

**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

Työsuunnitelma

Projektin työsuunnitelmaan ei otettu huomioon mahdollisia töitä, tai esitutkimuksia joita suoritettiin ennen projektin aloittamista.

Projektin työsuunnitelma muodostui kaksi vaiheiseksi. Ensin projektiryhmä tutki varmenteiden teoriaa sekä pyrki löytämään niistä mahdollisia heikkouksia. Koska varmenteiden käsite on laaja, varatiin tähän lähes puolet projektin työaikabudjetista. Tämän kautta myös pyrittiin hahmottamaan kahden eri avainhallinta järjestelmän (Chain of trust, Web of trust) eroja sekä hyötyjä niiden käyttäjille.

Projektin toisessa vaiheessa projektiryhmä jakautui kahteen eri työryhmään. Yksi ryhmä suunnittelisi ja kirjoittaisi peruskäyttäjille suunnatun ohjeistuksen verkkovarmenteiden käytöstä, toisen ryhmän tarkoitus oli pyrkiä testaamaan sekä mahdollisesti hyötykäyttämään varmenteiden heikkouksia.

Suunnitelmassa huomioitiin se seikka, että mahdolliset suoritettavat testit sekä ohjeistuksen rakenne hahmottuisi vasta ensimmäisen vaiheen tutkimuksen aikana.

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

## Aikataulu sekä suunnitellut tehtävät

Työn aikataulu pyrittiin pitämään realistisena, mutta huomioiden mahdollisuus sen joustamiseen. Tähän ratkaisuun päädytiin sillä oletuksella että joko ensimmäisen, tai toisen vaiheen aikana esiin tulleisiin havaintoihin tai ongelmiin pystyttäisiin reagoimaan, ilman merkittävää vaikutusta projektin kokonaisaikatauluun. Tarkoitus 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 infrastructure tutkiminen | 10 tuntia | parempi tietämys PKI:a |
| Käytettävien salakirjoitusmenetelmien tutkiminen (PGP, X.509) | 10 tuntia | Parempi ymmärrys käytettävistä salakirjoitusmenetelmistä |
| Web of Trust tutkiminen | 10 tuntia | parempi tietämys WOT:a |
| Certificate authorities tutkiminen | 10 tuntia | parempi tietämys CA:a |
| Kahden varmennemenetelmän vertailu, haitat sekä hyödyt | 15 tuntia | varmennemenetelmien haitat sekä hyödyt selvitetty |
| Ongelmakohtien tunnistaminen varmennemenetelmistä | 20 tuntia | Varmenteiden ongelmakohdat tunnistettu |

| **Tehtävät, vaihe 2** | **Työmäärät** | **Lopputulokset** |
| --- | --- | --- |
| Suoritettaviena testien suunittelu | 25 tuntia | valmis testi suunnitelma |
| Testien toteutus | 50 tuntia | valitut testit suoritettu |
| Oheistuksen laatiminen | 12.5 tuntia | Valmis ohjeitus |
| loppuraportin laatiminen | 12.5 tuntia | valmis loppuraportti |

## Dokumentointi sekä versionhallinta

Projektityhmän jäsenille oli Git:n käyttäminen entuudestaan jo hyvin tuttua. Tämä nähtiin luontevana sekä versionhallinnan kannalta parhaimpana ratkaisuna projektiin liittyvän dokumentoinnin jakamiseen. Projekti varten luotiin yksityinen Git repository, joihin kaikilla projektinjäsenillä oli käyttöoikeudet.

# Varmenteiden tutkimus

Koska varmenteiden teorista sekä niiden käytöstä on runsaasti tietoa, koettiin tutkimuksen alkuvaiheessa parhaimmaksi lähestymistavaksi projektiryhmän jäsenten itsenäinen lähteinen etsiminen. Perusajatuksena oli että projektiryhmän jäsen arvioi lähteen sekä sen sisältämän tiedon projektin kannalta, ja vasta tämän jälkeen esitteli sen muille projektiryhmän jäsenille. Projektiryhmä päätti yhdessä lähteen käyttöön ottamisesta, hyväksytyt lähteet otettiin kaikkien projektiryhmän jäsenten luettavaksi.

Tällä, hieman luovalla lähestymistavalla pyrittiin tehostamaan tutkimustyötä sekä käytettävien lähteiden määrää.

## PKI ja julkisen avaimen salausmenetelmä

Vaikka projektiryhmällä oli jo entuudestaan peruskäsitys varmenteista, aloitettiin tutkimus PKI:n (Public key infrastructure, julkisten avainten hallintajärjestelmä) sekä julkisen avaimen salauksen peruskäsitteistä. Tämä nähtiin johdonmukaisena aloitusaiheelta, jonka kautta projektiryhmä pystyi etenemään tutkimuksissaan projektin varsinaiseen aiheeseen, verkkovarmenteisiin. Tärkeäksi nähtiin myös tarve oppia enemmän .x509:n varmenteesta, PGP salatuista avaimista sekä missä osassa ne ovat liittyen CA (Certificate authorities) tai WoT (Web of Trust) käyttöön.

## Julkisten avainten hallintajärjestelmät

Johdonmukainen jatkumo verkkovarmenteiden tutkimuksessa, oli siirtyä kahden yleisimmin käyttössä olevan julkisten avainten hallintajärjestelmän (CA, WoT) tutkimukseen.

Koska verkkovarmenteiden julkinen käyttö (verkkosivut) perustuu lähes yksinomaan CA:n käyttöön, keskittyi projektiryhmä ensin tähän menetelmään.

WoT käytön tutkiminen verkkovarmenteissa muodostui haastavaksi. Tämä johtui pääasiassa siitä, 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 yksitellen, ennen kuin ne ovat vertailukelpoisia keskenään.

### CA

CA varmenteiden käytön hyötyjä verkkopohjaisissa ratkaisuissa ei voi kiistää, ne luovat yksityisyyden suojaa sekä luottamusta alati vaarallisemmaksi muuttuvassa internetissä. Ne ovat mahdollistaneet verkkokauppa toiminnan luottamuksellisuuden sekä luonut tälle toimialalle merkittävää kasvua. Nykyään peruskäyttäjä useimmiten ymmärtää pienen ”s” kirjaimen, tai salauksesta kertovan lukko ikonin merkityksen verkko-osoitteessa ja ideaalitapauksisa osaa olla luottamasta sivustuihin, joilta verkkovarmenne puuttuu, tai varmenteessa on ongelma. Voi näin ollen sanoa että .x509 varmenteen käyttö on tuonut internetin käytön entistä lähemmäksi peruskäyttäjää.

.x509 varmenteiden yleinen käyttö SSL/TLS yhteyksissä kohdistaa siihen omalta osaltaan myös uhkia. Koska digitaalinen kaupankäynti on hyvin riippuvainen näiden käytöstä, yritykset mahdollisten heikkouksien löytämiseen sekä hyväksikäyttämiseen on äärimmäisen houkutteleva, varsinkin rikollisuuden näkökannasta.

Tutkimuksen aikana .x509 varmenteen tai CA:n rakenteesta löytyi ongelmakohtia, jotka tulee ottaa huomioon.

**Heikko salaus**

.x509 varmenne on mahdollista allekirjoittaa myös heikolla salauksella (SHA1, MD5). Tähän on suurimmat selaimet (Chrome, Firefox, Safari, Edge) jo tähän päivään mennessä reagoineet ja ne eivät enää suoraan hyväksy kyseisiä varmenteita ja antavat tästä virheilmoituksen. Tätä ominaisuutta ei ole poistettu .x509 itsestään.

**Varmenteen tarkistus**

Varmenteen tarkistus CA:a tapahtuu käyttäen joko CRL (Certificate revocation list) tai OCSP ( Online Certificate Status Protocol) protokolaa käyttäen. CRL on lista ennenaikaisesti suljetuista varmenteista, sitä päivitettään ajoittain, OCSP protokola tarkastaa varmenteen aitouden realiaikaisesti verkosta OCP palvelimelta. OCSP:n on tarkoitus tulevaisuudessa korvata vanhempi CRL menettely (Ainakin Firefox versiosta 28 eteenpäin). Molemmissa menetelmissä on mukana digitaalinen allekirjoitus, jonka tarkoitus on estää varmenteen peukalointi. Tämä jättää auki mahdollisuuden teoreettiselle välimieshyökkäykselle, jossa selaimen ja OCSP palvelimen välillä kaikki OCSP paketit pudotetaan. Tämä estää selainta tarkistamatta varmennetta, mikä estää SSL yhteyden muodostamisen. Toinen teoreettinen lähetymistapa tähän on toteuttaa DoS (Denial of service) hyökkäys OCSP palvelinta vastaan.

**Kaikki varmenteet ei ole saman arvoisia.**

Koska CA varmenteiden käyttö on yleistä, on niiden myyminen/myöntäminen myös kannattavaa liiketoimintaa. Kokematon käyttäjä ei välttämättä ole tietoinen jos varmenteessa on heikolla salauksella luotu allekirjoitus, tai informoiko varmenteen myöntäjä käyttäjää niissä tilanteissa kun heidän osaltaan on ilmennyt ongelma. Näitä tilanteita voi olla varmenteen allekirjoittaman avaimen vaarantuminen tai varmenteen vanheneminen. CA rootcert luottamus?

**Välimieshyökkäys**

SSL salauksen yksi tarkoitus, on ollut estää välimieshyökkäyksen toteuttaminen verkossa. Suojattu yhteys usein ilmenee käyttäjälle vain ”HTTPS” merkinnästä osoitteen edessä. Moxie Marlinspike, tunnettu tietoturva-asiantuntija esitteli vuonna 2009 ohjelman nimelta SSLStrip. Ohjelma käyttää hyväkseen hyökkäystä nimeltä HTTPS stripping attack. Tämä toteutetaan välimieshyökkäyksen kautta, missä käyttäjän sekä palvelimen väliin tunkeutunut hyökkääjä kaapaa sen läpi menevän liikenteen. Liikenne hyökkääjän sekä palvelimen välillä tapahtuu suojattua HTTPS yhteyttä käyttäen, mutta käyttäjän ja hyökkääjän välinen liikenne on HTTPS liikenteeksi naamioitua HTTP liikennettä, mikä on täysin salaamatonta. Tässä kyseisessä skenariossa, käyttäjä on täysin tietämätön välimieshyökkäyksestä, koska mikään hänen laitteella ei indikoi ongelmasta.

Koska CA varmenteiden käyttö on hyvin yleistä ja niiden käyttöönotto on käyttäjän näkökannasta lähes huomaamatonta sekä kyseenalaistamatonta, piilee niiden arkipäiväisyydessä käyttäjän kannalta olennaisia ongelmia mitä tulee huomioida.

Ensiksi tulee ymmärtää, että varmenteet yleisenä- sekä teknisenäkäsitteenä on usein peruskäyttäjälle lähes mystinen. Niiden käyttöä suositellaan vahvasti, mutta harva osaa perustella sitä muilla väitteillä kuin oman yksityisyyden turvaaminen. Tästä johtuen mahdolliset ongelmat verkkovarmenteiden käytössä saattaa kostautua käyttäjän kautta myös palvelun tarjoajalle. Tästä esimerkkinä verkkokaupan varmenteen vanheneminen, joka johtaa siihen että liiketoiminta lakkaa siihen asti että uusi varmenne otetaan käyttöön.

### WoT

Web of Trust on itsessään erittäin luotettava tapa hallinoida luotettuja avaimia. Tässä kyseissä hallintatavassa piilee myös syy miksi WoT ei ole yleisesti käytössä verkkoselaimissa. Luottamus ei perustu keskitettyihin tahoihin vaan luottamussuhde on lähes henkilökohtainen kahden osapuolen välillä.

Tämä hallinta tapa on tehokas silloin kun kaikki osapuolet luottavat toisiinsa, tästä esimerkkinä yritysten sisäisesti käytetyt PGP salatut avaimet. Voisi sanoa että WoT vahvuus on myös sen heikkous. Kun luottamuksenverkko kasvaa, avainten hallinta muodostuu haasteellisemmaksi.

### Voiko näitä vertailla

WoT:n sekä CA:n vertailu keskenään on haastavaa, koska niiden käyttäjä arkkitehtuurien tarpeet ovat hyvin erillaiset.

CA soveltuu loistavasti silloin kun käyttäjä ei tunne vastaanottajaa, tämä tulee parhaiten esille selainyhteyksiä käyttäessä.

WoT 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, mitkä olivat tulleet esiin ensimmäisen vaiheen tutkimuksen aikana. Tavoiteltua lopputulosta tälle ei asetettu projektin alkuvaiheessa, johtuen siitä että oli hyvin todennäköistä että löytyneet haavoittuvuudet olivat jo korjattu siihen menessä kun projektiryhmä pääsi niitä testamaan.

Testauksen toteutuksessa huomioitiin mahdollisia rajoittavia seikkoja, jotka vaikuttivat testauskohteiden valintaan.

Ensiksi, kaiken testauksen tulisi tapahtua lain salliman rajan sisällä. Tämän edellytti että tehtävien testausten scope tuli selvittää, 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 testausympäristö, johtuen siitä että sen nähtiin ylittävän projektin aikataulubudjetin rajat.

Mahdolliset, löytyneet havoittuvuudet otettiin valikoiden, edellä mainittuja sääntöjä noudattaen mukaan testausvaiheeseen.

Valintaan vaikutti osaltaan myös projektiryhmän jäsenten oma osaaminen, sekä ennakkokäsitys testattavan haavoittuvuuden onnistumis mahdollisuuksista.

Ellei toisin mainita, kaikki testaukset suoritettiin KaLi Linux käyttöjärjestelmällä.

## OCSP

OCSP protokolan hyödyntäminen nousi esille tutkimuksen alkuvaiheessa, selain tarkistaa varmenteen aitouden erilliseltä OCSP palvelimelta. Paluu vastaus on joko 1) validi, 2) ei validi tai 3) ei tietoa.

Kahdessa viimeisemmässä tapauksessa varmennetta ei voida hyväksyä.

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

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

### Muutos

Ensimmäisen testin tarkoitus oli tutkia mahdollisuutta muutta selaimen vastaantottaman OCSP viestin sisältöä, ajatuksena muuttaa varmenteen tarkastuksen arvoa. Tätä ei ikävä kyllä voitu toteuttaa, johtuen viestin allekirjoitetusta sisällöstä sekä aikakoodista, mikä johti siihen että vastaanottaja olisi ne hylännyt.

### Blokkaus

Toinen lähetymistapa oli estää OCSP viestien lähettäminen kokonaisuudessaan. Idea oli tarjota vaihtoehtoinen lähetymistapa traditionaalisempaan palvelunestohyökkäyksen, johtuen siitä että jos selainvarmennetta ei saisi varmistettua, ei käyttäjä vierailisi sivustolla. Vaikka saimme OCSP viestit estettyä käyttämällä Burp suite:a, ei sillä ollut vaikutusta selaimen toimintaan.

Projektin loppuvaiheessa nousi esille mahdollisuus että proxyn oma varmenne on välimieshyökkäyksessä saattanut vaikuttaa tähän. OSCP viestit on pudotettu hyökkääjäkoneen sekä palvelimen välillä, mutta käyttäjälle kaikki on näkynyt normaalina, johtuen proxyn varmenteesta.

## Root cert asennus

Tätä on projektiryhmä jäsenet jo aikaisemmin testannut harjoitellessaan OWASP zap:n käyttöä. Ajatus on asentaa itse luotu root varmenne kohdekoneelle, tämä mahdollistaa sen että selain hyväksyy proxyn käyttämän varmenteen ongelmitta, näin päästäisiin käsiksi kohdekoneesta lähtevään ja saapuvaan verkkoliikenteeseen. Skenariossa käytännön haaste tässä olisi saada valtuudet varmenteen asentamiseen kohdekoneelle, tämän voisi tehdä joko etänä tai fyysisesti.

Tätä testattua validilla CA:n myöntämällä varmenteella.

## CA / OCSP palvelimen palvelunesto hyökkäys

Tämä nähtiin vaihtoehtoisena menettelynä aikaisemmin suorittamaamme OCSP testaukseen. Teoriassa jos varmennupalvelin saataisiin kaaadettua, ei varmenteita voisi tarkistaa. Tätä ei koskaan testattu johtuen siitä että projektiryhmällä ei ollut aikataulullisia resursseja testausympäristön rankentamiseen (oma OCSP palvelin). Palvelunesto hyökkäyksen suorittaminen toiminnassa olevaa palvelinta vastaan on nyky lainsäädännön mukaan laitonta, joten tätä vaihtoehtoa ei koskaan harkittu.

## SSLStrip

Tutkimusvaiheessa esiin tullut hyökkäysmetodi, joka käytännössä on välimieshyökkäys. Teoriassa hyökkääjäkone välittää kaikki kohdekoneen kaapatun liikenteen portin 80 (HTTP) kautta salaamattomana. Hyökkääjän ja palvelimen välinen liikenne tapahtuu HTTPS yhteyttä käyttäen. Tämä hyökkäys edellyttää että kohdekoneelle suoritetaan ARP myrkytys, jossa hyökkääjä naamioidaan verkon reitittimeksi.

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, päättimme testata valmista framework:a joilla pystyisi suorittamaan saman hyökkäyksen.

Testaus piti alunperin tehdä käyttäen Ettercat ohjelmaa, tutkinnan kautta selvisi että siitä oli kehittyneempi versio, Bettercap. Tämä ohjelma valikoitui loppujen lopuksi viimeiseen välimieshyökkäys testaukseen. Hyökkäys järjestys on lähes identtinen, tässä voidaan hyödyntää SSLSniff hyökkäyksessä käytettyä metodia, jossa hyökkääjän varmenne välitetään/asennetaan kohdekoneelle.

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 saaneet tätä onnistumaan. Huomasimme testauksen aikana että verkkosivustot jotka käyttävät Certificate Pinning funktiota, nämä sivustot huomaatavat hyökkäysyrityksen väärennetyllä varmenteella ja antavat selaimessa virheilmoituksen.

## Tulosten yhteenveto

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

Projektin aikana ei löytynyt sopivaa hyökkäysvektoria WoT kohtaan, tästä syystä 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.

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

# Suositukset varmenteiden käyttöön ( vai meneekö yhteenvetoon?)

Yhteenvedon voit halutessasi myös jättää pois.

# Yhteenveto

# Lähteet

# Liitteet

## Liite 1: Tuotekortti: Tietoturvan peruskartoitus

## Liite 2. Kartoituskysymykset vastauksineen