

**Verkkovarmenteet yrityskäytössä**

Tatu Erkinjuntti, Jussi Isosomppi, Eino Kupias, Saku Kähäri

Tietoturvan hallinta -kurssi

Tietojenkäsittelyn koulutusohjelma

11.9.2017

|  |  |
| --- | --- |
| **Tekijä(t)**  Tatu Erkinjuntti, Jussi Isosomppi, Eino Kupias, Saku Kähäri | |
| **Koulutusohjelma** | |
| **Raportin nimi** Verkkovarmenteet yrityskäytössä | **Sivu- ja liitesivumäärä**  19 + 2 |
|  | |
| **Asiasanat**  SSL/TLS, Sertifikaatti, Verkkoselain, | |

Sisällys

[1 Projektin työsuunnitelma 2](#__RefHeading___Toc2467_294765894)

[1.1 Aikataulu sekä suunnitellut tehtävät 3](#__RefHeading___Toc2469_294765894)

[1.1.1 Projektin yleinen aikataulu 3](#__RefHeading___Toc2471_294765894)

[1.1.2 Projektin suunnitellut työvaiheet 3](#__RefHeading___Toc2473_294765894)

[1.2 Dokumentointi sekä versionhallinta 4](#__RefHeading___Toc2475_294765894)

[2 Varmenteiden tutkimus 5](#__RefHeading___Toc2477_294765894)

[2.1 PKI ja julkisen avaimen salausmenetelmä 5](#__RefHeading___Toc2479_294765894)

[2.2 Julkisten avainten hallintajärjestelmät 5](#__RefHeading___Toc2481_294765894)

[2.3 Hyödyt sekä heikkoudet 5](#__RefHeading___Toc2483_294765894)

[1.1.1 CA 6](#__RefHeading___Toc662_633604879)

[1.1.1 WoT 8](#__RefHeading___Toc664_633604879)

[1.1.2 Voiko näitä vertailla 8](#__RefHeading___Toc666_633604879)

[3 Testaus 9](#__RefHeading___Toc2485_294765894)

[3.1 OCSP 9](#__RefHeading___Toc2487_294765894)

[3.1.1 Muutos 9](#__RefHeading___Toc2489_294765894)

[3.1.2 Blokkaus 9](#__RefHeading___Toc2491_294765894)

[3.2 Root cert asennus 10](#__RefHeading___Toc2493_294765894)

[3.3 CA / OCSP palvelimen palvelunesto hyökkäys 10](#__RefHeading___Toc2495_294765894)

[3.4 SSLStrip 10](#__RefHeading___Toc2497_294765894)

[3.5 Ettercap / bettercap 10](#__RefHeading___Toc2499_294765894)

[3.6 Tulosten yhteenveto 11](#__RefHeading___Toc2501_294765894)

[4 Ohjeistus 12](#__RefHeading___Toc2503_294765894)

[4.1 Kenelle miksi 12](#__RefHeading___Toc2505_294765894)

[4.2 Koonti / tarina kehityksestä 12](#__RefHeading___Toc2507_294765894)

[5 Suositukset varmenteiden käyttöön ( vai meneekö yhteenvetoon?) 13](#__RefHeading___Toc2509_294765894)

[6 Yhteenveto 14](#__RefHeading___Toc2511_294765894)

[Lähteet 15](#__RefHeading___Toc2513_294765894)

[Liitteet 16](#__RefHeading___Toc2515_294765894)

[Liite 1: Tuotekortti: Tietoturvan peruskartoitus 16](#__RefHeading___Toc2517_294765894)

[Liite 2. Kartoituskysymykset vastauksineen 17](#__RefHeading___Toc2519_294765894)

# 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 sekä 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.

## Julkisten avainten hallintajärjestelmät

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 sekä 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.

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

vaiheet

- Tiedonhaku: Aiheen kehykset olivat entuudestaan tuttuja. Lähdettiin liikkeelle tutustumalla kokonaisuuteen lukemalla ja katselemalla videoita.

- Aiheen valinta, työn jakaminen: Kokonaisuuden tultua hieman selkeämmäksi, lähdettiin etsimään spesifimpää tietoa heikkouksista, ongelmista ja käyttötarkoituksista. Näiden puute pakotti meidät erilaiseen projektiin, kuin mitä olimme alunperin tavoitelleet.

- Luotiin rautalankamalli ohjeistuksen rungosta, ja alettiin kirjoittamaan valituista aihealueista.

- Ensimmäisenä keskityttiin omien osuuksien teoriapuoleen ja niiden yhdistelemistä ohjeistuspohjaan.

- Ohjeistuksen projektiin liittyvän osuuden kirjoitus

- valmis ohjeistus

haasteet / ongelmat

- Ryhmätyön aiheen valinta

- Olettamukset aiheeseen liittyen

- Lähteiden valinta

- Aihealueen / lähestymistavan valitseminen

- Kirjoitetaan rinnakkain WoT ja CA-mallista, mutta niitä ei oikein voi vertailla, niin kuin alun perin olimme suunnitelleet.

- Eheän kokonaisuuden luonti

--------------------------------------------------------

Vaihe: Tiedonhaku

Termit kuten sertifikaatit ja avaimet olivat ryhmällemme entuudestaan tuttuja, mutta kokonaisuus oli tuntematon. Yksi hyvä keino oli mielestämme tutustua aiheeseen videoiden avulla. Koko ryhmämme työskenteli aluksi tiedonhaun parissa, ja PKI (Public Key Infrastructure) oli käsite, josta oli luonnollista aloittaa.

Ongelma: Mitä?

Aiheesta oppiminen toi mukanaan myös kysymyksiä. Mm. sen, että mikä projektimme varsinainen aihe on? Lukemamme ei välttämättä vastannut odotuksia, ja käytimme pitkän aikaa aiheen tai lähestymistavan valinnassa. Lukiessamme aiheesta lisää, kävi selväksi, että heikkouksista ei löydy varsinkaan ajankohtaista tietoa. Tämä pakotti ryhmämme muuttamaan lähestymistapaansa.

Vaihe: Aiheen valinta ja työtaakan jakaminen

Ryhmämme ei kokenut hedelmälliseksi jatkaa työskentelymallia, jossa koko projektiryhmä pyrkii etsimään haavoittuvuuksia. Tämän johdosta jaoimme projektiryhmämme kahtia: E ja S lähtivät suunnittelemaan ohjeistusta, kun taas J ja T ottivat oman osuutensa.

Ohjeistuksen aloitimme luomalla alustavan kehyksen projektin alkuvaiheiden keskustelujen perusteella. Tähän sisältyi intro julkisen avaimen infrastruktuurista, PKI, WoT ja miten ne vertautuvat toisiinsa. Teimme selkeän työjaon: Saku tutki PKI:ta ja Eino WoT:ia. Teimme nämä osuudet ensin valmiiksi, ja sen jälkeen yhteisesti intron ja outron yhteisten keskusteluiden perusteella.

Ongelma: Hidas alku, mielenkiinto projektiin laski. Asiaa ei myöskään helpottanut, että monialaprojekti tapahtui rinnalla.

Vaihe: Rautalankamalli ohjeistuksen rungosta.

Keskustelimme koko ryhmän kanssa siitä, mistä ohjeistus voisi koostua. Valmiita malleja tämänkaltaiselle dokumentille ei juurikaan löytynyt, joten päälinjojen luominen jäi puhtaasti ryhmämme vastuulle.

Haaste: Ei vertailukohtia

Vaihe: Rautalankamallin täyttäminen.

Luotuamme ohjeistuksen pääpisteet, työskentelymme helpottui huomattavasti, koska tiedonhaulle syntyi konkreettinen tavoite. Google Scholar oli hyvä lähde materiaalin etsintään, koska normaalit hakutulokset toivat huomattavan paljon tuloksia, joilla oli kaupallisia sidoksia, varsinkin CA-mallien osalta.

Ohjeistuksen kirjoittaminen jäi Einon ja Sakun vastuulle, ja he jakoivat tämän aiheen vielä kahteen osaa, joten tutkimus- ja kirjoitustyö tapahtui pää-asiallisesti yksilötyönä.

Haaste: Ohjeistuksen osalta relevantin materiaalin valitseminen. Lähteiden luotettavuus. Hankalan aihekokonaisuuden avaaminen helposti ymmärrettävään muotoon. Eri aiheiden sovittaminen yhtenäiseksi ohjeistukseksi.

Vaihe: Varsinaisen version luonti

Saatuamme ohjeistukseen tarpeeksi sisältöä, lähdimme muotoilemaan siitä yhtenäisempää versiota, joka sisälsi tietosisällön lisäksi myös projektiin liittyviä osa-alueita. Sisällön luettaminen muun projektiryhmän kanssa johti mm. toteamukseen siitä, että ohjeistuksen aiheita (WoT vs. CA-mallit) ei kannata vertailla.

# Suositukset varmenteiden käyttöön ( vai meneekö yhteenvetoon?)

Yhteenvedon voit halutessasi myös jättää pois.

# Yhteenveto

# Lähteet

# Liitteet

## Liite 1: Tuotekortti: Tietoturvan peruskartoitus

## Liite 2. Kartoituskysymykset vastauksineen