on kyseenalaistettu, sillä alemman tason tallennettu sertifikaatti hyväksytään, vaikka CA-sertifikaatti olisikin syystä tai toisesta kelvoton.

Koska CA-varmenteiden käyttö on hyvin yleistä ja niiden käyttöönotto on käyttäjän näkökannasta lähes täysin huomaamatonta, piilee niiden käytössä olennaisia teknologian arkipäiväisyydestä kumpuavia ongelmia, jotka tulisi huomioida.

Ensinnäkin tulee ymmärtää, että varmenteet yleisenä ja teknisenä käsitteenä ovat usein peruskäyttäjälle mystisiä. Niiden käyttöä suositellaan vahvasti, mutta perustellaan harvoin muuten kuin oman yksityisyyden turvaamisella. Turvattomasta tietoliikenteestä aiheutuu kuitenkin herkästi mittavaa vahinkoa käyttäjän itsensä lisäksi myös tämän edustamalle organisaatiolle. Tästä hyvänä esimerkkinä toimii vaikkapa verkkokaupan käyttämän verkkovarmenteen vanheneminen, joka voi aiheuttaa mittavaakin haittaa liiketoiminnalle asiakkaiden saamien turvallisuusvirheilmoitusten johdosta.

## WoT

Web of Trust on itsessään erittäin luotettava tapa hallinoida luotettuja avaimia. Tässä kyseisessä hallintatavassa luotetaan yksittäisiin avaimiin, sekä muihin näiden luottamiin avaimiin. WoT ei olekaan yleisesti käytössä verkkoselaimissa, kun vastakkaiseen, vieraaseen osapuoleen ei varsinaisesti suoraan luoteta. WoT:ssä luottamus ei siis perustu keskitettyihin ylempiin tahoihin vaan luottamussuhde on lähes henkilökohtainen kahden osapuolen välillä.

Tämä hallintatapa onkin parhaimmillaan silloin, kun kaikki osapuolet luottavat toisiinsa, kuten esimerkiksi yritysten sisäisesti käytettyjen PGP-salattujen avainten kohdalla. Voidaankin sanoa, että WoT:n vahvuus on myös sen heikkous. Kun luotettujen avainten määrä kasvaa, avainten hallinta muodostuu haasteellisemmaksi ja mahdollisen epäkelvon avaimen mukaan pääseminen muuttuu todennäköisemmäksi.

## Vertailukelpoisuus

WoT:n sekä CA:n vertailu keskenään on haastavaa, koska näiden arkkitehtuurien käyttötarkoitukset ovat hyvin erilaiset.

CA soveltuu käyttöön loistavasti silloin, kun käyttäjä ei tunne toista käyttäjää, kuten esimerkiksi selainyhteyksiä käyttäessä.

WoT:n vahvuus perustuu nimenomaan sen käyttäjien väliseen luottamukseen. Tämän toteuttaminen verkkopohjaisissa ratkaisuissa on haastavaa, ellei jopa mahdotonta, johtuen käyttäjien vaihtuvuudesta ja määrästä.

# Testaus

Projektin testausvaiheen pääasiallinen tarkoitus oli testata mahdollisia haavoittuvuuksia, jotka olivat tulleet esiin ensimmäisen vaiheen tutkimuksen aikana. Tavoiteltua lopputulosta tälle ei asetettu projektin alkuvaiheessa, sillä oli hyvin todennäköistä että löytyneet haavoittuvuudet olisi jo korjattu siihen menessä kun projektiryhmä pääsi niitä testamaan.

Testauksen toteutuksessa huomioitiin erityisesti muutamia mahdollisia rajoittavia seikkoja, jotka vaikuttivat testauskohteiden valintaan:

Ensiksi, kaiken testauksen tulisi tapahtua lain salliman rajan sisällä. Tämä edellytti sitä, että tehtävien testausten laajuus (scope) tuli selvittää erittäin tarkasti ennen testauksen suorittamista, jotta mahdollisilta rikkeiltä, rikoksilta tai häiriöiltä pystyttiin välttymään.  
  
Toisekseen, testaus tuli voida suorittaa projektin aikataulun rajoissa. Tästä syystä projektia varten ei rakennettu omaa edistynyttä testausympäristöä, johtuen siitä että sen oletettiin suoraan ylittävän projektin aikataulubudjetin rajat.

Ensimmäisen vaiheen aikana löydetyt, valitut havoittuvuudet otettiin edellä mainitut säännöt noudattaen mukaan testausvaiheeseen. Testauskohteiden valintaan vaikuttivat omalta osaltaan myös projektiryhmän jäsenten oma osaaminen sekä ennakkokäsitykset testattavien haavoittuvuuksien onnistumismahdollisuuksista.

Testauksen lähtökohtana olivat eri hyökkäysmetodit, joita voitaisiin käyttää erityisesti PK-yrityksiä kohtaan. Tämän lähestymistavan valinta perustui oletukseen siitä, ettei näissä yrityksissä tietoturvaa tai verkonvalvontaa ole huomioitu samalla tasolla kuin suuremmissa yrityksissä. Tavoitteinamme olivat yrityksen verkossa tapahtuvan liiketoiminnan häiritseminen tai yrityksen yksityisten tietojen vakoilu.

Ellei toisin mainita, kaikki testaukset suoritettiin Kali Linux -käyttöjärjestelmällä, Haaga-Helian eristetyssä laboratorioverkossa. Testauksessa valiut kohdekoneet olivat testausryhmän hallinnassa testauksen aikana, eikä haittaa oikeille käyttäjille tai heidän tiedoilleen ollut.

## OCSP-viestiliikenteen häirintä

OCSP-protokollan hyödyntäminen nousi esille tutkimuksen alkuvaiheessa. OCSP-protokollassa selain tarkistaa varmenteen aitouden erilliseltä OCSP palvelimelta. Palvelimen vastaus on joko 1 - kelvollinen, 2 - epäkelpo tai 3 - ei tietoa. Kahdessa viimeisemmässä tapauksessa varmennetta ei voida hyväksyä.

Testauksen alussa pystytiin osoittamaan WireShark-verkontutkintatyökalua käyttäen, että kohdekoneen selain sekä lähettää, että vastaanottaa runsasti OCSP-viestejä.

Testausyökaluina tässä yhteydessä käytettiin Burp Suitea sekä OWASPin Zed Attack Proxyä (ZAP).

## Välimieshyökkäys vihamielistä root-sertifikaattia käyttäen

Projektiryhmän jäsenet olivat jo aikaisemmin testanneet osaa tästä harjoitellessaan OWASP ZAPin käyttöä. Tarkoituksena on asentaa hyökkäystyökalulla luotu root-sertifikaatti kohdekoneelle, joka mahdollistaa kohdekoneen verkkoliikenteen urkkimisen. Tässä tapauksessa käytännön haaste olisi saada valtuudet varmenteen asentamiseen kohdekoneelle, jota voisi yrittää tehdä joka etänä tai fyysisesti.

Tätä ei testattu validilla CA:n myöntämällä varmenteella, sillä koimme sen olevan projektin laajuuden ulkopuolella.

## CA / OCSP-palvelimeen kohdistuva palvelunestohyökkäys

Tätä pidettiin vaihtoehtoisena menettelynä aikaisemmin suorittamaamme OCSP-testaukseen. Teoriassa jos sertifikaatteja myöntävä tai varmistava palvelin saataisiin kaaadettua, ei varmenteita voisi luoda tai tarkistaa.

Tätä ei koskaan testattu, sillä projektiryhmällä ei ollut aikataulullisia resursseja testausympäristön rankentamiseen (oma OCSP- tai CA-palvelin, testiverkkopalvelin). Palvelunestohyökkäyksen suorittaminen toiminnassa olevaa palvelinta vastaan on lainsäädännön mukaan erittäin laitonta, eikä kyseistä vaihtoehtoa koskaan harkittu.

## SSLstrip

Tutkimusvaiheessa esiin tullut hyökkäysmetodi, joka käytännössä on välimieshyökkäys. Teoriassa hyökkääjäkoneen ja kohdekoneen välinen liikenne tapahtuu salaamattomana niin, että hyökkääjä naamioi liikenteen näyttämään käyttäjän silmissä salatulta. Tämä hyökkäys edellyttää ARP-myrkytyksen suorittamista verkossa, eli hyökkääjän naamioimista verkon oletusyhdyskäytäväksi.

Testauksen kautta emme pystyneet tätä kyseistä hyökkäystä toteuttamaan, ARP myrkytys esti kaiken liikenteen kohdekoneelta ja näin olen SSLStripp hyökkäykset toimimista ei voitu osoittaa.

## Ettercap / Bettercap

Koska aikaisempi SSLStrip hyökkäystä ei saatu menestyksekkäästi onnistumaan, päätimme testata valmista frameworkia, jolla pystyisi suorittamaan saman hyökkäyksen.

Testaus kokeiltiin aluksi käyttäen Ettercat ohjelmaa, mutta kokeilujen ja tutkimuksen kautta selvisi, ettei sitä enää tänä päivänä kehitetä tai tueta. Ettercapin kehittyneempi versio, Bettercap, valikoitui loppujen lopuksi laajempaan välimieshyökkäysten testaukseen. Hyökkäysmetodi on tässä yhteydessä käytännössä täysin sama kuin edellämainitussa SSLstrip-hyökkäyksessä, mutta kehittyneemmillä versioilla työkaluista ja toimivalla käyttöliittymällä.

Ohjelman alkutestaus sujui pääsääntöisesti hyvin; ARP-myrkytys suoritettiin onnistuneesti ja liikenne saatiin ohjattua hyökkäyskoneen kautta. Teoriassa tämän jälkeen hyökkäyskoneen lähettämät ja vastaanottamat salatut viestit pitäisi voida purkaa.

Testauksessa emme kuitenkaan saaneet tätä onnistumaan. Huomasimme testauksen aikana että verkkosivustot, jotka käyttävät Certificate Pinning -funktiota, huomaavat hyökkäysyrityksen väärennetyllä varmenteella ja antavat selaimessa virheilmoituksen.

## Tulosten yhteenveto

Testauksen tulokset omalta osaltaan puhuvat CA-verkkovarmenteiden käytön puolesta. Yksikään suoritetuista testeistä ei pystynyt murtamaan kohdekoneen viestien salausta onnistuneesti. Koska verkkovarmenteita vastaan kehitetään aktiivisesti uusia hyökkäyksiä, on vastatoimien kehittäminen myös aktiivista. Tästä syytä voidaan olettaa että testauksessa käytettyjä hyökkäyksiä vastaan on kehitetty jo korjaus.

Projektin aikana ei löytynyt sopivaa hyökkäysvektoria WoT:n testaukseen, jonka takia testaus keskittyi pääsääntöisesti CA-varmenteiden testaamiseen.

# Ohjeistus

Projektin toinen osatuote oli luoda ohjeistus sertifikaateista peruskäyttäjälle. Peruskäyttäjällä tarkoitetaan henkilöä, joka saattaa tiedostaa sertifikaattien olemassaolon esimerkiksi s-kirjaimesta selaimen osoiterivillä (http:// vs. https://) tai selaimen varoitusviestistä verkkosivusta. Kohde henkilöllä ei kuitenkaan välttämättä ole sen syvällisempää tietoa mitä sertifikaatit ovat, tai miten ne toimivat.

## Työn toteutus

Ohjeistus aloitettiin luomalla raaka kehikko ohjeistukselle. Tämä sisälsi intro kappaleen julkisen avaimen järjestelmään, PKI, WoT sekä näiden kahden järjestelmän vertailu keskenään. Loppuun sunnittelimme koottavaksi tarkastuslistaa sertifikaattien käyttössä mahdillisesti ilmenevistä ongelmista arkikäytössä. Tästä esimerkkinä, käyttäjä ei työympäristössä päässyt kirjautumaan työnantajan verkkoon. Tämä johtui siitä, että BIOS:sin kello oli väärässä ajassa, minkä johdosta myöskin sertifikaatin tarkistus epäonnistui. Lopullisesta tuotteesta päätimme kuitenkin tiputtaa tämän tarkistuslistan pois kokonaan. Syynä oli, että emme löytäneet juurikaan muita ilmeisiä ja ”arkipäiväisiä” ongelmia sertifikaattien käyttöön liittyen, emmekä kokeneet että on järkevää luoda listausta vain muutamaa ongelmaa varten.

Sovimme että jaamme vielä ohjeistuksesta vastuun PKI:n ja WoT:in kappaleista kirjoittajien kesken. Sovimme myös, että jokainen suorittaa itsenäisesti tutkimukset omista vastuualueistaan sekä kirjoittaa näitä koskevat kappaleet, ja että muihin kappaleisiin palataan yhdessä. Tarkoituksena oli väitellä PKI:n ja WoT:in ominaisuuksista ja paremmuudesta ja tämän pohjalta luoda vertailu.

Saatuamme vastuu alueemme valmiiksi totesimme, että PKI:n ja WoT:in vertailu keskenään toisiaan ei ole tarkoituksenmukaista. Järjestelmät eroavat käyttökohteiltaan huomattavasti toisistaan, joten muutimme viimeisen kappaleen käsittelemään järjestelmien heikkouksia omissa ympäristöissään. Tämän ohessa loimme dokumentille myös esipuheen sekä lyhyen selityksen julkisen avaimen kryptografiasta, johon kumpiki PKI ja WoT -järjestelmät perustuvat.

Koska ohjeistus on tarkoitettu peruskäyttäjälle, ja kuitenkin aiheen luenteen takia sisältää pakostikin paljon teknistä kieltä, lisäsimme ohjeistukseen vielä sanaston siinä käytetyistä termeistä.

## Tulokset

Tuloksena tästä osaprojektista syntyi ohjeistus dokumentti, Sertifikaattien Salattu maailma.

Ohjeistus koostuu seruvaavista kappaleista:

* Esipuhe
* Sanatoa
* Julkisen avaimen kryptografia
* Mikä on PKI?
* PGP ja WoT
* Mitä ongelmia niissä on?

Mielestämme ohjeistuksesta tuli tarkoituksenmukainen, eli se avaa sertifikaatien maailmaa peruskäyttäjälle.

Olemme koe luetuttaneet ohjeistuksen yhdellä henkilöllä, joka tunsi sertifikaattien käsitteen ennalta töiden puolesta, kun yrityksen sertifikaatti oli vanhentunut. Kuitenkaan syvällistä tietoa henkilöllä ei näiden toiminnasta ennalta ollut.

Tämän koehenkilön mukaan ohjeistus avasi aihetta hyvin ja oli selkeä ja helppolukuinen.

# Yhteenveto

Projekti alkoi innokkaissa merkeissä, emmekä tunteneet aihealuettamme niin hyvin, että olisimme voineet alusta lähtien määritellä projektin kulkua tarkasti. Niinpä oppimisprosessin myötä myös projektin tavoitteet ja toimintatavat mukautuivat tilanteeseen. Tämä toi mukanaan omat haasteensa, joihin ryhmämme onnistuikin vastaamaan kiitettävästi. Pysähdyimme useaan otteeseen ajattelemaan, että missä menemme ja mikä olisi seuraava lähestymistapa. Se, että kykenimme seisahtumaan ja kyseenalaistamaan työskentelymenetelmämme, johti siihen, että projektimme muodostui sellaiseksi kuin se tällä hetkellä on. Jokainen ryhmämme jäsen on samaa mieltä, että ryhmän jakautuminen ohjeistuksen ja testaamisen pariin oli oikea ratkaisu.

Ohjeistuksen rakenne muokkautui hyvin myöhäiseen vaiheeseen asti oppimamme mukaan. Lopputulos oli mielestämme selkeä dokumentaatio, joka pyrkii selittämään vaikeita aiheita yksinkertaisella tavalla.

# Liiteet

## Liite 1. Sertifikaattien salattu maailma

## Liite 2. Verkkovarmenteiden testausraportti