Lauri Soivi & Niko Kiuru

VERKKOSIVUJEN TLS-SALAUKSEN TASO HELSINGIN PÖRSSIYHTIÖILLÄ

JYVÄSKYLÄN YLIOPISTO
INFORMAATIOTEKNOLOGIAN TIEDEKUNTA
2022

TIIVISTELMÄ

Soivi, Lauri & Kiuru, Niko
Verkkosivujen TLS-salauksen taso Helsingin pörssiyhtiöillä
Jyväskylä: Jyväskylän yliopisto, 2022, 147 s.
Kyberturvallisuus, Pro gradu -tutkielma
Ohjaajat: Lehto, Martti & Hämäläinen, Timo

Internetin käyttö on lisääntynyt viime vuosien aikana merkittävästi, mikä on johtanut siihen, että verkossa jaetaan enenevissä määrin arkaluonteista tietoa. Verkkosivun ja verkkoselaimen välillä siirrettävän arkaluonteisen tiedon turvalliseen välittämiseen on kehitetty TLS-salaus, mutta tämän käyttöönottaminen ei yksinään takaa riittävää turvallisuutta. Tämä johtuu siitä, että TLS-salaus koostuu useista eri asetuksista, jotka verkkosivustojen ylläpitäjien pitää huomioida, jotta salaus voidaan toteuttaa riittävän vahvasti. Jos salausta ei toteuteta riittävän hyvin, voi pahantahtoisille tahoille vuotaa arkaluonteista tietoa, kuten käyttäjätunnuksia tai salasanoja.

Tämän tutkielman päätavoitteena oli perehtyä Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen tasoon ja tarkastella, onko kyseiset verkkosivut salattu riittävän vahvalla TLS-salauksella. Tutkimuksen alatavoitteena oli kehittää työkalu, jolla voidaan kerätä tietoa verkkosivujen TLS-salauksesta. Tutkimuksen teoriaosuudessa esitellään oleellisia käsitteitä liittyen verkkosivustojen toimintaan sekä keskitytään TLS-salaukseen ja siihen liittyviin osioihin. Teoriaosuuden lopussa käsitellään vahvan TLS-salauksen tärkeyttä sekä luodaan TLS-salauksen suositeltu vähimmäistaso, jota vasten kerättyjen verkkosivujen TLS-salausta verrattiin. Kehitetyn työkalun avulla kerättiin 4431 kappaletta Helsingin pörssiyhtiöiden julkisia verkkosivuja, joiden TLS-salauksen tasoa arvioitiin. Työkalu julkaistiin julkiseen käyttöön, jotta kuka tahansa pystyy hyödyntämään tai kehittämään työkalua omiin tarpeisiinsa.

Tutkimuksessa vain 3,5 prosenttia kerätyistä verkkosivuista täytti kaikki asetetut TLS-salauksen vähimmäistason suositukset. Tulos viittaa siihen, että tutkittujen verkkosivujen TLS-salauksien vahvuudessa olisi parannettavaa. Tutkimuksessa avattiin tarkemmin kaikki kymmenen asetettua suosituskategoriaa, jotta nähtiin yleisimmät syyt sille, miksi verkkosivut eivät läpäisseet asetettuja suosituksia. Nämä yleisimmät puutteet löytyivät salaussarjoista, OCSP staplingista ja HSTS:stä. Tutkimuksessa esitettiin mahdollisia syitä, miksi verkkosivujen TLS-salauksen eri osa-alueet eivät täyttäneet asetettuja suosituksia, mutta todellisten syiden selvittäminen jätettiin jatkotutkimuksiin, koska niiden selvittäminen ei kuulunut tämän tutkimuksen tavoitteisiin.

Asiasanat: TLS, HTTPS, verkkosivu, kyberturvallisuus, salaus, Nasdaq Helsinki, X.509

ABSTRACT

Soivi, Lauri & Kiuru, Niko
TLS encryption strength of websites by companies listed on Nasdaq Helsinki
Jyväskylä: University of Jyväskylä, 2022, 147 p.
Cyber security, Master's Thesis
Supervisors: Lehto, Martti & Hämäläinen, Timo

The use of the Internet has increased significantly in recent years, which has led to the sharing of more sensitive data online. TLS encryption has been developed for the secure transmission of sensitive information transferred between a website and a web browser, but the implementation of this alone does not guarantee sufficient security. This is because TLS encryption consists of several different settings that website administrators need to consider for the encryption to be strong enough. If the encryption is implemented poorly, sensitive information, including usernames and passwords, can be leaked to malicious parties.

The main objective of this research was to investigate the strength of TLS encryption of websites listed on Nasdaq Helsinki and to examine whether these websites have been encrypted with a sufficiently strong TLS encryption. The sub-objective of this research was to develop a tool that can be used to collect information regarding the TLS encryption of websites. The theoretical section of the research introduces the key concepts related to websites and focuses on TLS encryption. The end of the theory section includes an explanation of the importance of using secure TLS encryption, as well as a compilation of the recommended minimum level of TLS encryption, which was compared against the TLS encryption of the collected websites. The developed tool was used to collect 4431 public websites of Nasdaq Helsinki, and their TLS encryption strength was assessed. The tool was also released for public use, so anyone can use or develop it for their own purposes.

The research findings showed that only 3.5 percent of the collected websites met all the TLS encryption recommendations. This finding suggests that the TLS encryption strength of the websites would require improvement. The research also reviewed all the ten TLS encryption recommendation settings to see the most common reasons why the websites did not meet the recommendations. Most commonly, flaws were found in the lack of cryptographic sets, OCSP stapling, and HSTS. This research also considered reasons why the different settings of TLS encryption on the websites did not meet general recommendations, however, finding the actual reasons behind this phenomenon were left for future research, as it was not among the objectives of this research.

Keywords: TLS, HTTPS, website, cyber security, encryption, Nasdaq Helsinki, X.509

KUVIOT

TAULUKOT

SISÄLLYS

1 JOHDANTO

Internetin käyttö tunneissa mitattuna on kasvanut maailmanlaajuisesti vuodesta toiseen, ja tällä hetkellä Internetiä käytetään keskimäärin seitsemän tuntia päivässä käyttäjää kohden. Tämä on johtanut siihen, että verkossa jaetaan kasvavissa määrin arkaluonteista tietoa, mikä nostattaa aivan uudenlaisia uhkakuvia turvallisuudelle (Global Web Index & Datum Future, 2019, s. 7.).

Edward Snowden paljasti vuoden 2013 kesäkuussa Yhdysvaltain kansallisen turvallisuusviraston (NSA) laajan vakoiluohjelman (Scheuerman, 2014), mikä vahvisti sitä, että valtiolliset toimijat ovat yhä kiinnostuneempia seuraamaan Internetin käyttäjiä useilla eri tavoilla. Kasvava Internetin käyttö tekee verkkoliikenteen seuraamisesta houkuttelevan kohteen valtiollisten toimijoiden lisäksi muun muassa verkkorikollisille, jotka yrittävät hyödyntää Internetistä löytyviä heikkouksia lähes kaikilla mahdollisilla keinoilla.

Vuonna 2019 alkanut COVID-19-pandemia johti useisiin erilaisiin sosiaalisiin rajoituksiin, joiden takia ihmisten vapaa-aika ja työskentely siirtyivät enenemissä määrin fyysisestä maailmasta Internetiin, mikä on myös kasvattanut arkaluonteisen materiaalin käsittelyä Internetissä. Pandemian myötä verkkorikollisuuden määrät ovat myös kasvaneet, mikä on koskettanut yritysten lisäksi tavallisia Internetin käyttäjiä. Nämä verkkorikokset voivat olla esimerkiksi sellaisia, joissa yritetään saada käyttäjiä rikollisen omille verkkosivustoille tai salakuunnella käyttäjän ja verkkosivun välistä liikennettä. Tällaisten toimien avulla rikollinen yrittää saadaan käyttäjältä huijattua muun muassa rahaa tai arkaluontoista tietoa, joka voi olla esimerkiksi käyttäjätunnuksia ja salasanoja. (Bou Sleiman & Gerdemann, 2021, ss. 37–41.)

Historian varrelta löytyy useita erilaisia tapauksia, joissa yritykset ovat vuotaneet käyttäjiensä tietoja; esimerkiksi vuonna 2019 Alibaba-yhtiö menetti yli miljardin asiakkaansa tietoja, ja vuonna 2021 LinkedIn-verkkosivu menetti 700 miljoonan käyttäjänsä tietoja (Hill & Swinhoe, 2021). Tällainen tiedon vuotaminen voi vaikuttaa myös yritykseen ja yrityksen maineeseen kohtalokkain seurauksin. Yhtenä esimerkkinä arkaluonteisen tiedon menettämisestä ja yrityksen maineen kärsimisestä on vuonna 2020 tietoon tullut Psykoterapiakeskus Vastaamon tietomurtotapaus, jossa yrityksen asiakkaiden arkaluontoista tietoa

päätyi verkkorikollisille, minkä myötä yritys hakeutui konkurssiin (Hämäläinen, 2021).

Noussutta turvallisuusuhkaa voitaisiin hillitä panostamalla Internetissä tarjottavien palveluiden tietoturvaan esimerkiksi salaamalla verkkoliikenne TLS-salauksella. Krombholz ym. (2019, s. 246) havaitsivatkin tutkimuksessaan, että verkkosivujen ylläpitäjät ymmärtävät salatun verkkosivun merkityksen. Salaamattoman verkkoliikenteen muuttaminen salatuksi onkin näkynyt viime vuosien aikana muun muassa siinä, että vuoden 2014 jälkeen TLS-salattujen verkkosivujen määrät ovat nousseet alle 30 prosentista tämän hetken noin 80 prosenttiin (Let's Encrypt, 2022). Pelkkä salaus ei kuitenkaan itsessään riitä, vaan se pitää myös toteuttaa riittävän turvallisesti, jotta siitä on hyötyä.

TLS-salaus koostuu useista asetuksista, joiden eri arvot vaikuttavat salauksen turvallisuuteen. TLS-salauksessa voidaan yhä käyttää muun muassa vanhoja salausprotokollia, heikkoja salausalgoritmeja tai haavoittuviksi todettuja ominaisuuksia, jotka tekevät verkkosivusta heikosti salatun. SSL Labs (2021) -verkkosivun julkaiseman tilaston mukaan 52,8 % verkkosivuista käyttää riittämätöntä salausta, mikä viittaa siihen, että suurinta osaa verkkosivuista ei ole turvattu riittävän vahvasti. Salaus kadottaa merkitystään, jos sitä ei ole toteutettu luotettavasti.

Tässä tutkielmassa keskityttiin tutkimaan salattujen verkkosivujen TLS-salauksen vahvuutta. Tutkielman päätavoitteena oli selvittää, mikä on Helsingin pörssiyhtiöiden (Nasdaq Helsinki) verkkosivujen salauksen taso. Helsingin pörssi tuli tutkimukseen valituksi, koska se oli riittävän hyvin rajattu, ja pörssiyhtiöiltä voisi olettaa löytyvän tarpeeksi resursseja turvallisten verkkosivujen ylläpitämiseen.

Tutkimuksen alatavoitteena luotiin työkalu, jonka avulla pystytään tutkimaan suurta joukkoa verkkosivuja ja selvittämään niiden TLS-salauksen vahvuus. Työkalu luotiin, koska tiedeyhteisön käytössä ei tiettävästi ole tämänkaltaista työkalua. Työkalulla myös kerättiin tämän tutkimuksen tutkimusaineisto, jolla todistettiin työkalun toimivuus.

1.1 Aiemmat tutkimukset

Salattua HTTPS-yhteyttä on tutkittu siitä lähtien, kun HTTP-yhteyksiä on salattu SSL/TLS-salausprotokollalla. Teknologioiden kehittymisen ja tiedon lisääntymisen myötä monen tutkimuksen tieto on jo ehtinyt vanhentumaan. Tätä tutkimusta varten perehdyttiin vanhempaankin materiaaliin, mutta oleellisimmat tutkimukset löytyivät viimeisen neljän vuoden ajalta, mihin yhtenä syynä on se, että vuonna 2018 julkaistiin uusin versio TLS-protokollasta.

Kyberturvallisuuden näkökulmasta TLS-salaukseen liittyviä tutkimuksia löytyy jonkin verran, mutta niitä on toteutettu melko erilaisilla tavoilla, sillä TLS-salauksen turvallisuuden mittaamiseen ei ole yhtä selkeää tapaa. Tutkimuksissa tarkasteltavat kohteet keskittyvät TLS-salauksen eri osa-alueisiin. Aikaisemmissa tutkimuksissa on tarkasteltu muun muassa verkkosivuilla käytet-

täviä salaussarjoja (Weerasinghe & Disanayake, 2018), TLS-protokollan versioita (Silva & Fonte, 2019) ja varmenteita (Alashwali ym., 2019). Näissä tutkimuksissa on kuitenkin jätetty huomioimatta muita TLS-salauksen osa-alueita, kuten TLS-pakkaus, 0-RTT ja OCSP stapling, jotka vaikuttavat TLS-salauksen luotettavuuteen.

Aiemmissa tutkimuksissa ei ole määritelty kattavasti, mikä TLS-salauksen vähimmäistason pitäisi olla, vaan niissä on poimittu muutamia TLS-salauksen osa-alueita ja keskitytty vain niihin (Alashwali ym., 2019; Weerasinghe & Disanayake, 2018). Todellisuudessa TLS-salaus on yksi kokonaisuus, jonka kaikkien osa-alueiden tulisi olla määriteltynä riittävälle tasolle, jotta TLS-salaus olisi luotettava. Useampi tunnettu toimija onkin julkaissut TLS-salauksen tasosta suosituksensa, joita verkkosivujen ylläpitäjien olisi hyvä noudattaa (Dutch National Cyber Security Center, 2020; McKay & Cooper, 2019; Mozilla, 2020b; SSL Labs, 2020). Aikaisemmissa tutkimuksissa näitä suosituksia on kuitenkin laiminlyöty, eikä suosituksia ole hyödynnetty esimerkiksi viitekehyksenä, jonka avulla voitaisiin arvioida verkkosivujen TLS-salauksen tasoa ja selvittää, mihin osa-alueisiin verkkosivujen ylläpitäjien pitäisi keskittyä. Koska vähimmäistasoa ei ole aikaisemmissa tutkimuksissa määritelty ja eri toimijoiden suositukset saattavat erota toisistaan, olisi suotavaa luoda näistä toimijoiden suosituksista konsensus, jota voitaisiin käyttää verkkosivujen TLS-salauksen vähimmäistasona.

Useammassa eri tutkimuksessa on tutkittu TLS-salauksen turvallisuutta erilaisilla verkkosivuilla, kuten verkkokauppojen (Strzelecki & Rizun, 2020), valtion (Ali & Murah, 2018), rahoitusalan (Weerasinghe & Disanayake, 2018) sekä julkisen sektorin (Silva & Fonte, 2019) verkkosivuilla, mutta yhdessäkään tutkimuksessa kohteena ei ole ollut suomalaisten yhtiöiden verkkosivut. Kaikille yllä mainituille tutkimuksille on yhtenäistä se, että valtaosassa (>50 %) verkkosivujen TLS-salauksesta löytyi jotakin parannettavaa. Kiinnostavaa onkin selvittää suomalaisten yhtiöiden verkkosivujen TLS-salauksen tasoa ja tarkastella, onko myös niiden salauksessa parannettavaa.

Joissakin tutkimuksissa (Ali & Murah, 2018; Silva & Fonte, 2019; Strzelecki & Rizun, 2020) TLS-salauksen turvallisuuden mittaamiseen on käytetty Qualys Inc -yhtiön tarjoamaa SSL Labs -työkalua, joka kerää erilaisia TLS-salauksen asetuksia ja antaa verkkosivulle TLS-salauksen turvallisuudesta arvosanan. Näissäkin tutkimuksissa on nostettu esille vain muutamia eri osa-alueita TLS-salauksesta tai tarkasteltu vain SSL Labsin antamaa arvosanaa, joka ei itsessään paljasta, mitä heikkouksia verkkosivujen TLS-salauksessa on ollut ja mihin osa-alueisiin TLS-salauksessa pitäisi kiinnittää huomiota, jotta TLS-salaus olisi toteutettu turvallisemmin.

SSL Labs -työkalu voisi olla toimiva tapa mitata verkkosivujen TLS-salausta, mutta sen teknisten rajoitteiden vuoksi se ei sovellu tutkimuksiin, joissa tutkittavien verkkosivujen määrä kasvaa tuhansiin. Aiemmissa tutkimuksissa tutkittavat verkkosivut oli valmiiksi kerätty tai ne olivat helposti saatavilla (Alashwali ym., 2019; Strzelecki & Rizun, 2020), minkä takia tutkittavia verkkosivuja ei kerätty ohjelmallisesti.

Aiemmissa tutkimuksissa ilmeni puutteena, ettei tiedeyhteisön käytössä tiettävästi ole työkalua, jolla pystyisi keräämään isoa määrää tutkittavia verkkosivuja ja tarkastelemaan niiden TLS-salauksen tasoa. Tämän vuoksi työkalulle nähtiin tarvetta ja se toteutettiin tässä tutkimuksessa. Työkalu on sittemmin julkaistu julkiseen käyttöön, jotta kuka tahansa voi hyödyntää sitä omiin tarpeisiinsa. Työkalulla kerättiin myös tässä tutkimuksessa tarvittava aineisto, jolla todistettiin työkalun toiminta käytännössä.

Vaikka HTTP-yhteyden salaaminen ei ole uusi keksintö ja TLS-salausta on tutkittu useassa tutkimuksessa, on kyseistä aihealuetta tutkittu liian vähän ja aiheeseen liittyvistä tutkimuksista löytyy useita huomioimattomia kohtia. Sellaista tutkimusta, joka ottaisi kantaa tässä luvussa mainittuihin puutteisiin, ei tiettävästi ole tehty, minkä takia tutkimukselle nähtiin tarvetta. Seuraavassa luvussa asetetaan tutkimuksen tavoitteet, joissa huomioidaan aiemmissa tutkimuksissa havaitut puutteet ja ongelmakohdat, joihin puutuimme tässä tutkimuksessa.

1.2 Tutkimuksen tavoitteet

Tutkimusongelmamme kiteytyi siihen, että Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen tasosta ei ole aikaisempaa tietoa, minkä takia sitä pyrittiin tässä tutkimuksessa selvittämään. Tutkimuksen päätavoitteena oli selvittää Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso, mistä päätutkimuskysymykseksi muodostui:

- Mikä on Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso?

Tutkimuksen alatavoitteena oli toteuttaa työkalu, jolla on mahdollista kerätä aineistoa verkkosivujen TLS-salausta käsitteleviä tutkimuksia varten. Työkalu toteutettiin, koska tiedeyhteisöllä ei tiettävästi ole ollut käytössä tämänkaltaista työkalua, jolla voitaisiin kerätä laaja määrä verkkosivuja ja tietoa niiden TLS-salauksesta. Tutkimukselle asetettiin kaksi eri tutkimustavoitetta, koska päätutkimustavoitetta ei ole mahdollista toteuttaa ilman alatutkimustavoitteen toteuttamista. Molempien tavoitteiden toteuttamisella pyrittiin todistamaan työkalun toimivuus. Helsingin pörssiyhtiöiden verkkosivujen lukumäärä mahdollistaa työkalun toimivuuden testaamisen käytännössä. Seuraavaksi esitellään tutkimuksen alatavoitteeseen liittyvät tutkimuskysymykset.

1.2.1 Tutkimuksen alatavoite

Tutkimuksen alatavoitteena kehitettiin työkalu aineiston keräämistä varten. Työkalulla oli tarkoitus kerätä pääverkkotunnuksiin liittyvät aliverkkotunnukset sekä niiden TLS-salausasetukset. Jotta voitiin arvioida, täyttikö työkalu tutkimuksen alatavoitteen, sille asetettiin kolme tutkimuskysymystä. Tutkimusta

varten toteutettavan työkalun tulisi kerätä aineistoa, jota sen on suunniteltukin keräävän. Tästä syystä alatavoitteen ensimmäisenä tutkimuskysymyksenä on:

- Onko työkalun keräämä data validia?

Tässä tapauksessa validilla tarkoitetaan sitä, että työkalu löytää Helsingin pörssiyhtiöille kuuluvia aliverkkotunnuksia ja että TLS-salauksesta kerätty aineisto on virheetöntä. Tähän kysymykseen vastattiin työkalun toteutusvaiheessa, jossa työkalun keräämää dataa tarkistettiin manuaalisesti ajon eri vaiheissa. Työkalun ajosta myös kerättiin kaikki siinä ilmenneet virheet. Virheistä tarkistettiin, ilmenivätkö ne työkalussa olevan virheen vuoksi vai johtuivatko ne ajettavasta kohteesta. Työkalun tarkoituksena oli kerätä aineisto tutkimusta varten, minkä takia alatavoitteen toiseksi tutkimuskysymykseksi asetettiin:

- Mahdollistaako työkalu aineiston keräämisen tämänkaltaista tutkimusta varten?

Työkalua hyödynnettiin päätavoitteen aineiston keräämiseen, jonka avulla myös validoitiin työkalun toimiminen isommassa mittakaavassa. Jos työkalulla pystyttiin keräämään ainestoa päätavoitetta ja sen tutkimuskysymyksiä varten, on aineiston kerääminen onnistunut. Keräämisessä myös huomioitiin se, että aineisto saatiin kerättyä kohtuullisen ajan puitteissa. Tässä tapauksessa kohtuulliseksi ajaksi määriteltiin kuukausi, koska muutoin itse tutkimuksen toteuttaminen olisi venynyt liiaksi, varsinkin jos datan keräämistä olisi pitänyt tehdä useampia kertoja esimerkiksi työkalussa löytyneen virheen vuoksi.

Tutkimuksen valmistumisen jälkeen työkalu julkaistiin, jotta kuka tahansa voi vahvistaa työkalun toiminnan ja tutkimuksen tulokset sekä hyödyntää työkalua omassa tutkimustyössään. Koska tutkimuksen alatavoitteena oli toteuttaa työkalu, jota voitaisiin hyödyntää tulevaisuudessa muissakin tutkimuksissa, alatavoitteen kolmanneksi ja viimeiseksi tutkimuskysymykseksi asetettiin:

- Miten työkalun toimintaa voitaisiin kehittää?

Seuraavassa luvussa käsitellään tutkimuksen päätavoite ja asetetaan siihen liittyvät tutkimuskysymykset.

1.2.2 Tutkimuksen päätavoite

Tutkimuksen päätavoitteena oli selvittää, mikä on Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso. Päätavoitteeseen haettiin vastaus neljän tutkimuskysymyksen avulla.

Nykypäivänä TLS-salaus on noussut laajalti julkisuuteen erinäisten tietoturvaloukkausten takia, ja verkkosivujen käyttäjiäkin kehotetaan tarkastamaan selaimen osoiteriviltä niin sanotun ”vihreä lukon kuva”, eli käyttääkö verkkosi-

vu HTTPS-yhteyttä. Tämä saattaa aiheuttaa monille ihmetystä, minkä takia nähtiin oleellisena asettaa päätavoitteen ensimmäiseksi tutkimuskysymykseksi:

- Miksi turvallisen TLS-salauksen käyttäminen on tärkeää?

Ensimmäinen tutkimuskysymys asetettiin, jotta ymmärretään tarkemmin syitä, miksi verkkosivuilla pitäisi käyttää TLS-salausta ja miksi TLS-salaus pitäisi toteuttaa turvallisesti. Tähän kysymykseen vastataan perehtymällä kirjallisuuskatsauksessa siihen, miten salaamaton HTTP-protokolla toimii, miten TLS-salaus toimii ja miten HTTP-verkkoliikenne salataan sekä mitä haittaa on heikosti toteutetusta TLS-salauksesta tai siitä, jos HTTP-verkkoliikennettä ei ole ollenkaan salattu.

TLS-salaus koostuu useasta eri asetuksesta, jotka voidaan asettaa eri arvoihin. Ajan myötä osa TLS-salauksen asetuksista on todettu turvattomiksi, minkä takia TLS-salaus voidaan myös toteuttaa heikosti. Tästä syystä toiseksi tutkimuskysymykseksi asetettiin:

- Mikä on TLS-salauksen suositeltu vähimmäistaso?

Toiseen tutkimuskysymykseen vastataan kirjallisuuskatsauksessa, jossa luodaan konsensus TLS-salauksen suosituksista neljän eri toimijan julkaisemia TLS-salauksen suosituksia hyväksikäyttäen. Tässä tutkimuksessa suositellulla vähimmäistasolla tarkoitettiin turvallisten TLS-salauksen asetuksien senhetkisiä minimivaatimuksia.

Toinen tutkimuskysymys asetettiin, jotta pystyttiin vastaamaan kolmanteen tutkimuskysymykseen. Päätavoitteen kolmantena tutkimuskysymyksenä on:

- Kuinka moni Helsingin pörssiyhtiöiden verkkosivu läpäisi asetetut TLS-salauksen suositukset?

Kolmas tutkimuskysymys asetettiin, jotta saatiin kartoitettua tämän hetken TLS-salauksen tilanne Helsingin pörssiyhtiöiden verkkosivuissa. Tähän kysymykseen vastattiin vertaamalla TLS-salauksen suosituksia verkkosivujen TLS-salausta vasten. Päätavoitteen neljänneksi tutkimuskysymykseksi asetettiin:

- Mikä on yleisin syy sille, että verkkosivut eivät läpäisseet asetettuja TLS-salauksen suosituksia?

Tutkimuksen tuloksissa (LUKU 5) esitetään jokainen TLS-salauksen suositus omana kategorianaan. Kategoria, johon sijoittui eniten suosituksen alittaneita verkkosivuja, oli vastaus neljänteen tutkimuskysymykseen. Neljännen tutkimuskysymyksen tarkoituksena oli selvittää, mihin TLS-salauksessa yhtiöiden pitäisi vastaisuudessa kiinnittää enemmän huomioita, jotta useampi verkkosivu täyttäisi asetetut TLS-salauksen suositukset.

Seuraavassa luvussa käsitellään tutkimuksen rakenne, joka noudattaa valitun tutkimusmenetelmän tutkimusprosessia.

1.3 Tutkimuksen rakenne

Tutkimusraportti koostuu seitsemästä luvusta, joiden sisältö käsitellään lyhyesti tässä luvussa. Tutkimuksessa käytetään sovelletusti konstruktiivista tutkimusotetta, johon kuuluu seitsemän eri prosessin vaihetta, joita noudatimme tutkimuksessamme.

Konstruktiivisen tutkimusotteen ensimmäisenä vaiheena on etsiä relevantti ongelma, joka halutaan ratkaista. Tämä relevantti ongelma kuvailtiin yllä, jossa käsiteltiin tutkimuksen motiivit ja määriteltiin tarkemmin tutkimuksen tavoitteet. Tutkimusmenetelmän toisena vaiheena on löytää kohdeorganisaatio, jolle ongelma ratkaistaisiin, mutta tässä tutkimuksessa sovellettiin valittua tutkimusmenetelmää tavalla, jossa kohdeorganisaatiota ei valita. Tutkimusmenetelmän soveltamisesta kerrotaan Konstruktiivinen tutkimusote prosessina -luvussa (LUKU 4.3), jossa myös perustellaan, miksi tutkimuksessa ei valittu kohdeorganisaatiota. Tutkimuksessa hyödynnettiin tahoja Jyväskylän Yliopistolta sekä Suomen Kyberturvallisuuskeskukselta, mutta tutkielmaa ei toteutettu kummankaan toimeksiantona. Tutkimuksen kohdeyleisönä on tiedeyhteisö, ja tutkimuksessa toteutettava konstruktio on julkaistu muita tutkijoita ja tahoja varten, mikäli he haluavat konstruktiota hyödyntää omiin käyttötarkoituksiinsa.

Kolmannessa vaiheessa hankitaan syvällinen aiheen tuntemus. Tämä tehdään toisessa ja kolmannessa luvussa, joissa käsitellään tutkimuksen aihepiirin kirjallisuutta ja teoriaa. Toisessa luvussa perehdytään tietoliikenneprotokollista koostuvaan TCP/IP-pinoon, jonka avulla useat sovellukset, kuten verkkoselaimet, keskustelevat keskenään Internetissä. Samassa luvussa käsitellään myös, mitä ovat URL, verkkotunnus ja DNS. Kolmannessa luvussa käsitellään HTTPS-protokollaa ja syvennytään tarkemmin TLS-salaukseen, varmenteisiin ja TLS-salauksen heikkouksiin sekä muodostetaan tutkimusosuutta varten konsensus TLS-salauksen suosituksista.

Neljännessä vaiheessa innovoidaan ja kehitetään konstruktio. Tämä tehdään tutkielman neljännessä luvussa, jossa konstruktio, eli tutkimuksessa toteutettava työkalu, kehitetään sekä validoidaan. Neljännessä luvussa myös käsitellään meidän kontribuutiomme tutkimukseen liittyen, syvennytään tarkemmin konstruktiiviseen tutkimusotteeseen sekä kerrotaan tarkemmin, miten kyseistä tutkimusmenetelmää hyödynnettiin tässä tutkimuksessa. Luvun lopussa dokumentoidaan, kuinka tutkimuksen aineisto on kerätty ja miten aineistoa käsiteltiin tässä tutkimuksessa.

Tutkimusmenetelmän viidennessä vaiheessa toteutetaan ja testataan konstruktio. Neljännessä luvussa toteutettu konstruktio ajetaan tutkimuskohteita vasten, mikä toimii konstruktion kokonaisvaltaisena testauksena. Tästä testauksesta saatu aineisto on tutkimuksessa käytetty aineisto, joka esitellään tutkielman tulososiossa eli viidessä luvussa.

Kuudennessa vaiheessa pohditaan tutkimuksen soveltamisalaa, ja seitsemännessä vaiheessa analysoidaan teoreettinen kontribuutio. Nämä molemmat vaiheet toteutetaan Pohdinnassa (LUKU 6), jossa vastataan kaikkiin tutkimuskysymyksiin, arvioidaan kriittisesti tutkimuksen reliabiliteettia ja validiteettia sekä analysoidaan tutkimuksen käytännön ja tieteellinen kontribuutio.

Tutkimuksen viimeisenä lukuna on Johtopäätökset (LUKU 7), jossa on koko tutkimuksen yhteenveto ja esitellään uusia mahdollisia jatkotutkimusaiheita. Seuraavassa luvussa kerrotaan tutkimukseen asetetut rajaukset.

1.4 Tutkimuksen rajaus

Tutkimuksessa jouduttiin tekemään rajauksia ja oletuksia, joista keskeisimmät rajaukset käydään läpi tässä luvussa.

Helsingin pörssiyhtiöiden pääverkkotunnukset noudettiin Yahoo Financen rajapinnasta saatavilla olevasta kotisivusta. Todellisuudessa ei ole rajoitteita, etteikö yhtiöllä voisi olla useita eri kotisivuja, mutta tässä tutkimuksessa sillä viitataan nimenomaan Yahoo Financen rajapinnasta löytyneeseen verkkosivuun.

Yhtiöillä on myös hyvin todennäköisesti useita eri pääverkkotunnuksia eri päätteillä (ylätason verkkotunnuksilla), mutta tässä tutkimuksessa verkkotunnukset rajattiin vain pääverkkotunnukseen, joka löytyi yhtiön kotisivusta. Tämä johtui siitä, että Internetissä ei julkisesti pystytä tunnistamaan verkkotunnuksen omistajaa. Vaikka yhtiö omistaisi yhden päätteellisen pääverkkotunnuksen, toista päätettä se ei välttämättä enää omistaisikaan. Tutkimuksessa tehtiin kotisivun kohdalla oletus, että yhtiö omistaa sen kotisivun pääverkkotunnuksen, joka löytyi Yahoon Financen rajapinnasta.

Työkalulla suoritettiin TLS-salauksen testit ainoastaan HTTPS:n oletusporttiin 443. Todellisuudessa verkkosivut voidaan asettaa myös toiseen porttiin, eikä vain oletusporttiin. Tämä rajaus tehtiin siksi, että kaikkien muiden mahdollisten porttien tarkastaminen WWW-palvelun varalta olisi kasvattanut ajoaikaa kohtuuttoman paljon. Muiden kuin 443-porttien skannaamista olisi voitu myös pitää laittomana.

Tutkimus ei tuo vastausta siihen, miksi verkkosivujen TLS-salaus on toteutettu tietyllä vahvuudella. Työkalulla pystytään selvittämään, mikä salauksen taso on, mutta se ei kerro, mitkä syyt johtavat heikosti tai vahvasti toteutettuun TLS-salaukseen. Tämän vuoksi syyt, jotka johtavat selvitettävään salauksen tasoon, on rajattu pois tästä tutkimuksesta.

Tutkimuksessa ei myöskään oteta kantaa verkkosivujen sisältöön, yritysten kokoihin, toimialoihin tai vastaaviin muihin ominaisuuksiin, koska näihin perehtyminen nähtiin omaksi tutkimuskohteeksi. Tutkimuksessa käsitellään kaikkia kerättyjä verkkosivuja yhdenvertaisina.

Seuraavaksi siirrytään teoriaosuuteen ja perehdytään TCP/IP:n toimintaan.

2 Internet

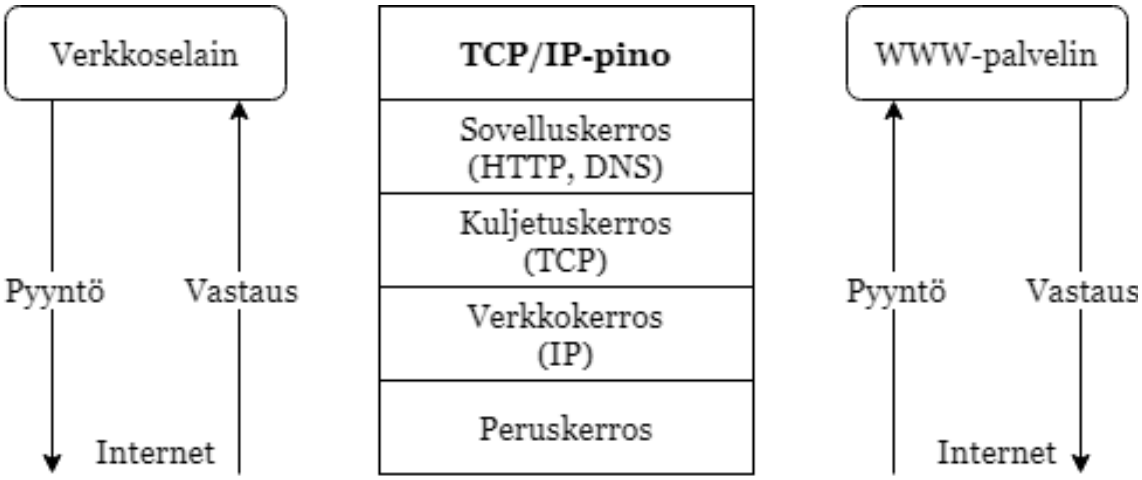
Väitetysti yksi suurimpia ihmisen tekemiä teknologisia taidonnäytteitä on Internet, joka koostuu siihen kytketyistä verkkolaitteista, kuten käyttäjien tietokoneista ja WWW-palvelimista (Kurose & Ross, 2013, s. 27). Internetissä nämä verkkolaitteet ovat laajalti kytkeytyneet toisiinsa käyttäen TCP/IP-pinoa (engl. TCP/IP model), jonka avulla verkkolaitteet kommunikoivat keskenään (Goralski, 2009, s. xxxii). TCP/IP on yhdistelmä useista eri tietoliikenneprotokollista, ja sitä voidaan kutsua myös nimellä TCP/IP-pino. TCP/IP-pino koostuu neljästä kerroksesta (engl. layer) (Loshin, 2003, s. 78), joiden yhteistoiminta mahdollistaa esimerkiksi verkkosivun (WWW-sivu, web-sivu tai nettisivu) lataamisen selaimella Internetissä olevasta WWW-palvelimesta (Kurose & Ross, 2013, s. 35).

TCP/IP-pinon jokaisessa kerroksessa toimii useita erilaisia protokollia, joista kaikilla on oma tehtävänsä Internetin toiminnassa (Goralski, 2009, s. 3). TCP/IP-pinon toiminnan ymmärtäminen ei ole rakettitiedettä, mutta sen ymmärtäminen vaatii tietämystä käsitteistä sekä niiden merkityksestä ja eri kerroksissa toimivien protokollien tehtävistä (Loshin, 2003, s. 5). Tämän luvun alaluvussa kerrotaan tutkimuksen kannalta oleellisimmista TCP/IP-pinon kerroksista ja protokollista sekä niihin liittyvistä teknologioista, jotta ymmärretään verkkoliikenteen toimintaa.

Alaluvuissa perehdytään sovelluskerrokseen kuuluviin HTTP- ja DNS-protokolliin, kuljetuskerrokseen liittyvään TCP-protokollaa sekä verkkokerrokseen kuuluvaan IP-protokollaan. Alaluvuissa kerrotaan myös, mitä ovat URL ja verkkotunnus, joita hyödynnetään muun muassa verkkosivustojen hakemisessa Internetistä. Luvun lopussa olevassa yhteenvedossa nivotaan kaikkien näiden teknologioiden toiminta yhteen ja kerrotaan, miten käyttäjän verkkoselain pyytää verkkosivustoa WWW-palvelimelta ja käyttäjän verkkoselain saa pyydetyn verkkosivuston WWW-palvelimelta. Seuraavassa alaluvussa käsitellään TCP/IP-pinoa ja siihen kuuluvia kerroksia.

2.1 TCP/IP-pino

TCP/IP-pino koostuu neljästä eri kerroksesta, jotka ovat sovelluskerros, kuljetuskerros, verkkokerros ja peruskerros (tunnetaan myös siirtoyhteyskerroksena tai fyysisenä kerroksena) (KUVIO 1). Peruskerrosta ja sen toimintaa ei käsitellä tutkimuksessa tarkemmin, koska tutkimuksessa tarkastellaan verkkoliikenteen salaamista, johon peruskerros ei liity.



KUVIO 1 TCP/IP-pino (Loshin, 2003, p. 79; Oppliger, 2016, p. 3)

Kaikilla kerroksilla on omat tehtävänsä. Lisäksi kerrokset toimivat yhteistyössä ylä- ja alapuolella olevien kerrosten kanssa ja välittävät tietoa keskenään. (Loshin, 2003, ss. 78–80.) Kerrokset ja protokollat toimivat yhteistyössä esimerkiksi silloin, kun käyttäjän selain pyytää verkkosivustoa WWW-palvelimelta ja WWW-palvelin vastaa verkkosivulla tähän pyyntöön. Kerrosten välinen yhteistyö on kuvattu yllä olevaan kuvioon (KUVIO 1).

Kuviossa (KUVIO 1) selain lähettää verkkosivustosta pyynnön, joka lähtee sovelluskerroksesta (engl. application layer) ja menee yksitellen kerroksia kuviossa alaspäin, kunnes se saapuu alimmalle kerrokselle eli peruskerrokselle. Jokaisessa kerroksen vaiheessa tehdään erilaisia toimia, jotta pyyntö osataan ohjata Internetissä oikealle WWW-palvelimelle ja jotta vastaanottava WWW-palvelin ymmärtää vastaanotetun pyynnön oikein. Lähettäjän tekemän käsittelyn jälkeen pyyntö ohjataan Internetissä oikealle WWW-palvelimelle, joka käsittelee vastaanotetun pyynnön kuviossa kerroksissa alhaalta ylöspäin ja jokainen kerros käsittelee pyynnön tehtävänsä mukaisesti. Käsiteltyään vastaanotetun pyynnön WWW-palvelin lähettää pyydetyn verkkosivuston vastauksena takaisin selaimelle vastaavalla tavalla kuin se oli saapunutkin, alkaen WWW-palvelimella olevasta sovelluskerroksesta. (Loshin, 2003, ss. 81–83.)

Selaimen ja WWW-palvelimen välistä tiedonsiirtoa voi kuvata postikortin lähettämisenä postilla (Loshin, 2003, s. 6). Annetaan tämän tiedonsiirron toimintaan käytännön esimerkki. Henkilö A, kutsutaan häntä Liisaksi, haluaa pyy-

tää juhliinsa henkilön B, kutsutaan häntä Petriksi. Liisa lähettää Petrille postikortin, jossa kutsutaan Petri juhliin. Kutsussa on mukana pyyntö, jossa pyydetään vastaamaan, pääseekö Petri osallistumaan juhliin. Liisa lähettää postikortin käyttäen postia. Posti toimittaa postikortin Petrille, hyödyntäen postikortissa olevaa postiosoitetta. Petri lukee Liisan lähettämän pyynnön ja vastaa Liisalle lähettämällä vastauksen postikortissa käyttäen Postia. Liisa vastaanottaa Petrin lähettämän postikortissa olevan vastauksen (Loshin, 2003, s. 6.).

Samalla tapaa myös Internetissä TCP/IP-pinoon kuuluvilla teknologioilla toimitetaan postikortteja eli paketteja tietokoneilta ja sovelluksilta toiselle. Edellä kerrottua postikorttiesimerkkiä käytetään myöhemmissä luvuissa, jotta on helpompi ymmärtää TCP/IP-pinon toimintaa käytännössä. Edellä esitetyssä kuviossa on myös mainittuna muutamia TCP/IP-pinon kerroksissa toimivia protokollia, joita käsitellään myöhemmissä luvuissa (KUVIO 1). Seuraavassa alaluvussa tutustutaan sovelluskerroksella toimivaan HTTP-protokollaan ja sen toimintaan.

2.2 HTTP

Internetiä käyttävät ohjelmistot pyytävät ja lähettävät erilaisia sähköisiä resursseja (engl. resource), joita esimerkiksi selaimet pyytävät WWW-palvelimilta. Nämä resurssit voivat olla muun muassa verkkosivustoja, dokumentteja, kuvia, videoita ja paljon muuta. (Loshin, 2003, ss. 257–259.)

HTTP (Hypertext Transfer Protocol) on sovelluskerroksessa toimiva protokolla, jonka avulla käyttäjän selain ja WWW-palvelin kommunikoivat keskenään (Loshin, 2003, ss. 260–261). Kaikki HTTP-liikenne on pyyntöjä (engl. requests) tai vastauksia (engl. responses). Asiakasohjelmat, kuten käyttäjien selaimet, pyytävät resursseja WWW-palvelimilta, jotka vastaavat asiakasohjelmien tekemiin resurssipyyntöihin esimerkiksi verkkosivustolla. (Loshin, 2003, ss. 121–122.)

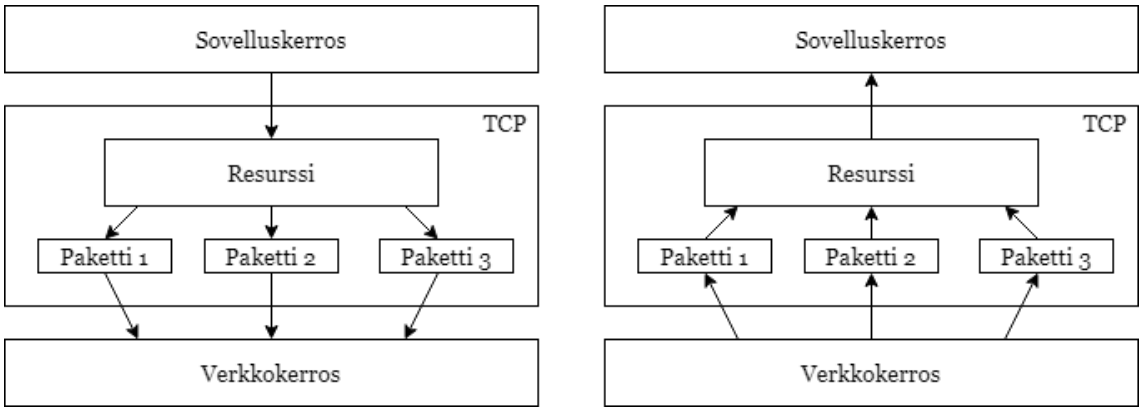
HTTP-protokollan voisi yksinkertaisuudessaan ajatella olevan aikaisemmin mainitussa postikorttiesimerkissä postikortissa oleva teksti, jolla Liisa kutsuu Petrin juhliin sekä pyytää Petriltä vastausta tähän kutsuun. Tässä esimerkissä HTTP-pyyntö olisi ”Vastaatko minulle, voitko osallistua juhliini”, johon Petri voisi vastata ”Kyllä, osallistun”. Vastaavasti selain voi lähettää pyynnön WWW-palvelimelle, jossa selain pyytää ”Vastaatko minulle tämän verkkosivuston”, johon WWW-palvelin lähettää verkkosivun ja vastaa ”Tässä on verkkosivu, jonka pyysit”.

HTTP-protokolla ei yksinään riitä lähettämään pyyntöjä ja vastauksia Internetissä. Jotta selain pystyy lähettämään pyynnön ja WWW-palvelin pystyy vastaamaan tähän pyyntöön, pitää kyseinen tieto saada toimitettua Internetissä selaimelta WWW-palvelimelle ja takaisin. HTTP-protokolla hyödyntää tässä tiedon siirtämisessä TCP-protokollaa, josta kerrotaan seuraavassa luvussa. (Loshin, 2003, s. 124.)

2.3 TCP

TCP (Transmission Control Protocol) on kuljetuskerrokseen kuuluva protokolla, jonka tehtävänä on varmistaa tiedon siirtyminen sovellukselta toiselle, esimerkiksi selaimelta WWW-palvelimelle ja sieltä takaisin (Goralski, 2009, s. 38).

Internetissä ei ole mahdollista siirtää kerrallaan isoja resursseja, kuten videoita, vaan resurssit jaetaan eli segmentoidaan (engl. segmentation) pienempiin paketteihin (engl. packet) eli segmentteihin (engl. segment), jotka lähetetään eteenpäin yksi kerrallaan. TCP:n tehtävänä on vastaanottaa lähettävältä sovelluskerrokselta tuleva data, joka tarvittaessa jaetaan pienempiin paketteihin ennen lähetystä. Vastaanottavassa päässä TCP kokoaa datan takaisin alkuperäiseen muotoon ja siirtää sen eteenpäin vastaanottavan sovelluskerroksen käsiteltäväksi. (Goralski, 2009, ss. 38–39; Kurose & Ross, 2013, s. 259.) Alla olevasta kuviosta näkee edellä olevan prosessin kuvattuna (KUVIO 2).



KUVIO 2 TCP-protokolla (Goralski, 2009, p. 39; Kurose & Ross, 2013, p. 259)

TCP-protokolla myös varmistaa, että kaikki paketit siirtyvät eheänä lähettävältä sovellukselta vastaanottavalle sovellukselle, esimerkiksi selaimelta WWW-palvelimelle. Tämä käytännössä tarkoittaa sitä, että lähettävässä päässä numeroidaan kaikki lähetettävät paketit. Vastaanottaja kuittaa lähettäjälle vastaanottamansa paketit, jotta lähettäjä tietää, mitkä paketit ovat saapuneet perille. Jos lähettäjä ei syystä tai toisesta saa yksittäisestä paketista saapumiskuittausta, lähetetään paketti uudelleen, jotta kaikki paketit tulevat varmasti perille. (Goralski, 2009, ss. 289–291; Kurose & Ross, 2013, ss. 268–272.)

Jokaiseen pakettiin lisätään myös mukaan paketin tarkastussumma (engl. checksum), jonka avulla vastaanottaja voi tarkistaa, tuliko kyseinen paketti eheänä vai virheellisenä perille, esimerkiksi puuttuuko paketista osia (Goralski, 2009, ss. 284–285; Kurose & Ross, 2013, ss. 468–469). Jos paketti ei vastaa paketin mukana tullutta tarkastussummaa, hylkää vastaanottaja virheellisen paketin. Tämä vastaa tilannetta, ettei paketti olisi koskaan saapunut perille, jolloin paketti lähetetään uudelleen. (Goralski, 2009, ss. 284–285.)

Tarvittaessa lähettävä pää voi lähettää samoja paketteja useita kertoja uudestaan, jos paketit eivät saavu perille tai saapuvat perille virheellisinä, tai jos lähettävä pää ei saa kuittauksia vastaanotetuista paketeista jostain muusta syystä. Näin TCP-protokolla käytännössä yrittää varmistaa sen, että siirrettävä tieto siirtyy eheänä lähettäjältä vastaanottajalle. (Goralski, 2009, ss. 289–291; Kurose & Ross, 2013, ss. 268–272.)

Yksinkertaisuudessaan TCP-protokollan voisi ajatella postiesimerkissä olevan postin virheenhallintajärjestelmä, joka varmistaa postilaatikkoon saapuvien postikorttien tai pakettien siirtymisen postilaatikosta vastaanottavalle taholle, jotta kaikki varmasti saapuu perille muuttumattomana. TCP-protokollan tehtävänä on siis virheenhallinta ja isojen resurssien pilkkominen, mutta pakettien siirtäminen Internetissä ei kuulu TCP-protokollan tehtäviin.

Pakettien siirtämisestä Internetissä vastaa verkkokerroksella toimiva IP-protokolla, johon perehdytään myöhemmin IP-protokollaa käsittelevässä luvussa. Sitä ennen käsitellään muutamia teknologioita, joiden avulla IP-protokolla osaa toimittaa paketit oikeaan paikkaan Internetissä. Seuraava luku esitteleekin URL:ää, jolla kerrotaan halutun verkkosivuston sijainti Internetissä.

2.4 URL

Internetiä käyttävät ohjelmistot pyytävät ja lähettävät erilaisia sähköisiä resursseja, joiden pyytämisessä hyödynnetään URL:ia (Uniform Resource Locator). URL on teksti, jonka käyttäjä kirjoittaa selaimen osoiteriviin, kun hän haluaa ladata jonkin verkkosivun, ja jonka avulla kerrotaan pyydetyn resurssin sijainti Internetissä. (Goralski, 2009, ss. 565–566.)

URL:n avulla selaimen lähettämä pyyntö, joka voi olla esimerkiksi verkkosivu, osataan ohjata oikealle WWW-palvelimelle. URL:n avulla WWW-palvelin myös tietää, mitä resurssia selain on pyytänyt. Pyynnön saatuaan WWW-palvelin vastaa selaimelle takaisin pyydetyn resurssin. URL:n voi siis mieltää esimerkiksi postikortissa olevaksi postiosoitteeksi, jonka avulla posti osaa toimittaa postikortin oikeaan sijaintiin. (Loshin, 2003, ss. 257–259.)

URL:n muoto voi olla esimerkiksi seuraavanlainen: http://esimerkki.fi. Tässä esimerkissä http-teksti on skeema (engl. scheme), joka on myös käytettävän pyynnön protokolla, joka on aikaisemmin esitelty HTTP-protokolla. Tämän jälkeen tuleva esimerkki.fi-teksti on käytettävä verkkotunnus (engl. domain), joka kertoo WWW-palvelimen, johon pyyntö pitäisi Internetissä toimittaa. (Loshin, 2003, ss. 259–260.) Seuraavassa luvussa perehdytään tarkemmin näihin verkkotunnuksiin.

2.5 Verkkotunnus

Verkkotunnus on palvelun tekstimuotoinen osoite (Traficom, 2019c). Tutkimuksessa jaetaan verkkotunnukset kahteen eri kategoriaan, jotka ovat pääverkkotunnus (engl. main domain), kuten ylemmässä luvussa esitelty esimerkki.fi-verkkotunnus, ja aliverkkotunnus (engl. subdomain), joka esitellään tässä luvussa.

Jokaiseen verkkotunnukseen kuuluu ylätason verkkotunnus (engl. top-level domain, TLD), joka löytyy verkkotunnuksen lopusta, kuten esimerkki.fi-osoitteessa ylätason verkkotunnus on fi-pääte. Näitä ylätason verkkotunnuksia on useita erilaisia käytössä, esimerkiksi com, net ja org. Jokaista ylätason verkkotunnusta hallinnoi oma toimijansa, kuten yksityis- tai julkisoikeudellinen säätiö tai puolueeton yhteisö, joka ylläpitää ylätason verkkotunnusta omien lakiensa ja sopimusten mukaisesti. Monelta maalta löytyy myös oma ylätason verkkotunnus, joka on Suomella fi-pääte, jota hallinnoi Liikenne- ja viestintävirasto Traficom. (Traficom, 2019a.)

Verkkotunnukseen kuuluu ylätason verkkotunnuksen lisäksi myös toinen taso, joka on esimerkki.fi-osoitteessa sana esimerkki. Kaikki verkkotunnukset ovat uniikkeja, eikä täysin samanlaista verkkotunnusta voi olla käytössä kahdessa eri palvelussa. Eri palveluita voidaankin tarjota esimerkiksi vaihtamalla ylätason verkkotunnusta, kuten esimerkki.fi ja esimerkki.com, tai vaihtamalla toisen tason tekstiä, kuten esimerkki1.fi ja esimerkki2.fi. (Loshin, 2003, s. 152.)

Taho, kuten yritys, pystyy lähtökohtaisesti rekisteröimään sille sopivimman fi-päätteisen pääverkkotunnuksen, kunhan vain kyseinen pääverkkotunnus on vapaa, eikä sitä ole rekisteröity toiselle taholle. (Elisa, 2021; Traficom, 2019b). Muiden kuin fi-päätteisten pääverkkotunnusten hankkimiseen voi olla erilaisia kriteerejä, joita säätelee ylätason verkkotunnusta hallinnoiva toimija omien lakiensa ja sopimustensa mukaisesti (Traficom, 2019a; Wilson, 2020).

Pääverkkotunnukseen voidaan myös liittää aliverkkotunnuksia, kuten esimerkki.fi-osoitteeseen sivu1.esimerkki.fi tai sivu2.esimerkki.fi (Loshin, 2003, s. 152). Aliverkkotunnuksien avulla taho pystyy tarjoamaan useampia palveluita käyttäjilleen, jolloin tahon ei tarvitse hankkia useampia pääverkkotunnuksia (Rashid ym., 2019, s. 1). Aliverkkotunnukset voivat olla myös monitasoisia, esimerkiksi xx.yy.sivu1.esimerkki.fi (Loshin, 2003, s. 152), minkä takia yhteen pääverkkotunnukseen liitettyjä aliverkkotunnuksia voi olla huomattava määrä erilaisia (Rashid ym., 2019, s. 1).

Verkkotunnuksen tarkoituksena on helpottaa käyttäjiä, jottei heidän tarvitse muistaa ulkoa palveluiden käyttämiä monimutkaisia IP-osoitteita (Traficom, 2019c), joista kerrotaan seuraavassa luvussa.

2.6 IP

IP (Internet Protocol) on verkkokerroksen protokolla ja sen tehtävänä on siirtää paketteja Internetissä (Kurose & Ross, 2013, s. 77). IP-protokollan voisi mieltää Internetin postiksi; tahoksi, joka siirtää postikorttiesimerkissä Liisan lähettämän postikortin postilaatikosta vastaanottajalle eli Petrille.

IP-protokollaan kuuluu myös yllä mainittu IP-osoite, jonka avulla verkkolaitteet yksilöidään ja tunnistetaan Internetissä (Kurose & Ross, 2013, s. 365). Rooney (2010, ss. 1–2) mainitseekin teoksessaan, että Internetiin kytketyt verkkolaitteet ovat kuin puhelimia: jokaisessa puhelimessa on yksilöity puhelinnumero, jonka avulla puhelimella pystyy ottamaan toiseen puhelimeen yhteyttä (Rooney, 2010, ss. 1–2). Vastaavasti Internetissä jokaisella verkkolaitteella pitää olla oma yksilöity IP-osoite, jotta se pystyy kommunikoimaan Internetissä oikein muiden verkkolaitteiden kanssa (Loshin, 2003, ss. 399–400; Rooney, 2010, ss. 5–6).

IP-osoitteista on käytössä kaksi eri versiota IPv4 (IP versio 4) ja IPv6 (IP versio 6) (Kurose & Ross, 2013, s. 382), joista ensimmäisenä mainittua käsittelemme tässä tutkimuksessa. IPv4-osoite on 32-bittinen luku, joka tarkoittaa sitä, että IP-osoite on pituudeltaan 32 numeroa, jotka ovat joko ykkösiä tai nollia (Goralski, 2009, s. 119). Nämä 32 bittiä yleensä esitetään ihmiselle helpommin ymmärrettävässä muodossa, joka kuvataan neljän tavun eli 8 bitin kokonaislukuna pisteellä erotettuna, esimerkiksi IP-osoite 01111111.00000000.00000000.00000001 vastaisi selkeämmin ymmärrettävässä muodossa IP-osoitetta 127.0.0.1. (Rooney, 2010, ss. 26–29) Jokaiseen IP-protokollan lähettämään pakettiin lisätään vastaanottajan IP-osoite, jotta paketti osataan ohjata oikealle verkkolaitteelle Internetissä (Kurose & Ross, 2013, ss. 360–361).

Yksinkertaisuudessaan IP-protokolla toimii seuraavalla tavalla. Lähettävän pään IP-protokolla saa paketin kuljetuskerrokselta, esimerkiksi TCP-protokollalta (Goralski, 2009, s. 35). Tämän jälkeen IP-protokolla toimittaa paketin Internetissä olevalle vastaanottavalle verkkolaitteelle hyödyntäen pakettiin lisättyä vastaanottajan IP-osoitetta (Rooney, 2010, s. 3). Vastaanottavassa päässä IP-protokolla siirtää paketin eteenpäin vastaanottavan pään kuljetuskerrokselle (Goralski, 2009, s. 35). Tätä toimintaa selkeyttää TCP-luvussa (LUKU 2.3) esitelty kuvio (KUVIO 2).

Seuraavassa luvussa käsittelemme Internetin nimipalvelujärjestelmää, jonka tehtävänä on selvittää verkkotunnukselle kuuluva IP-osoite pakettien toimittamista varten Internetissä.

2.7 DNS

Annoimme URL:ia käsittelevässä luvussa (LUKU 2.4) esimerkin, jossa selain pyysi resurssia esimerkki.fi-verkkotunnusta hallinnoivalta WWW-palvelimelta.

Todellisuudessa Internetissä pakettien osoitteena ei käytetä URL:ssa olevaa verkkotunnusta, vaan verkkotunnukselle pitää hakea sitä vastaava IP-osoite, jolla paketit löytävät perille Internetissä (Goralski, 2009, s. 483). Tämän IP-osoitteen hakemiseen hyödynnetään Internetin nimipalvelujärjestelmää eli DNS:ää (Domain Name System), joka toimii HTTP-protokollan tapaan sovelluskerroksella (Rooney, 2010, s. 8).

Internetissä pakettien toimittamisen voi mieltää puhelimella soittamisena kaverille. Harva varmaankaan enää nykyään muistaa ulkoa kavereidensa puhelinnumeroita, vaan puhelinnumerot on tallennettu puhelimeen, josta ne etsitään kaverin nimen perustella. Vastaavasti Internetissäkin toimii verkkotunnus, IP-osoite ja DNS. DNS:n voisi ajatella olevan Internetin puhelinluettelo, josta haetaan puhelinnumero nimen perusteella. DNS:stä hakemisessa nimenä käytetään verkkotunnusta ja numerona vastaanotetaan IP-osoite. Jokaiselle käytössä olevalle verkkotunnukselle löytyy sitä vastaava IP-osoite, joka lisätään jokaiseen lähetettävään segmenttiin, jotta paketti osataan ohjata oikeaan osoitteeseen Internetissä. (Rooney, 2010, ss. 8–9.) Vastaavalla tavalla jokaiseen postikorttiinkin pitää lisätä vastaanottajan osoite, jotta posti osaa toimittaa postikortin oikeaan osoitteeseen.

DNS toimii yksinkertaisuudessa niin, että Internetiä käyttävä ohjelmisto, esimerkiksi selain, poimii URL:sta verkkotunnuksen, josta lähetetään DNS-palvelimelle kysely, johon DNS-palvelin vastaa takaisin verkkotunnuksen IP-osoitteella (Kurose & Ross, 2013, s. 157). Tämä toimisi esimerkiksi näin, ”kerro minulle esimerkki.fi verkkotunnusta hallinnoivan WWW-palvelimen IP-osoite”, johon DNS-palvelin vastaisi, ”tässä on pyytämäsi IP-osoite” sekä WWW-palvelimen IP-osoite liitettynä vastaukseen.

Seuraavassa luvussa yhdistetään kaikki aikaisemmin mainitut teknologiat ja kerrotaan, miten ne toimivat keskenään.

2.8 Yhteenveto Internet-luvusta

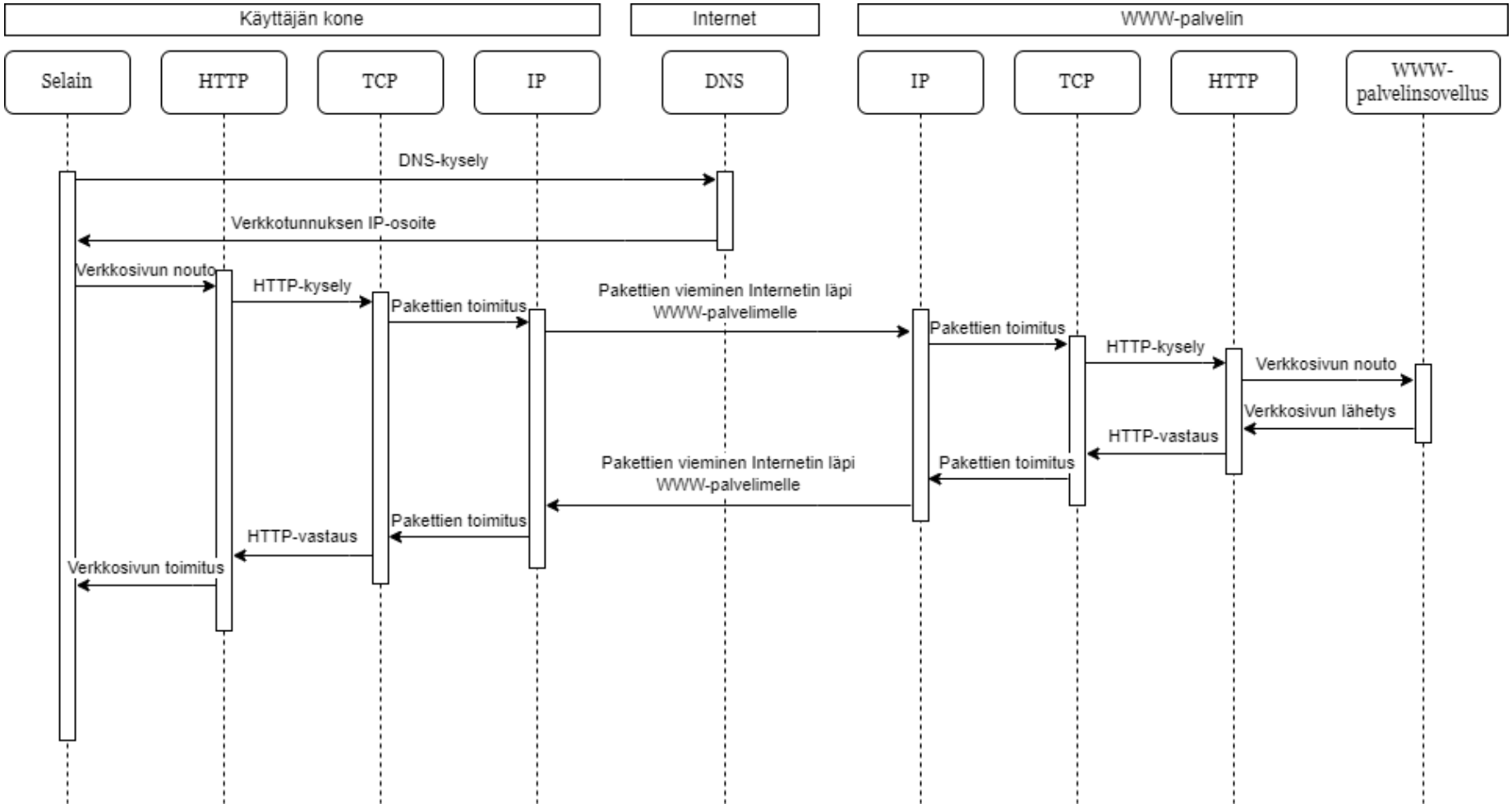
Tässä luvussa kuvataan, kuinka edellä kuvatut teknologiat toimivat yhteistyössä käytännön esimerkin avulla. Seuraavassa esimerkissä kerrotaan, kuinka käyttäjä saa WWW-palvelimelta pyytämänsä verkkosivun selaimeensa:

1. Käyttäjä kirjoittaa selaimen osoitepalkkiin halutun verkkosivun URL:n, kuten http://sivu1.esimerkki.fi ja pyytää selainta hakemaan kyseisen verkkosivun
2. Pyydetystä URL:sta poimitaan verkkotunnus sivu1.esimerkki.fi, josta lähetetään kysely DNS:lle
3. DNS vastaa takaisin verkkotunnuksen IP-osoitteen, kuten 127.0.0.1
4. IP-osoite otetaan vastaan ja se liitetään mukaan lähetettävään HTTP-pyyntöön halutusta verkkosivusta

5. TCP-protokolla tarvittaessa segmentoi lähetettävän HTTP-pyynnön pienempiin osiin sekä lisää jokaiseen pakettiin järjestysnumeron ja tarkastussumman
6. TCP-protokolla antaa lähetettävät paketit IP-protokollan toimitettavaksi
7. IP-protokolla toimittaa lähetettävät paketit vastaanottavalle WWW-palvelimelle käyttäen DNS:ltä saatua IP-osoitetta
8. Vastaanottavassa päässä IP-protokolla siirtää kaikki saapuneet paketit vastaanottavan pään TCP-protokollalle käsiteltäväksi
9. Kaikkien pakettien eheys tarkastetaan käyttäen tarkastussummaa ja kaikki eheät vastaanotetut paketit kuitataan vastaanotetuiksi
10. TCP-protokolla kokoaa HTTP-pyynnön alkuperäiseen muotoon ja siirtää sen sovelluskerrokselle
11. Sovelluskerroksella toimiva WWW-palvelinsovellus vastaanottaa HTTP-pyynnön ja prosessoi pyyntöä varten vastauksen
12. WWW-palvelinsovellus lähettää verkkosivuston HTTP-vastauksella
13. Vastaus tulee WWW-palvelimen TCP-protokollalle, joka segmentoi vastauksen, numeroi ja lisää tarkastusnumerot sekä antaa sen IP-protokollan käsiteltäväksi
14. IP-protokolla toimittaa vastauksen verkkosivun pyytäjän TCP-protokollalle, joka tarkistaa tarkistussummat ja kuittaa vastaajalle eheät saapuneet paketit
15. TCP-protokolla kokoaa pilkotut paketit takaisin alkuperäiseen muotoon ja siirtää paketit sovelluskerroksessa toimivalle selaimelle
16. Kun HTTP-vastaus on kokonaisuudessaan saapunut, selain käsittelee saapuneen verkkosivuston, joka näytetään selaimessa käyttäjälle

Edellä oleva ketju löytyy esitettynä alla olevassa kuviossa (KUVIO 3). Näin yksinkertaisuudessaan verkkoselain ja WWW-palvelin keskustelevat keskenään Internetissä ilman salausta.

Seuraavassa luvussa käsitellään, miten selaimen ja WWW-palvelimen välinen keskustelu salataan, jotta se on suojattuna kolmansilta osapuolilta, jotka voivat esimerkiksi salakuunnella tai muokata tätä keskustelua.



KUVIO 3 Verkkosivun noutaminen käyttäen TCP/IP-pinoa

3 HTTPS

HTTPS (Hypertext Transfer Protocol Secure) on salattu versio HTTP-protokollasta, jonka tarkoitus on välittää salatusti tietoa WWW-palvelimen ja verkkoselaimen välillä. HTTPS-liikenteen salaaminen tapahtuu käytännössä nykypäivänä TLS-protokolalla, josta kerrotaan lisää tulevissa alaluvuissa. Verkkoliikenne tulee aina salata, jotta saadaan kasvatettua tietoturvaa. Verkkoliikenteen salaamisen edut voidaan tiivistää kolmeen eri kohtaan, jotka ovat salakuuntelemisen estäminen, verkkoliikenteen eheyden varmistaminen ja verkkosivun identiteetin todentaminen. (Google Developers, 2022.)

HTTPS on suunniteltu turvaamaan verkkoliikenne salakuuntelulta ja aktiiviselta hyökkäykseltä. Salaamatonta verkkoliikennettä voidaan helposti passiivisesti salakuunnella lähiverkosta, joka mahdollistaa istunnon varastamisen. HTTPS suojaa käyttäjää myös aktiivista hyökkäystä vastaan, jossa vihamielinen hyökkääjä yrittää saada käyttäjän käyttämään väärentämäänsä sivua. Nykypäivänä verkkoselaimet ovat kehittyneet sen verran, että aktiivinen hyökkäys ei ole mahdollista ilman selaimen tuottamaa varoitusta käyttäjälle. (Jackson & Barth, 2008, s. 525.)

HTTPS estää myös verkkoliikenteen muokkaamisen takaamalla WWW-palvelimen ja verkkoselaimen välisen yhteyden eheyden. Aikaisemmin on mm. tehty tutkimusta siitä, että Internet-palveluntarjoajat ovat muokanneet välissä verkkoliikennettä ja lisänneet ulkopuolisille sivuille omia mainoksiaan, jotka ovat olleet yksi osa heidän ansaintamenetelmäänsä (Reis ym., 2008, s. 31). Verkkoliikenteen muokkaaminen on mahdollista salaamattomassa HTTP-liikenteessä, mutta turvallinen HTTPS estää verkkoliikenteen muokkaamisen (Reis ym., 2008, s. 38). Salaamatonta verkkosivua on siis mahdollista muuttaa tavalla, jota sen ylläpitäjät eivät haluaisi käyttäjilleen tarjota. Tämä kertoo sen, että salaaminen ei ole tärkeätä vain verkkosivuilla, joissa käsitellään arkaluonteista tietoa, vaan salausta tulisi käyttää kaikissa verkkosivustoissa.

Salaamisen kolmas hyvä puoli on siinä, että se mahdollistaa sivuston identiteetin todentamisen, joka tapahtuu hyväksikäyttäen varmenteita. Varmenteella mahdollistetaan verkossa palvelujen tunnistaminen, mikä tapahtuu sitomalla

varmenteeseen verkkosivua yksilöivä tunniste (Oppliger, 2016, s. 201). Varmenteista kerrotaan lisää alaluvussa 3.3.

Aikaisemmin tutkimuksissa on etsitty syitä sille, miksi HTTPS ei aina ole ollut käytössä tai miksi sitä ei ole asianmukaisesti konfiguroitu. On havaittu, että normaali käyttäjä aliarvioi salatun HTTPS-verkkoliikenteen merkitystä, kun taas palveluiden ylläpitäjillä on yleisesti hyvä ymmärrys siitä, mikä HTTPS on ja mitä sillä voidaan suojata (Krombholz ym., 2019, s. 246). Vaikka ylläpitäjillä on hyvä ymmärrys HTTPS:stä, verkossa on silti paljon palveluita, jotka ovat heikosti salattuja.

SSL Labs -verkkosivusto julkaisee tilastoja maailman tunnetuimpien verkkosivujen SSL/TLS-salauksesta. Sivuston julkaiseman tilaston mukaan 20.9.2021 tutkituista verkkosivuista jopa 52,8 % oli riittämättömästi salattuja (SSL Labs, 2021). Krombholz ym. (2017, s. 1347) tutkivat HTTPS salauksen käyttöönoton käytettävyyttä opiskelijoilla ja he havaitsivat, että sen käyttöönotto on hankalaa jopa asiantunteville käyttäjille. Krombholz ym. (2017, s. 1347) listasivat tutkimuksessaan useita eri ongelmakohtia, joita olivat mm. parhaiden käytänteiden puute, heikot oletusasetukset, konfiguraatiotiedostojen sekava rakenne, monimutkainen konfiguraatioprosessi ja se, että käyttöönotto vaatii liian paljon taustatietoa.

Seuraavissa alaluvuissa käsitellään TLS-protokolla ja sen vanhempaa SSL -protokollaversiota, varmenteita, SSL/TLS-protokollan heikkouksia sekä luodaan tutkimusosuutta varten SSL/TLS-salauksen suositukset. Alla oleva luku käsittelee SSL/TLS-protokollan historiaa, jotta ymmärretään tarkemmin, miksi SSL/TLS-salauksesta on tullut yksi verkkoliikenteen turvallisuuden kulmakivistä.

3.1 SSL/TLS-protokollan historia

Verkkosivujen yleistyessä 1990-luvun alussa yleistyivät myös verkkosivujen kautta tapahtuvat kaupankäynnit. Kaupankäyntiin kuului se, että oston yhteydessä ostava taho luovutti omia maksutietojaan, kuten luottokorttitietoja, verkkosivujen kautta myyjälle, jotta myyjä pystyi veloittamaan ostajalta ostohinnan. Moni suhtautuikin varauksella maksutietojen luovuttamiseen verkkosivujen kautta, minkä takia moni yritys ja tutkija alkoi selvittää turvallisempia tapoja toteuttaa kaupankäyntiä verkkosivuilla. (Oppliger, 2016, s. 11.)

Tähän aikaan ei ollut vielä selkeää konsensusta, miten ja millä tekniikalla verkkosivujen käyttämisestä tehtäisiin turvallisempaa, mutta yksimielisiä oltiin siinä, että siihen pitäisi käyttää jonkinlaista salausmenetelmää. Konsensuksen löytämisen haasteena oli se, että salauksen toteuttamiseen löytyy monia eri tapoja ja tekniikoita. (Oppliger, 2016, ss. 11–12.)

IETF-organisaatio (Internet Engineering Task Force) kehitti vuonna 1994 oman salausprotokollansa S-HTTP (Secure Hypertext Transfer Protocol), jonka spekuloitiin olevan tulevaisuudessa tärkeä osa HTTP-liikenteen salausta (Oppliger, 2016, s. 12). Samana vuonna Netscape Communications julkaisi yhti-

ön sisäiseen käyttöön oman salausprotokollansa SSL (Secure Sockets Layer). Ensimmäisessä SSL-versiossa oli muutamia vikoja ja puutteita, jotka piti korjata, jotta SSL-protokolla voitiin ottaa julkiseen käyttöön. Netscape Communications korjasikin nämä SSL-version 1.0 ongelmat ja julkaisi SSL-protokollasta version 2.0 sekä uuden Netscape Navigator -selaimen, joka tuki tätä uutta salausprotokollaa. (McKay & Cooper, 2019, s. 1; Oppliger, 2016, ss. 12–14.)

Uuden SSL-protokollan ja sitä tukevan uuden Netscape Navigator -selaimen ansiosta verkkosivujen käyttö kasvatti suosiotaan, mikä sai useamman yhtiön miettimään mahdollisia menetyksiä markkinoilla, jos he eivät pysy teknologian kehityksessä mukana (Oppliger, 2016, s. 14). Näistä yhtenä yrityksenä oli Microsoft. Microsoft kehittikin 1995 vuoden puolivälissä oman salausprotokollansa PCT (Private Communication Technology), joka otettiin käyttöön Microsoftin uudessa Internet Explorer -selaimessa (McKay & Cooper, 2019, s. 1; Oppliger, 2016, s. 14). PCT-protokolla oli käsitteellisesti ja teknisesti todella samanlainen SSL-versio 2.0:n kanssa, minkä takia WWW-palvelimien oli helppo tukea molempia protokollia (Oppliger, 2016, s. 14).

PCT-protokolla toi HTTP-yhteyden salaamiseen uusia parannuksia, jotka otettiin käyttöön vuonna 1996 julkaistussa SSL-versio 3.0:ssa (Oppliger, 2016, s. 14). Uudessa SSL-versiossa korjattiin myös vanhasta 2.0 versiosta löytyneitä puutteita ja turvallisuusongelmia (McKay & Cooper, 2019, s. 1). Turvallisuusyhteisöissä oli paljon epäselvyyttä, koska PCT- ja SSL-protokollilla oli omat käyttäjäkuntansa. Tilannetta ei myöskään auttanut se, että Microsoft oli julkaissut toisen salausprotokollan STLP:n (Secure Transport Layer Protocol), joka oli käytännössä muunnelma SSL-protokollaversiosta 3.0 ja johon oli tuotu mukaan uusia ominaisuuksia. (Oppliger, 2016, ss. 15–16.)

Salausprotokollien sekavaan tilanteeseen haluttiin selkeytystä, minkä takia perustettiin IETF:n työryhmä, jonka tehtävänä oli standardoida yhtenäinen salausprotokolla, jonka nimeksi tulisi TLS (Transport Layer Security) (McKay & Cooper, 2019, s. 1; Oppliger, 2016, s. 16). Teknisesti tehtävä oli helpohko, koska SSL 3.0, PCT ja STLP olivat teknisesti samankaltaisia, mutta silti standardointi kesti kolme vuotta ja TLS-protokollan 1.0 versio julkaistiin vuonna 1999 (Oppliger, 2016, ss. 16–17). Nimen muutoksesta huolimatta uusi TLS-protokolla, oli pohjimmiltaan SSL 3.0, johon oli tuotu uusia parannuksia. (McKay & Cooper, 2019, s. 1; Oppliger, 2016, ss. 16–17). Tästä syystä Oppliger (2016, s. 17) mainitseekin, että TLS 1.0:aa saatetaankin kutsua nimellä SSL 3.1, vaikka se ei sitä virallisesti olekaan.

TLS-protokolla 1.0-version julkaisun jälkeen vuonna 1999 IETF:n työryhmä jatkoi TLS-protokollan kehittämistä. Vuonna 2006 julkaistiin uusi TLS-protokollan versio 1.1, mutta kyseisestä versiosta löytyi puutteita, jotka haluttiin korjata nopeasti. Tästä syystä jo vuonna 2008 julkaistiin uusi TLS-protokollan versio 1.2. (Oppliger, 2016, s. 16.)

IETF ilmoitti vuonna 2011 SSL-protokollaversion 2.0 sekä vuonna 2015 version 3.0 vanhentuneiksi ja turvattomiksi, eikä niiden käyttöä pitäisi jatkaa verkkosivustojen suojaamiseen (Barnes ym., 2015; Turner & Polk, 2011). TLS-protokollasta julkaistiin sen uusin versio 1.3 vuonna 2018 (Rescorla, 2018). Ky-

seiseen TLS-versioon korjattiin ongelmia, joita viimeisen kymmenen vuoden aikana oli noussut esille edellisiin versioihin liittyen (McKay & Cooper, 2019, s. 2). Versioista 1.0 ja 1.1 on löytynyt haavoittuvuuksia, minkä takia useampi toimija on lopettanut 1.0 ja 1.1. version tukemisen niiden turvattomuuden takia (Langley & Benjamin, 2021; Microsoft, 2021a; Mozilla, 2020a).

Alla olevaan taulukkoon on koottu kaikki olemassa olevat SSL- ja TLS-protokollan versiot, niiden julkaisuvuodet ja mikä niiden status on tänä päivänä (TAULUKKO 1).

TAULUKKO 1 SSL/TLS-protokollan versiot (Barnes ym., 2015; Langley & Benjamin, 2021; Microsoft, 2021a; Mozilla, 2020a; SSL Labs, 2021; Turner & Polk, 2011)

Protokolla	**Julkaistu**	**Lisätietoa**
SSL 1.0	1994 (epävirallinen)	Ei otettu julkiseen käyttöön
SSL 2.0	1995	Vanhentunut (2011)
SSL 3.0	1996	Vanhentunut (2015)
TLS 1.0	1999	Useampi toimija lopettanut tukemisen (2020)
TLS 1.1	2006	Useampi toimija lopettanut tukemisen (2020)
TLS 1.2	2008	Tällä hetkellä eniten käytetty
TLS 1.3	2018	Uusin TLS-versio

Yllä olevasta taulukosta näkeekin suhteellisen helposti, ettei nykypäivänä olisi suositeltavaa käyttää kuin vain versioita TLS 1.2 ja TLS 1.3, mutta tästä kerrotaan tarkemmin myöhemmin tulevassa SSL/TLS-salauksen suositukset luvussa (LUKU 3.6). Siirrytään seuraavaksi käsittelemään, miten tämä SSL/TLS-salausprotokolla toimii.

3.2 SSL/TLS-protokollan toiminta

SSL ja TLS ovat kryptografisia salausprotokollia, joiden tarkoitus on tarjota suojaa kryptografisin keinoin (Oppliger, 2016, s. 18). SSL-etuliitettä käytetään monesti edelleen, vaikka jo vuonna 2015 IETF:n julkaisema RFC7568 esitti sen viimeisimmän 3.0 version turvattomaksi ja vanhentuneeksi, eikä sen käyttöä tulisi näin ollen enää jatkaa (Barnes ym., 2015). Tästä huolimatta TLS- ja SSL-protokollat ovat hyvin samankaltaisia ja monessa kohdassa jopa identtisiä keskenään (Oppliger, 2016, s. 21).

Tässä tutkimuksessa keskitytään TLS-salauksen käyttöön WWW-palvelimien kanssa, mutta todellisuudessa TLS-salausta pystytään käyttämään vielä laajemmin. TLS-salaus ei ole sovellusriippuvainen ja sillä voidaan suojata minkä tahansa TCP-protokollaa hyödyntävän sovelluksen data. Myös UDP-protokollaan (User Datagram Protocol) perustuvan verkkoliikenteen salaaminen onnistuu DTLS-protokolalla (Datagram Transport Layer Security), mutta tässä tutkimuksessa ei syvennytä siihen. (Oppliger, 2016, s. 18.)

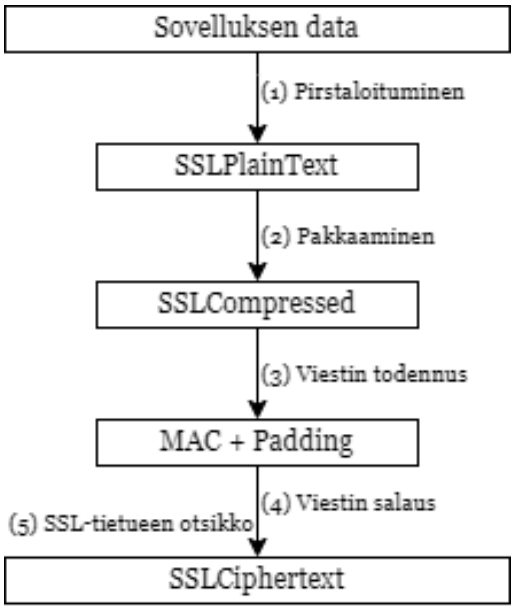
TLS-protokollalla on kolme perusominaisuutta, jotka ovat datan alkuperän todentaminen, yhteyden luottamuksellisuuden- ja yhteyden eheyden säilyttäminen. (McKay & Cooper, 2019, s. 3; Oppliger, 2016, s. 22)

Vertauskuvallisesti TLS-salausta voi verrata postikortin salaamiseen. Edellisessä TCP/IP-luvussa (LUKU 2.11.4) annetussa esimerkissä henkilö 1 lähetti postikortin henkilölle 2. Jos postikortti lähetetään ilman, että sitä suljetaan kirjekuoren sisään, voi kuka tahansa toimitusketjun aikana lukea, mitä postikortissa lukee. Jos postikortti suljettaisiin kirjekuoreen, jonka tässä esimerkissä olisi mahdollista saada auki vain postikortin saaja, postikortissa oleva informaatio pysyisi turvassa koko sen toimitusketjun ajan lähettäjältä saajalle. Samalla tapaa toimii myös pakettien salaus. (Loshin, 2003, s. 102.)

3.2.1 SSL/TLS-tietueprotokolla

SSL/TLS-protokolla koostuu SSL/TLS-tietueprotokollasta, joka pitää sisällään kolme eri aliprotokollaa, joita käytetään istunnon yhteyden hallitsemiseen. Kolme aliprotokollaa ovat kättely-, salaussarjan sovinta- ja hälytysprotokolla. Kättelyprotokollaa käytetään SSL/TLS istunnon parametrien kuten salaussarjojen sovittamiseen. Hälytysprotokollaa puolestaan käytetään virhetilanteiden viestimiseen. (McKay & Cooper, 2019, s. 4.)

Kokonaisuudessaan SSL/TLS-tietueiden käsittely koostuu aliprotokollien lisäksi viidestä eri vaiheesta, jotka ovat datan pirstoutuminen, pakkaaminen, viestin todennus, viestin salaus ja SSL/TLS-tietueen otsikko. (Oppliger, 2016, ss. 31–32). Seuraavassa kuviossa (KUVIO 4) havainnollistetaan SSL-protokolalla, mistä vaiheista tietueprotokolla koostuu.



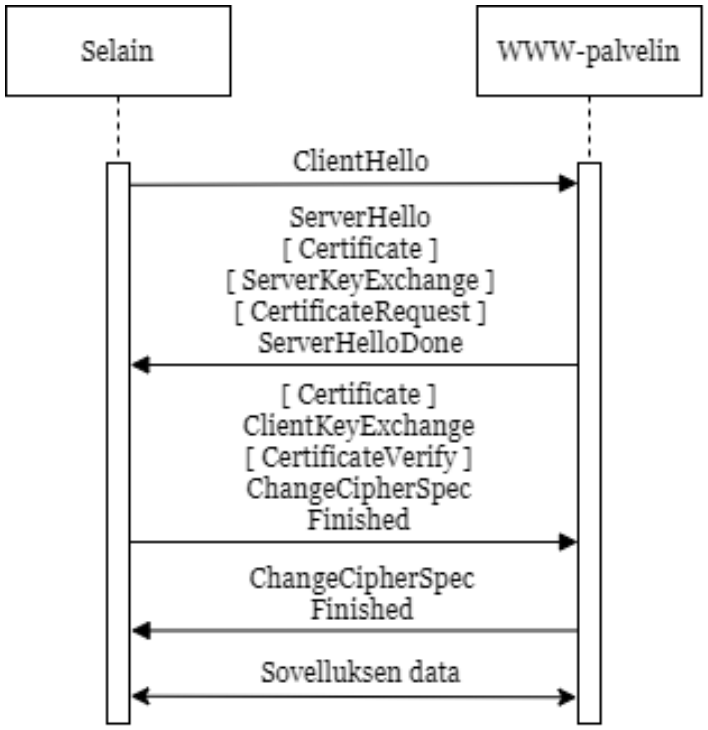
KUVIO 4 SSL-tietueen vaiheet (Oppliger, 2016, s. 32)

3.2.2 SSL/TLS-kättelyprotokolla

SSL/TLS-kättelyprotokolla on SSL/TLS-tietueprotokollan päälle kerrostettu prosessi, joka käynnistää SSL/TLS-salausta käyttävän viestintäistunnon. SSL/TLS-kättelyn aikana kaksi eri osapuolta, kuten selain ja WWW-palvelin vaihtavat viestejä kuitatakseen toisensa ja neuvotellakseen käytettävät salaussarjat (engl. cipher suites) ja istuntoavaimet. (Dierks & Rescorla, 2008, s. 33.)

SSL-kättelyprotokolla pitää sisällään neljä eri joukkoa viestejä, jotka vaihdetaan selaimen ja WWW-palvelimen välillä. Jokainen viestijoukko voidaan lähettää yhdessä TCP-segmentissä. Yhdestä yksittäisestä viestistä käytetään myös nimitystä lento, eli kokonaisuudessaan SSL/TLS-kättely pitää sisällään neljä eri lentoa. Virheet tai poikkeukset johtavat tässä ketjussa aina yhteyden katkaisemiseen. (Oppliger, 2016, ss. 46–47.)

Seuraavassa kuviossa (KUVIO 5) on havainnollistettu, mitä eri tietoja SSL/TLS-kättelyprotokollan eri vaiheissa vaihdetaan selaimen ja WWW-palvelimen välillä.



KUVIO 5 SSL/TLS-kättelyprotokolla (Oppliger, 2016, s. 47)

1. Selain lähettää WWW-palvelimelle *ClientHello*-viestin, jonka mukana tulee mm. selaimen viimeisin tuettu TLS-versio, satunnainen luku, selaimen tukemat salausalgoritmit ja pakkausmenetelmät.
2. WWW-palvelin vastaa *ClientHello*-viestiin *ServerHello*. *ClientHello*:n tapaan *ServerHello*-viesti pitää sisällään WWW-palvelimen valitseman vahvimman yhteisesti tuetun TLS-version, palvelimen generoiman sa-

tunnaisen luvun, palvelimen valitseman salausalgoritmin ja pakkausmenetelmän.
3. Välittömästi *ServerHello*-viestin jälkeen tulevat viestit *Certificate* ja *ServerHelloDone*. *Certificate* on WWW-palvelimen X.509 formaatissa oleva digitaalinen varmenne, joka sisältää myös palvelimen julkisen avaimen. Palvelimen tämän lennon viestit loppuvat *ServerHelloDone*-viestiin.
4. Seuraava vaihe on *ClientKeyExchange,* jossa selain lähettää väliaikaisen salaisuuden WWW-palvelimelle salaamalla sen palvelimen varmenteen mukana tulevalla julkisella avaimella.
5. WWW-palvelin vastaanottaa väliaikaisen salaisuuden ja sekä selain että WWW-palvelin luovat yleisen salaisuuden.
6. Selain lähettää *ChangeCipherSpec*-viestin, jossa se kertoo alkavansa käyttää uutta istuntoavainta tiivisteissä ja viestien salaamisessa. Viesti loppuu *Finished*-viestiin.
7. WWW-palvelin vastaanottaa selaimen *ChangeCipherSpec*-viestin ja muuttaa tilan symmetriseen salaukseen käyttäen hyväksi istuntoavainta. WWW-palvelin lopettaa lennon *Finished*-viestiin.
8. Tässä kohden kättely on hoidettu ja sekä selain että WWW-palvelin voivat jatkaa keskustelua salattua kanavaa pitkin. (Dierks & Rescorla, 2008, ss. 34–35.)

3.2.3 Salausalgoritmit ja salaussarjat

SSL/TLS suojaa verkkoliikennettä käyttäen hyväksi kryptografisia algoritmeja, jotka varmistavat verkkoliikenteen luottamuksellisuuden, eheyden ja aitouden. Käytettävät salausalgoritmit sovitaan TLS-kättelyn aikana, jossa selaimen ja WWW-palvelimen on löydettävä yhteinen tuettu salausalgoritmi. (McKay & Cooper, 2019, s. 14.)

Kryptografisia algoritmeja on tuhansia erilaisia, mutta kaikki ne voidaan lajitella kolmeen eri ryhmään: salaus-, allekirjoitus- ja tiivistealgoritmit. Salausalgoritmeja käytetään datan salaamiseen, mikä tuottaa luottamuksellisuuden. Allekirjoitusalgoritmin tarkoituksena on luoda digitaalinen allekirjoitus, joka varmistaa datan aitouden. Viimeistä tiivistealgoritmia käytetään datan eheyden varmistamiseen. (Shinder & Cross, 2008.)

Tiivistealgoritmilla on keskeinen rooli salaussarjoissa. Tiivistealgoritmi ottaa vastaan määräämättömän pituisen merkkijonon ja tekee siitä kiinteän pituisen tiivistearvon. Tarkoituksena on saada luotua yksilöllinen tunnistettavissa oleva merkkijono tiiviissä muodossa. Tiivistealgoritmit eivät rajoita vastaanotettavan merkkijonon pituutta. Tällöin lopputuloksena syntyy kiinteän pituinen tiivistearvo, jossa samaan lopputulokseen voidaan päätyä useilla eri merkkijonoilla. Näitä samoja tiivistearvon lopputuloksia kutsutaan yhteentörmäykseksi. (Menezes ym., 1996, s. 321.)

Tiivisteitä käytetään datan eheyden varmistamisessa, jossa tiivistearvoa hyväksikäyttäen luodaan digitaalinen allekirjoitus. Tätä erillistä käyttötapaa kutsutaan myös nimellä viestien todennuskoodit (engl. message authentication

code, MAC). Viestin allekirjoitus toimii siten, että allekirjoitukseen tarvitaan itse viesti sekä salainen avain ja näistä kahdesta parametrista muodostetaan tiivistearvo. Samaa tiivistearvoa ei täten pystytä luomaan ilman, että tiedetään salaista avainta, ja tämä takaa datan eheyden. (Menezes ym., 1996, ss. 321–322.)

TLS 1.2 -protokollassa ja sitä edeltävissä versioissa salaussarjat muodostuvat kolmesta eri algoritmista: avaimen vaihto-, salaus- ja viestin todennus -algoritmi. Yhdessä nämä kolme algoritmia ilmaistaan muodossa *TLS_AvaimenVaihtoAlg_WITH_SalausAlg_ViestinTodennusAlg*.

Esimerkiksi yksi salaussarja on *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA*, joka pitää sisällään Diffie-Hellman -avaimenvaihtoalgoritmin (ECDHE) yhdessä RSA-salauksen kanssa. AES_128_CBC-salausalgoritmi tarjoaa tässä salaussarjassa luottamuksellisuuden, ja viestin todennus tapahtuu HMAC_SHA-tiivistealgoritmilla (McKay & Cooper, 2019, s. 14.). Muutoksena edellisiin TLS 1.3 -protokollan tuetuissa salaussarjoissa ei enää erikseen mainita avaimenvaihtoalgoritmia (McKay & Cooper, 2019, s. 14).

SSL 3.0 tuki yhteensä 31 eri salaussarjaa (A. Freier, 2015, ss. 47–48), kun taas TLS 1.2 tarjosi alkuaan 37 eri salaussarjaa (Dierks & Rescorla, 2008, ss. 75–76). Seuraavassa taulukossa (TAULUKKO 2) on otanta eri salaussarjoista, joita TLS 1.2 -protokollassa on käytössä.

TAULUKKO 2 Otanta TLS 1.2 -protokollan salaussarjoista (Dierks & Rescorla, 2008, s. 83)

Cipher Suite	**Key Exchange**	**Cipher**	**Mac**
TLS_RSA_WITH_AES_128_CBC_SHA	RSA	AES_128_CBC	SHA
TLS_RSA_WITH_AES_256_CBC_SHA	RSA	AES_256_CBC	SHA
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA	AES_128_CBC	SHA256
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA	AES_256_CBC	SHA256
TLS_DH_RSA_WITH_AES_128_CBC_SHA	DH_RSA	AES_128_CBC	SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA	DHE_RSA	AES_128_CBC	SHA
TLS_DH_RSA_WITH_AES_256_CBC_SHA	DH_RSA	AES_256_CBC	SHA
TLS_DHE_DSS_WITH_AES_256_CBC_SHA	DHE_DSS	AES_256_CBC	SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA	DHE_RSA	AES_256_CBC	SHA
TLS_DH_RSA_WITH_AES_128_CBC_SHA256	DH_RSA	AES_128_CBC	SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_SHA256	DHE_RSA	AES_128_CBC	SHA256
TLS_DH_RSA_WITH_AES_256_CBC_SHA256	DH_RSA	AES_256_CBC	SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_SHA256	DHE_RSA	AES_256_CBC	SHA256

Aikaisemmassa tutkimuksessa (Kotzias ym., 2018, s. 415) on huomattu, että selaimet ottavat hyvinkin nopeasti käyttöön uusia salausalgoritmeja, mutta samalla ne ovat hitaita poistamaan tuen vanhoista. Tämä pääteltiin johtuvan siitä, että selaimet eivät uskalla pudottaa tukea vanhoista, koska ne pelkäävät estävänsä sillä monen verkkosivuston toimivuuden, jolloin käyttäjät voisivat vaihtaa toiseen selaimeen, jolla ne yhä toimivat. Samalla huomattiin, että WWW-palvelimien ylläpitäjät pelkää rikkovansa tuen vanhoihin selaimiin päivittämäl-

lä ohjelmaversiota. (Kotzias ym., 2018, s. 425.) Tämä taaksepäin yhteensopivuuteen liittyvä ongelma johtaa siihen, että käytetään vanhoja ohjelmistoversioita.

Seuraavaksi käymme läpi julkisen avaimen varmenteen, joka ei itsessään ole osa SSL/TLS-protokollaa, mutta kuuluu osaksi verkkosivun HTTPS-salausta.

3.3 Julkisen avaimen varmenne

Julkisen avaimen varmenteella mahdollistetaan verkossa olevien palvelujen tunnistaminen, mikä tapahtuu sitomalla varmenteeseen jokin järjestelmää yksilöivä tunniste tai myös mahdollisia lisätunnisteita (Oppliger, 2016, s. 201). Verkkoliikenteen salauksessa tämä tarkoittaa, että pitää olla keino, jolla varmenne voidaan sitoa verkkosivuun. Käytännössä verkkosivun varmenteen tulee pitää sisällään julkinen avain ja verkkotunnus (varmenteen yleinen nimi), jonka allekirjoittajana toimii jokin luotettu välittäjä. (Davies, 2010, s. 223.)

SSL/TLS-protokolla ei vaadi julkisen avaimen varmennetta, minkä vuoksi varmenteiden hallinta ei ole osa SSL/TLS-protokollaa, vaan varmenteet ovat osa julkisen avaimen infrastruktuuria. Julkisen avaimen infrastruktuuria käytetään julkisten avainten ja julkisten avainten varmenteiden myöntämiseen, validointiin ja palauttamiseen. (Oppliger, 2016, s. 202.) Ilman varmenteita emme pysty varmistumaan siitä, kenen ylläpitämiä verkkosivuja käytämme. Varmenteiden avulla pystymme varmistumaan, että asioimme oikean verkkosivun kanssa ja tämän vuoksi ne ovat keskeinen osa verkkosivujen salausta.

Esimerkiksi reaalimaailmassa asiakkaat ja kauppiaat luottavat luottokortteihin, jotka suorittavat ostotapahtuman. Kauppias pystyy todentamaan asiakkaan yhdistämällä luottokortissa olevan nimen esimerkiksi ajokortilla todennettuun henkilöllisyyteen. Kauppias luottaa luottokortissa olevaan tietoon ja ostotapahtuman tilaan, jonka hän saa vahvistukseksi luottokortin myöntäjältä onnistuneesta ostotapahtumasta. Vastaavasti asiakas suorittaa ostotapahtuman tietäen, että hän voi hylätä laskun, jos kauppias ei tarjoa tavaroita tai palvelujaan. Luottokortin myöntäjä on tässä tapauksessa tämä luotettu kolmas osapuoli. Sama toimintaperiaate pätee myös verkossa, jossa luotettu kolmas osapuoli on varmenteen myöntäjä ja julkisen avaimen infrastruktuuri tarjoaa alustan tälle. (Kuhn ym., 2001, s. 15.) Yksinkertaistettuna varmenne on siis asiakirja, joka todistaa jonkin totuuden tai jonkin omistamisen. (Oppliger, 2016, s. 201.)

Julkisen avaimen varmenteita on käytännössä kahta eri muotoa ja ne ovat OpenPGP- ja X.509-varmenne. Vaikka SSL/TLS-protokolla tukee kumpaakin näistä, silti X.509 on näistä dominoiva (Oppliger, 2016, s. 206.). Tämän vuoksi tässä tutkimuksessa keskitytään ainoastaan X.509-varmenteisiin.

3.3.1 X.509-varmenteen muoto

X.509 on ITU-T-suosituksen (ITU Telecommunication Standardization Sector) mukainen julkisen avaimen varmenteen muoto, joka ensimmäisen kerran esitettiin vuonna 1988 osana X.500-hakemistosarjan suosituksia (Oppliger, 2016, s. 206). X.509-varmenne voi sisältää paljon eri informaatiota, joista osa on pakollisia ja osa valinnaisia. Varmenne on suojattu varmenteen myöntäjän digitaalisella allekirjoituksella. Jos allekirjoitus pystytään vahvistamaan, selain voi luottaa, ettei varmenteen sisältöä ole jälkikäteen voitu muuttaa. Seuraavaksi esitellään joukko yleisiä lisäosia, jotka X.509-varmenteeseen pystytään tallentamaan, mutta näiden lisäksi varmenne voisi sisältää myös valinnaisen joukon laajennuksia. (Kuhn ym., 2001, s. 22.)

Seuraavassa taulukossa (TAULUKKO 3) esitellään X.509-varmenteen yleisimmät lisäosat ja avataan, mitä tietoa ne pitävät sisällään.

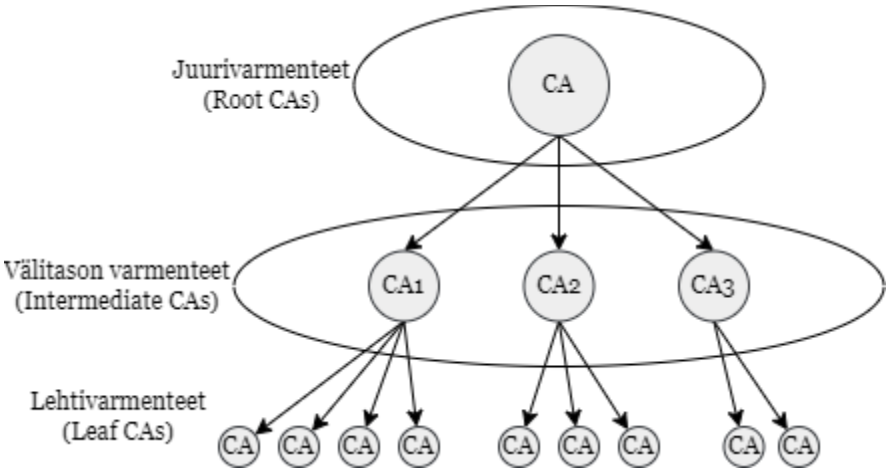
TAULUKKO 3 X.509-varmenteen yleiset lisäosat (Kuhn ym., 2001, ss. 22–23; Oppliger, 2016, ss. 207–208)

Lisäosa	Lisätietoa
Sarjanumero	Sarjanumero on varmenteen myöntäjän luoma kokonaisluku, jolla pystytään yksilöimään jokainen varmenteen myöntäjän allekirjoittama varmenne.
Varmenteen allekirjoitusalgoritmi	Allekirjoitusalgoritmi kertoo, millä algoritmilla (esim. SHA256RSA tai MD5RSA) varmenteen allekirjoitus on tehty.
Varmenteen myöntäjän nimi	Varmenteen myöntäjän erottuva nimi, joka on kuvattu X.500-standardissa.
Varmenteen käyttöiän pituus	Varmenteen käyttöiän pituudella määritellään, milloin varmenne astuu voimaan ja milloin sen käyttö vanhenee.
Aihe	Julkisen avaimen tietokenttä sisältää kohteen julkisen avaimen, valinnaiset parametrit ja algoritmitunnisteen.
Julkinen avain ja parametrit	Julkisen avaimen tietokenttä sisältää kohteen julkisen avaimen, valinnaiset parametrit ja algoritmitunnisteen.
Versionumero	X.509-varmenteesta on kolme eri versiota, joiden syntaksi eroaa toisistaan. Tätä kenttää käytetään kuvastamaan, mitä syntaksia käytetään. Jos versionumero puuttuu, käytetään alkuperäistä, version 1 syntaksia.
Laajennukset	Versiossa 3 varmenne voi sisältää myös laajennuksia. Jokainen laajennus sisältää tunnisteen, kriittisyyslipun ja laajennuksen arvon.

3.3.2 Varmenteen hierarkkinen luottamusmalli

X.509-varmenteet perustuvat hierarkkiselle luottamusmallille, joka muistuttaa rakenteeltaan puuta. Hierarkian ylimmällä tasolla toimii yksi tai useampi juurivarmenne. Juurivarmenne on itse allekirjoitettu varmenne, mikä tarkoittaa, että varmenteen myöntäjä ja sen aihekentät ovat samoja. Itse allekirjoitettu varmenne ei itsessään ole kovin hyödyllinen, koska kuka tahansa pystyy käytännössä allekirjoittamaan oman varmenteensa. Tämä luottamusmalli perustuu rajatusti valikoituihin juurivarmenteisiin, koska johonkin on pystyttävä luottamaan. (Oppliger, 2016, s. 209.).

Alla olevassa kuviossa (KUVIO 6) on hahmoteltu hierarkkista luottamusmallia, jossa kuvion yläreunassa ovat juurivarmenteet, jotka ovat allekirjoittaneet toisen tason varmenteet ja alimpana ovat puun lehtitason varmenteet.



KUVIO 6 Hierarkkinen luottamusmalli (Oppliger, 2016, s. 210)

3.3.3 Varmenteen peruutuslista

Varmenteelle on aina määrätty jokin päättymispäivä, mutta eteen voi tulla tilanne, jossa varmenteesta tulee epäluotettava ennen sen päättymistä (Kuhn ym., 2001, s. 24). Esimerkiksi WWW-palvelimen salainen avain voi vaarantua tai vuotaa ulkopuoliselle, jolloin varmenteesta tulee epäluotettava ja tulee tarpeelliseksi saada peruutettua se (Oppliger, 2016, s. 212). Varmenteen myöntäjä tarvitsee mekanismin, jolla se pystyy viestimään kaikkien varmenteiden tilan. Yksi X.509-varmenteen tilan viestimismekanismeista on varmenteen peruutuslista. (Kuhn ym., 2001, s. 24.)

Varmenteen peruutuslista on musta lista, joka luettelee kaikki peruutetut varmenteet. Peruutuslista ei kuitenkaan sisällä kaikkia vanhentuneita varmenteita, vaan pelkästään ne varmenteet, jotka on peruutettu ennen varmenteen vanhenemista. Koska varmenteen peruutuslistoista on tullut todella suuria, on otettu käyttöön RFC2560-standardissa esitelty verkossa toimiva varmenteen

tilaprotokolla (engl. Online Certificate Status Protocol, OCSP), joka kertoo varmenteen voimassaolosta. (Oppliger, 2016, s. 212.)

Tässä luvussa käsitellyt varmenteet ovat tärkeä osa verkkoliikenteen salaamista, mutta niitä ei hyödynnetä, jos verkkosivuja käytetään salaamattomasti. Seuraavassa luvussa käsitellään turvallisuusmekanismia, jolla saadaan varmistettua se, että verkkosivun käyttäjä käyttää aina salattua yhteyttä.

3.4 HSTS

Zalewskin (2012, s. 248) mielestä yksi verkkoliikenteen salauksen suurimmista ongelmista on se, että selaimet käyttävät oletuksena salaamatonta HTTP-yhteyttä, vaikka verkkosivusto tukisikin salattua HTTPS-yhteyttä. Käyttäjät voivat kirjoittaa selaimen osoiteriviin verkkotunnuksen, kuten esimerkki.fi, jolloin selain oletuksena käyttää URL:ia http://esimerkki.fi sen sijaan, että se käyttäisi salattua muotoa https://esimerkki.fi (Zalewski, 2012, s. 248). Tähän ongelmaan vastauksena IETF julkaisi vuonna 2012 turvallisuusmekanismin nimeltään HSTS (HTTP Strict Transport Security) (Hodges ym., 2012, s. 1; Zalewski, 2012, s. 248).

HSTS tarkoituksena on pakottaa HTTPS-yhteyden käyttö verkkosivustojen käyttäjille (Hodges ym., 2012, s. 9). Käytännössä verkkosivustojen ylläpitäjät voivat määritellä verkkosivustoillensa asetuksen, joka kertoo käyttäjän selaimelle, käyttääkö verkkosivusto pelkästään HTTPS-yhteyttä (Oppliger, 2016, s. 146). Käyttäjän selain tunnistaa HSTS:n käytön ja käyttää pelkästään HTTPS-yhteyttä, jos HSTS on laitettu päälle kyseiselle verkkosivustolle (Zalewski, 2012, s. 248). Näin käyttäjän selain ei käytä salaamatonta HTTP-yhteyttä verkkosivustoissa, joissa HSTS on otettu käyttöön (Hodges ym., 2012, s. 9). Nykyään monissa verkkoselaimissa on ominaisuus, joka varoittaa käyttäjää salaamattomasta HTTP-yhteydestä, mutta silti antaa käyttäjälle mahdollisuuden käyttää verkkosivustoa salaamattomasti. Jos HSTS on asetettu verkkosivulle päälle, se myös poistaa tämän mahdollisuuden käyttäjältä ja näin ollen estää käyttäjää käyttämästä verkkosivustoa salaamattomasti, vaikka käyttäjä näin haluaisikin (U.S Chief Information Officers Council, ei pvm.).

Oppliger (2016, s. 145) mainitsee kirjassaan kaksi eri tapaa, joilla käyttäjän selain voi saada tiedon HSTS:n käytöstä. Ensimmäisessä tavassa käyttäjän selain ottaa yhteyttä verkkosivustoihin, joissa verkkosivusto kertoo käyttäjän selaimelle, käyttääkö verkkosivusto HSTS:ää (Oppliger, 2016, s. 145). Tämän tavan heikkoutena on se, että ensimmäinen yhteys saatetaan tehdä käyttäen salaamatonta HTTP-yhteyttä, jolloin yhteydenotto on tapahtunut ilman salausta (Hodges ym., 2012, s. 36).

Toinen tapa on selaimeen valmiiksi ladattu lista, josta selain voi tarkistaa, käyttääkö verkkosivusto HSTS:ää ennen kuin se ottaa yhteyttä WWW-palvelimeen (Oppliger, 2016, s. 145). Valmiiksi ladattu lista tulee useimpien selaimien mukana, mutta se ei silti kuulu HSTS-standardiin (Hodges ym., 2012, s. 11; MDN Web Docs, 2021; Pflug, 2015; The Chromium Projects, 2021). Valmiiksi

ladatun listan etuna on se, että selain ei joudu erikseen tarkistamaan verkkosivustolta, onko HSTS-asetus päällä, jos verkkosivusto löytyy selaimessa olevalta listalta. (Hodges ym., 2012, s. 31). Käytännössä tämä tarkoittaa, että kun esimerkiksi käyttäjä asentaa selaimen tietokoneeseensa, asennuksen mukana tulee lista verkkosivuista, jotka ovat ottaneet HSTS käyttöönsä ja lisätty kyseiselle listalle (MDN Web Docs, 2021; Pflug, 2015; The Chromium Projects, 2021). Selaimen ottaessa yhteyttä verkkosivustoon tarkistetaan, löytyykö pyydetty verkkosivusto selaimen valmiiksi ladatulta HSTS-listalta (Pflug, 2015; The Chromium Projects, 2021). Jos verkkosivusto löytyy listalta, verkkoliikenne pakotetaan käyttämään HTTPS-yhteyttä (Oppliger, 2016, s. 145).

Verkkosivujen käyttäminen salattuna on tärkeää, mutta tämä salaus pitää toteuttaa luotettavasti. Seuraavassa luvussa tarkastellaan, mitä heikkouksia on löytynyt SSL/TLS-protokollasta.

3.5 SSL/TLS-protokollan heikkoudet

SSL/TLS-protokollan perusominaisuuksia ovat datan alkuperän todentaminen, yhteyden luottamuksellisuuden ja yhteyden eheyden säilyttäminen (McKay & Cooper, 2019, s. 3; Oppliger, 2016, s. 22). SSL/TLS-salaus ei pysty toteuttamaan perusominaisuuksiaan, jos salausta ei ole toteutettu turvallisesti. Tämä voi esimerkiksi tarkoittaa sitä, että käyttäjän selaimen ja WWW-palvelimen välillä lähetettävän datan eheys ja yksityisyys murtuu, ja vihamielinen taho voi saada haltuunsa arkaluontia tietoa, kuten käyttäjätunnuksia tai salasanoja.

SSL/TLS-protokollan historiaa käsittelevässä luvussa (LUKU 3.1) kerrottiin, että osa SSL/TLS-protokolla versioista on todettu vanhentuneiksi ja turvattomiksi, eikä niiden käyttöä pitäisi jatkaa verkkosivustojen suojaamiseen (Barnes ym., 2015; Turner & Polk, 2011). Tässä luvussa perehdytään tarkemmin syihin, miksi osaa SSL/TLS-protokolla versioista pidetään turvattomina sekä muihin salaukseen liittyviin riskeihin. Luvussa kerrotaan SSL/TLS-salauksen haavoittuvuuksista ja syistä, miksi SSL/TLS-salaus pitää olla riittävällä tasolla ja mitä siitä voi seurata, jos salauksen toteutus ei ole turvallinen.

Tässä luvussa käsitellään tarkemmin muutamia tunnettuja SSL/TLS-salaukseen liittyviä haavoittuvuuksia ja hyökkäyksiä, minkä tarkoituksena on kuvata sitä, minkälaisia uhkia heikosti toteutetusta SSL/TLS-salauksesta löytyy. Haavoittuvuuksista ja hyökkäyksistä tarkastellaan sellaisia, joita voidaan vielä nykypäivänä pitää mahdollisina uhkina. Alaluvuissa käsitellään POODLE- ja SLOTH-haavoittuvuuksia sekä FREAK- ja LogJam-hyökkäyksiä. Luvun lopusta löytyy yhteenveto heikkouksiin liittyen. Lukuihin valittiin muutamia toisistaan poikkeavia heikkouksia havainnollistamaan sitä, että että SSL/TLS-salauksesta löytyy useampia eri kohtia, joista voi löytyä heikkouksia.

Tässä luvussa haavoittuvuuksilta suojautumista katsotaan verkkosivustojen ylläpitäjien näkökulmasta. Mainittakoon silti, että verkkosivustojen käyttäjä pystyy suojautumaan osalta tiedossa olevilta haavoittuvuuksista sillä, että päivittää verkkoselaimensa uusimpaan versioon tai tekemällä selaimen turvalli-

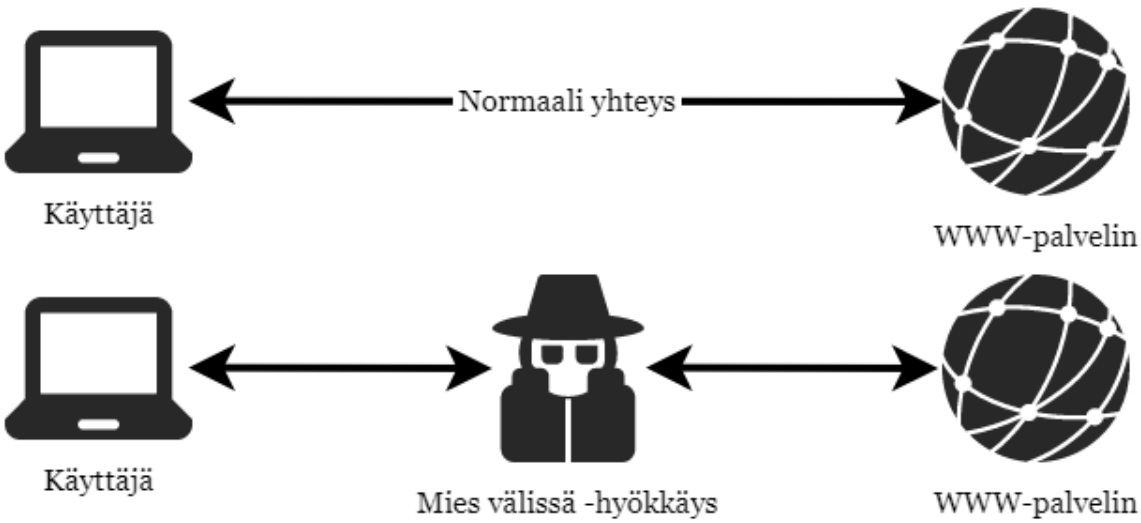
suutta vahvistavia toimia, kuten poistamalla haavoittuvaisia SSL/TLS-versioita käytöstä. Tätä syvällisemmin luvussa ei perehdytä itse verkkosivun käyttäjän suojautumistoimiin, koska tutkimus käsittelee nimenomaan verkkosivujen salauksien turvallisuutta.

Osassa heikkouksien hyödyntämisessä hyökkääjän pitää päästä verkkosivuston ja sen käyttäjän välisen yhteyden väliin, jotta hän pystyy toteuttamaan hyökkäyksensä. Tämänkaltaista hyökkäystä kutsutaan mies välissä-hyökkäykseksi, josta kerrotaan seuraavassa luvussa.

3.5.1 Mies välissä -hyökkäys

Mies välissä -hyökkäys on nimensä mukaisesti hyökkäys, jossa kolmas osapuoli tulee kahden osapuolen välisen yhteyden väliin, esimerkiksi WWW-palvelimen ja WWW-palvelimen käyttäjän väliin (Loshin, 2003, s. 558). Xia & Brustoloni (2005, s. 490) kertovat tutkimuksessaan, että kolmannella osapuolella on usein tavoitteena, ettei yhteydessä olevat osapuolet olisi tietoisia yhteyden välissä olevasta kolmannesta osapuolesta. Tämä tavallisesti saavutetaan niin, että kolmas osapuoli esittää yhteyden toisessa päässä olevaa osapuolta, eli verkkosivulle hyökkääjä esittää käyttäjää ja käyttäjälle hyökkääjä esittää verkkosivua (Xia & Brustoloni, 2005, s. 490).

Alla olevassa kuviossa on kuvattu kaksi tilannetta. Ylempi kuvastaa normaalia tilannetta, jossa käyttäjä ja WWW-palvelin keskustelevat keskenään, ilman miestä välissä. Alempana on mies välissä -hyökkäys, jossa normaalin yhteyden väliin tulee kolmas osapuoli (KUVIO 7.)



KUVIO 7 Mies välissä -hyökkäys

Mies välissä -hyökkäyksessä hyökkääjä yleensä yrittää jollakin tavalla vaikuttaa yhteyteen, esimerkiksi muokkaamalla yhteydessä vaihdettavia viestejä ja varastamalla arkaluonteista tietoa, kuten salasanoja ja käyttäjätunnuksia (Xia & Brustoloni, 2005, s. 490).

Mies välissä -hyökkäyksessä hyökkääjä voi myös vain salakuunnella liikennettä ilman, että yrittää vaikuttaa yhteyteen. Kuunnellakseen yhteydessä meneviä viestejä hyökkääjän ei aina ole välttämätöntä päästä yhteyden väliin. Hyökkääjän on myös mahdollista pelkästään kuunnella yhteyttä ilman, että on kolmantena osapuolena yhteyden välissä, esimerkiksi huonosti suojatuissa langattomissa yhteyksissä ja keskittimissä. (Xia & Brustoloni, 2005, s. 490.)

Otetaan käytännön esimerkkinä tilanne, jossa samassa huoneessa on kolme henkilöä. Kaksi henkilöä keskustelee keskenään ja kolmas henkilö kuuntelee tätä keskustelua. Tässä tilanteessa kolmas osapuoli ei ole mies välissä, mutta kuulee kaiken, mitä nämä kaksi henkilöä keskustelevat. Jos ensimmäinen henkilö kuiskaisi viestinsä kolmannelle osapuolelle ja kolmas osapuoli kuiskaisi tämän viestin eteenpäin keskustelun toiselle osapuolelle, olisi kolmas osapuoli mies välissä. Tällöin hän pystyisi halutessaan muuttamaan henkilöiden kuiskaamia viestejä ja vaikuttamaan keskusteluun, ellei ensimmäisellä ja toisella osapuolella olisi omaa salakieltään, jota kolmas osapuoli ei ymmärtäisi.

Samalla tapaa toimii TCP-yhteys ja SSL/TLS-salaus. Jos osapuolet keskustelevat keskenään ilman salakieltä eli salausta tai heikosti toteutetulla salauksella, kolmas osapuoli pystyy ymmärtämään keskustelua tai vaikuttamaan siihen (Loshin, 2003, s. 558). Jos osapuolten välillä salaus on toteutettu turvallisesti, ei kolmannen osapuolen pitäisi pystyä ymmärtämään keskustelua tai vaikuttamaan siihen (Xia & Brustoloni, 2005, ss. 490–491). Tästä syystä HTTP-yhteys kannattaa salata turvallisesti, mihin perehdytään tarkemmin myöhemmin tulevassa suosituksia käsittelevässä luvussa (LUKU 3.6).

Turvaton salaus on salaus, josta löytyy jonkinlainen heikkous, joka mahdollistaa kolmannelle osapuolelle salauksen purkamisen tai osia viestien purkamisesta. Näitä eri salauksen heikkouksia voi löytyä eri kohdista salaamista, kuten lohkon ketjuttamisesta, avaimien vaihdosta ja tiivistealgoritmeista. Seuraavassa luvussa perehdytään lohkojen ketjuttamiseen liittyvään heikkouteen nimeltä POODLE, joka on yksi löydetyistä haavoittuvuuksista SSL/TLS-salauksessa.

3.5.2 POODLE

Alkuperäinen POODLE-haavoittuvuus (Padding Oracle on Downgraded Legacy Encryption) löytyi vuonna 2014 ja se koskee SSL v3:n käyttämää lohkon ketjuttamista (engl. Cipher-Block Chaining, CBC) (The MITRE Corporation, 2014). Haavoittuvuuden avulla hyökkääjän on mahdollista kuunnella osaa salatusta verkkoliikenteestä ja saada haltuunsa tunnistautumisevästeitä (Nidecki, 2020; Oppliger, 2016, ss. 83–84). Käytännössä siis hyökkääjän on mahdollista saada esimerkiksi sivustolla uhrin käyttämä käyttäjätunnus ja salasana (Nidecki, 2020).

Paras keino suojautua alkuperäiseltä POODLE:lta on estää SSL v3:n käyttö kokonaan (Barnes ym., 2015, s. 3; Möller ym., 2014, s. 2). SSL Labsin (2021) ja Nideckin (2020) keräämien tietojen mukaan vielä noin kolme prosenttia heidän tutkimistaan sivustoista tuki vielä vanhentuneen ja turvattoman SSL v3:n käyttöä.

Alkuperäisen POODLE:n lisäksi SSL/TLS-salauksesta on löytynyt muitakin vastaavia lohkon ketjuttamiseen liittyviä haavoittuvuuksia, kuten Zombie POODLE ja GOLDENDOODLE, jotka löytyivät vuonna 2019 (Kyberturvallisuuskeskus, 2019; Young, 2019a). Näiden TLS versioita 1.0, 1.1 ja 1.2 koskevien haavoittuvuuksien avulla hyökkääjä pystyy purkamaan salattua yhteyttä. (Digicert, 2021; Kyberturvallisuuskeskus, 2019; Young, 2019a). Näiltä tiedossa olevilta lohkon ketjuttamiseen liittyviltä haavoittuvuuksilta voi suojautua, jos poistaa lohkon ketjuttamisen käytöstä tai aktivoi pelkästään TLS v1.3 käyttöön (Digicert, 2021; Young, 2019b). Noin 0,2 prosenttia SSL Labsin (2021) tutkimuksessaan keräämistä sivustoista oli haavoittuvaisia näille uudemmille POODLE-hyökkäykselle.

3.5.3 SLOTH

SSL/TLS-salausten käyttämien tiivistealgoritmien heikkoutena voi olla, että eri viesti voi muodostaa saman tiivisteen, mistä mainittiin aikaisemmin luvussa 3.2.3. Tätä tilannetta kutsutaan ”tiivisteiden yhteentörmäykseksi”, jota myös SLOTH-haavoittuvuus hyödyntää. SLOTH löydettiin vuonna 2015 ja se liittyy MD5-tiivistealgoritmistä löytyvään haavoittuvuuteen. (Red Hat, 2016.)

MD5-tiivistealgoritmin haavoittuvuutena on, että sen muodostama tiiviste on niin heikko, että samojen tiivisteiden luominen eri viesteillä on mahdollista. Tätä ”tiivisteiden yhteentörmäys” -heikkoutta hyväksi käyttämällä hyökkääjä pystyy lähettämään omatekemiään viestejä selaimelle tai sivustolle, jotka luulevat viestin tulleen alkuperäisestä lähteestä, koska hyökkääjä pystyy liittämään viestiinsä luotetun tiivisteen. (Sanchez, 2016.)

Sanchezin (2016) mukaan SLOTH-haavoittuvuuden avulla hyökkääjä pystyy lähettämään selaimelle ja sivustolle väärennettyjä viestejä sekä purkamaan selaimen ja sivuston välistä salattua liikennettä. SLOTH-haavoittuvuutta hyödyntämällä hyökkääjän on mahdollista esimerkiksi kaapata arkaluontoista tietoa, kuten viesteissä siirrettäviä salasanoja ja käyttäjätunnuksia (Sanchez, 2016). SLOTH-haavoittuvuudelta on mahdollista suojautua, kun lopettaa MD5-tiivistealgoritmin käyttämisen ja käyttää turvallisia tiivistealgoritmeja (Red Hat, 2016), joita esitellään myöhemmin SSL/TLS-salauksen suosituksia käsittelevässä luvussa (LUKU 3.6).

3.5.4 FREAK ja LogJam

FREAK-hyökkäys (Factoring RSA Export Keys) löytyi vuonna 2015 ja se hyödyntää 512-bittistä tai heikompaa RSA_EXPORT-salausta (Woodfield, 2015). Selaimen ja sivuston välissä oleva hyökkääjä pystyy hyödyntämään RSA_EXPORT-salauksen haavoittuvuutta yksinkertaisesti selitettynä seuraavalla tavalla.

Mies välissä -hyökkääjä kaappaa selaimen lähettämän *ClientHello*-viestin, jossa on mukana selaimen tuetut salausalgoritmit. Hyökkääjä väärentää tämän *ClientHello*-viestin, jossa pyydetään sivustoa käyttämään turvatonta

RSA_EXPORT-salausta, ja lähettää tämän väärennetyn pyynnön sivustolle. Sivusto vastaa hyökkääjälle turvattomalla export-luokan RSA-avaimella, jonka hyökkääjä välittää selaimelle. Tämän jälkeen hyökkääjä luo export-luokan RSA-avaimen, jonka avulla hän pystyy purkamaan kaiken salatun liikenteen, joka menee selaimen ja sivuston välillä. (Beurdouche ym., 2015, s. 545.)

Tätä FREAK-hyökkäystä muistuttaa samana vuonna löydetty Logjam-hyökkäys. Samalla tapaa kuin FREAK-hyökkäyksessä, Logjam-hyökkäyksessäkin välissä oleva hyökkääjä kaappaa selaimen lähettämän *ClientHello*-viestin ja pyytää sivustolta heikompaan salausta. Yksinkertaisuudessaan, miten Logjam-hyökkäys eroaa FREAK-hyökkäyksestä, on kun FREAK-hyökkäyksessä käytetään RSA_EXPORT-salausta, niin Logjam-hyökkäyksessä hyökkääjä pyytää sivustoa tarjoilemaan 512 bittiä tai alle olevaa export-luokan DHE-salausta (Diffie-Hellman Epheremal). Tutkijat pitivät myös mahdollisena, että esimerkiksi valtiolliset toimijat pystyisivät purkamaan 1024 bittistä DHE_EXPORT-salausta. (Adrian ym., 2015, s. 1.)

FREAK-hyökkäykseltä voidaan suojautua kytkemällä RSA_EXPORT-salauksen pois käytöstä, kun taas Logjam-hyökkäykseltä pystyy suojautumaan poistamalla DHE_EXPORT-salauksen käytöstä. Suositeltavaa on samalla poistaa käytöstä kaikki export-luokan -salaukset, jolloin on mahdollista suojautua molemmilta hyökkäyksiltä sekä mahdollisesti muilta vastaavilta export-luokan -salauksiin liittyviltä haavoittuvuuksilta. (Adrian ym., 2015, s. 6; Paganini, 2015; Woodfield, 2015.)

3.5.5 CRIME

Juliano Rizzo ja Thai Duong julkaisivat vuonna 2012 TLS-salauksesta löytyvän heikkouden nimeltään CRIME (Compression Ratio Info-leak Made Easy), joka käyttää hyväksi pakkauksen sivukanavahyökkäystä. Tämä hyökkäys hyödyntää TLS-pakkauksen heikkoutta paljastaen salatun verkkoliikenteen. (Omar Santos, 2013.). Rizzo ja Duong (2012, p. 16) nostivat esityksessään esille, että CRIME-hyökkäyksessä salauksen purkaminen on todella nopeaa, ja yhden tavun purkaminen vaatii keskimäärin vain kuusi HTTP-kyselyä.

TLS 1.3:aa vanhemmat versiot ovat haavoittuvia CRIME-hyökkäykselle, jossa selain pakkaa salatun datan ilman, että se peittää asianmukaisesti selväkielisen datan pituuden. Tieto selväkielisen datan pituudesta luo mahdollisuuden mies välissä -hyökkääjälle arvata HTTP-kyselyn otsikkotietoja tarkkailemalla arvausten pituuseroja ja näin varastaa arkaluontoista dataa. (NIST, 2012a.)

CRIME-hyökkäystä vastaan voi suojautua estämällä tuen TLS-pakkaukselle. Selain pystyy pakkaamaan datan ainoastaan, jos palvelin tukee sitä ja päinvastoin. Kumpikin voi erikseen kieltäytyä käyttämästä TLS-pakkausta ja estää näin CRIME-hyökkäyksen. (Acunetix, ei pvm.)

3.5.6 Heikkouksien yhteenveto

SSL/TLS-salauksesta löytyy useita erilaisia heikkouksia, joita vastaan on kehitetty edellisessä luvussa mainittujen hyökkäyksien lisäksi useita muitakin erilaisia hyökkäyksiä, kuten BREACH, Barmitzvah, BEAST, DROWN ja Lucky 13, joilla kaikilla yritetään jotenkin vaikuttaa SSL/TLS-salauksen turvallisuuteen. Kaikki mainittujen heikkouksien hyväksikäyttö vaatii hyökkääjän pääsyä kuuntelemaan tai mahdollisesti mies välissä-tilanteeseen muokkaamaan selaimen ja WWW-palvelimen välistä HTTPS-liikennettä.

Mainitut heikkoudet liittyvät SSL/TLS-salaukseen, mutta on hyvä ymmärtää, että HTTP-liikenteessä kaikki on salaamatonta. Täten hyökkääjä voi tehdä mitä tahansa aikaisemmin mainittua, jos hän pystyy tekemään mies välissä -hyökkäyksen, koska hänen ei tarvitse murehtia salauksen purkamisesta tai haavoittuvuuksien hyödyntämisestä, jotta hän pystyisi kuuntelemaan liikennettä ja muokkaamaan sitä sekä mahdollisesti varastamaan käyttäjien käyttäjätunnuksia ja salasanoja. Näistä syistä HTTP-yhteyden salaaminen on aina suositeltavaa.

On hyvä tiedostaa, että oletettavasti ei ole olemassa täydellistä salausta, ja jokainen salaus on todennäköisesti aina murrettavissa laskentatehojen kasvaessa. Jokaisesta salaukseen liittyvästä tekniikasta voi myös yllättäen löytyä jotain uusia haavoittuvuuksia, jotka tekevät niistä turvattomia. Näistä syistä verkkosivujen käyttäjien ja verkkosivustojen ylläpitäjien on syytä pitää ohjelmistot ajan tasalla ja päivittää niitä, kun uusia päivityksiä ohjelmistoista julkaistaan. Tämän lisäksi olisi hyvä asettaa TLS-salaus suositellulle tasolle, jotta TLS-salaus pystyy täyttämään sille asetetun perusominaisuudet: datan alkuperän todentaminen, yhteyden luottamuksellisuuden- ja yhteyden eheyden säilyttäminen.

Kaikilta tällä hetkellä tiedossa olevilta haavoittuvuuksilta ja hyökkäyksiltä pitäisi pystyä suojautumaan, kun WWW-palvelin on määritelty käyttämään turvallisia TLS-salauksen asetuksia. Seuraavassa luvussa kootaan TLS-salauksen suositukset, joita WWW-palvelimien tulisi käyttää, jotta TLS-salaus toteutettaisiin turvallisesti.

3.6 SSL/TLS-salauksen suositukset

SSL/TLS-salaus koostuu useasta eri asetuksesta, jotka voidaan asettaa eri arvoihin. Nämä eri asetukset käydään läpi seuraavissa luvuissa. Ajan myötä osa SSL/TLS-protokollan asetuksista on todettu turvattomiksi; esimerkiksi koko SSL-protokolla on esitetty turvattomaksi, ja sen käyttö on kehotettu korvaamaan uudella TLS-protokolalla (Barnes ym., 2015). Tämä tarkoittaa, että vaikka verkkosivu käyttäisi SSL/TLS-salausta, suojaus voi silti olla heikko tai helposti murrettavissa. Suositukset elävät jatkuvasti, ja käytetyillä salauksilla on painetta aika-ajoin vain kiristyä.

Durumeric ym. (2017, s. 10) tutkivat yrityksille suunnattuja tuotteita ja havaitsivat, että useimmat TLS-salauksen oletusasetukset tarjosivat heikkoa salausta, näiden asetusten muuttaminen oli sekavaa ja usein muutokset piti tehdä vähäisellä tai jopa kokonaan ilman dokumentaatiota. Tämä johtaa siihen, että palvelimien ylläpitäjät eivät voi tyytyä vain oletusasetuksiin, vaan heidän vastuullaan on viime kädessä huolehtia asianmukaisesta salauksesta.

Tässä luvussa käsitellään useaa toimijaa, jotka ovat julkaisseet omat suosituksensa TLS-salauksessa käytettävistä asetuksista. Muodostimme neljän eri toimijan julkaisemista suosituksista yhteisen konsensuksen, joka määrittelee TLS-salauksen vähimmäistason. Päädyimme konsensusmenettelyyn, koska vähimmäisasetuksiin ei ole oikeita tai vääriä arvoja ja halusimme päästä yhteisymmärrykseen TLS-salauksen asetuksista. Valitut neljä eri toimijaa edustivat erityyppisiä organisaatioita, ja vanhin suositus oli vuodelta 2019. Tutkimukseen rajattiin ainoastaan sellaiset toimijat, jotka olivat julkaisseet TLS-salauksen suosituksensa 2018 vuoden jälkeen, jonka aikana TLS 1.3 RFC8446 (E. Rescorla, 2018) julkaistiin.

Ensimmäisenä toimijana on Hollannin Kyberturvallisuuskeskus, jonka julkaisu oli kaikista valituista neljästä tuorein (Dutch National Cyber Security Center, 2020). Toisena on NIST (National Institute of Standards and Technology) (McKay & Cooper, 2019), jonka suosituksiin viitattiin myös muissa lähteissä, kuten NSA ja PCI-DSS (National Security Agency, 2021; Souppaya, 2018, s. 48). Kolmantena on SSL Labs, joka on Yhdysvaltalaisen Qualys Inc -yhtiön ylläpitämä TLS-salaukseen liittyvä verkkosivu ja jonka suositus on vuodelta 2020 (SSL Labs, 2020). Neljäntenä on voittoa tavoittelematon Mozilla-säätiön tekemät TLS-salauksen suositukset (Mozilla, 2020b).

Otimme tämän tutkimuksen suosituksiin mukaan ainoastaan minimivaatimukset TLS-salauksen asetuksista ja näistä muodostettiin yhteinen konsensus, josta saatiin TLS-salauksen suositukset. Todellisuudessa käytössämme on verkkosivuja, joissa osasta saatetaan käsitellä paljon arkaluontoista tietoa. Tällaisten verkkosivujen kohdalla, kuten esimerkiksi verkkopankeissa tai terveystietojärjestelmässä, voisi olla perusteltua käyttää minimivaatimusta vahvempaa salausta. Koska suositukset ovat tapauskohtaisia, tässä tutkimuksessa keskityttiin ainoastaan minimitason alittaviin verkkosivuihin. Minimivaatimukset koskettavat kaikkia, ja jokaisen verkkosivun pitäisi ne täyttää, joten minimivaatimuksia voidaan soveltaa kaikkiin verkkosivuihin riippumatta toimialasta ja verkkosivun kriittisyydestä.

Suosituksien suurimmat erot löytyivät tuetuista salaussarjoista, mutta pääpiirteittäin kerätyt suositukset olivat yhdenmukaiset, ja varsinaisia erimielisyyksiä ei löytynyt. Suurin haaste suositeltujen asetuksien etsimisessä oli tulkita, mikä lasketaan vähimmäisvaatimukseksi. Suosituksia kerättäessä jouduimme tulkitsemaan vaihtelevia sanamuotoja, jotka viittaisivat riittävän tietoturvatason kattamiseen. Konsensus muodostettiin riittämättömän ja riittävän välille. Julkaisijat ottivat myös vaihtelevasti kantaa eri TLS-salauksen asetuksiin. Suosituksissa saatettiin suositella jonkun asetuksen käyttöönottamista tai toisessa tapauksessa voitiin suositella jonkun asetuksen käytön estämistä. Konsensuk-

sen muodostamiseen riitti, jos kaksi julkaisijaa olivat suositelleet asetuksen käyttöä tai sen estämistä. Sellaista tilannetta, jossa kaksi toimijaa olisi suositellut jotain TLS-salauksen asetuksen käyttöönottamista ja kaksi olisi ollut sitä vastaan, emme kohdanneet.

Seuraavaksi tässä luvussa käydään läpi jokainen tutkimukseen mukaan otettu TLS-salauksen asetuskategoria, josta kerättiin suositukset neljältä eri toimijalta. TLS-salauksen asetukset ovat TLS-protokollien versiot, salaussarjat, TLS-pakkaus, 0-RTT, OCSP stapling, varmenteiden käyttöiän pituus, varmenteen voimassaolo, varmenteen avaimen koko ja HSTS. Lopuksi luvussa 3.6.10 vedettiin yhteen kaikki annetut suositukset.

3.6.1 SSL/TLS-protokollien versiot

SSL/TLS-salauksesta on kehitetty seitsemän eri versioita, joista kerrottiin aikaisemmin luvussa 3.1. Nämä seitsemän eri versiota ovat SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1, TLS 1.2 ja TLS 1.3.

SSL 1.0:ta ei koskaan otettu julkiseen käyttöön (Oppliger, 2016, s. 13). IETF on ilmoittanut, ettei SSL 2.0:sta ja SSL 3.0:sta pitäisi käyttää, koska ne ovat vanhentuneita ja turvattomia (Barnes ym., 2015; Turner & Polk, 2011). Useampi toimija on lopettanut TLS 1.0:n sekä TLS 1.1:n tukemisen niiden turvattomuuden takia (Langley & Benjamin, 2021; Microsoft, 2021a; Mozilla, 2020a). Moni taho myös suosittelee, ettei TLS 1.0:aa ja TLS 1.1:stä pitäisi enää käyttää HTTP-liikenteen salaamiseen (Dutch National Cyber Security Center, 2020, s. 15; McKay & Cooper, 2019, s. 8; National Security Agency, 2021, s. 5).

Kaikki neljä konsensuksen muodostukseen valittua julkaisijaa oli yksimielisiä siitä, että nykypäivänä riittävä minimivaatimus SSL/TLS-protokollan versiolle ovat TLS 1.2 ja TLS 1.3. Tämä tuki myös yleistä mielipidettä riittävistä käytettävistä TLS-protokollan versioista (Dutch National Cyber Security Center, 2020, s. 15; Kyberturvallisuuskeskus, 2021, s. 4; McKay & Cooper, 2019, s. 8; Mozilla, 2020b; National Security Agency, 2021, s. 5; SSL Labs, 2020).

3.6.2 Salaussarjat

Salaussarjat käytiin läpi aikaisemmassa luvussa 3.2.3, jossa todettiin niiden koostuvan kolmesta eri algoritmista: avaimen vaihto -algoritmi, salausalgoritmi ja viestin todennus -algoritmi. Tutkimuksessa koostettiin konsensus julkaisijoiden antamien salaussarjojen suositusten perusteella. Käytännössä tässä osuudessa keskityttiin ainoastaan TLS 1.2:ssa ja TLS 1.3:ssa käytössä oleviin salaussarjoihin, koska ainoastaan nämä versiot TLS-protokollasta läpäisivät valittujen julkaisijoiden suositukset.

TLS 1.3 (Rescorla, 2018, s. 133) tukee ainoastaan viittä eri salaussarjaa. TLS 1.3:n osalta konsensuksessa hyväksyttiin kaikki käytössä olevat salaussarjat, mutta tämä jakoi myös mielipiteitä. NIST (McKay & Cooper, 2019, s. 19) ja SSL Labs (2020) hyväksyivät kaikki TLS 1.3 -version salaussarjat, kun taas Mozilla-säätiö (2020b) ja Hollannin Kyberturvallisuuskeskus (2020, p. 25) eivät hyväk-

syneet salaussarjoja, jotka käyttivät hyväksi AES_128_CCM tai AES_128_CCM_8 salausalgoritmeja. Kaksi julkaisijaa ei kuitenkaan rajannut TLS 1.3 -salaussarjoja millään tavoin, minkä takia suositusten konsensukseksi hyväksyttiin kaikki TLS 1.3 -versioon kuuluvat salaussarjat.

TLS 1.2 (Dierks & Rescorla, 2008, ss. 75–76) tuki alun perin jopa 37:ää eri salaussarjaa. Myöhemmin TLS 1.2 on saanut salaussarjoihin päivityksiä, joiden myötä salaussarjojen määrä on kasvanut alkuperäisestä. TLS 1.2:n lisäyksenä on tullut esimerkiksi RFC7905, joka esitteli ChaCha20-Poly1305-salaussarjat, sekä RFC4492, jossa määriteltiin elliptisten käyrien -salaussarjat (Blake-Wilson ym., 2006; De Santis ym., 2017).

Suositusten suurimmat erot tulivat nimenomaan TLS 1.2 -salaussarjojen osalta, mutta julkaisupäivämäärällä on voinut olla merkitystä. Sallivimmat säännöt tulivat NIST:ltä (McKay & Cooper, 2019), jonka julkaisun ajankohta oli samalla kaikista vanhin. Konsensuksen muodostamisessa ei kuitenkaan huomioitu julkaisun ajankohtaa millään tavalla, vaan kaikki suositukset laskettiin olevan tarpeeksi tuoreita antamaan riittävän tarkan kuvan nykyhetken suosituksista. TLS 1.2:n osalta riittävään turvallisuustasoon ylsi lopulta 33 salaussarjaa. Lista tuetuista salaussarjoista löytyy tämän tutkimuksen liitteestä (LIITE 2).

3.6.3 TLS-pakkaus

TLS-yhteyden pakkaamista ei suositella, koska sen käytöllä on verkkoyhteyden salausta heikentävä vaikutus. Hyökkääjä pystyy hyväksikäyttämään pakkausalgoritmia suorittamalla suuren määrän kyselyitä ja tällä tavoin rakentamaan alkuperäisen datan uudestaan (Dutch National Cyber Security Center, 2020, s. 19). TLS-pakkauksen ongelmana on sivukanava-hyökkäys, josta tunnettu hyökkäys CRIME tuli kuuluisaksi (Dutch National Cyber Security Center, 2020, s. 19; McKay & Cooper, 2019, s. 32), mitä käsittelimme luvussa 3.5.5. TLS 1.3 -version standardista jätettiin pakkaus kokonaan pois (E. Rescorla, 2018, s. 9).

Yleinen konsensus oli, että pakkausta ei tulisi käyttää. Julkaisijoista ainoastaan Mozilla-säätiö (2020b) jätti ottamatta TLS-pakkaukseen ollenkaan kantaa, mutta muut kolme suosittelivat sen pois ottamista (Dutch National Cyber Security Center, 2020, s. 19; McKay & Cooper, 2019, s. 32; SSL Labs, 2020).

3.6.4 0-RTT

TLS 1.3 -versiossa esitettiin uusi Early Data Indication -lisäosa, joka toi verkkoselaimelle mahdollisuuden lähettää sovellukseen dataa TLS-kättelyn varhaisessa *ClientHello*-vaiheessa. Tästä datan lähettämisestä käytetään myös nimeä 0-RTT. On huomioitava, että TLS ei kuitenkaan suojaa Early Dataa toistohyökkäystä vastaan, mikä tekee siitä haavoittuvan. (McKay & Cooper, 2019, s. 28.)

Toistohyökkäyksen toimintaperiaate Early Data Indication -lisäosassa kuvattiin jo TLS 1.3 -standardissa, jossa painotettiin, että sen käyttö edellyttää aina suojautumismekanismeja toistohyökkäykseltä. Yksinkertaistettuna toistohyökkäyksellä tarkoitetaan verkkohyökkäystä, jossa hyökkääjä toistaa ensimmäisen

lennon (engl. Zero Round Trip Time Resumption, 0-RTT), minkä Early Data Indication mahdollistaa ilman erillistä suojautumista. (E. Rescorla, 2018)

Hollannin Kyberturvallisuuskeskus ja NIST ovat ottaneet kantaa, että Early Data Indication lisäosaa ei tulisi käyttää (Dutch National Cyber Security Center, 2020, s. 20; McKay & Cooper, 2019, s. 32). SSL Labs (2020) ja Mozilla (2020b) eivät puolestaan ole ottaneet tähän mitään kantaa. Yleiseksi konsensukseksi laskettiin, että Early Data -lisäosan ei tulisi olla käytössä.

3.6.5 OCSP stapling

OCSP-protokolla on luotu X.509-varmenteen oikeellisuuden tarkistamiseen. Verkkoselain pystyy ottamaan suoraan yhteyttä varmenteen toimittajaan, joka antaa OCSP-vastauksen, ja täten verkkoselain pystyy vahvistamaan varmenteen oikeellisuuden. Parannuksena tähän WWW-palvelin pystyy myös toimittamaan OCSP-vastauksen itse, mistä käytetään nimitystä OCSP stapling. Tämän ominaisuuden myötä verkkoselaimen ei tarvitse erikseen ottaa uutta salattua yhteyttä varmenteen toimittajaan, mikä parantaa tietosuojariskiä ja nopeuttaa vastauksen saamista. (Dutch National Cyber Security Center, 2020, s. 20.)

OCSP stapling -lisäosan kohdalla yleinen suositus oli, että sen tulisi olla päällä. Julkaisijoista Hollannin Kyberturvallisuuskeskus (2020, p. 20) mainitsi, että riittävän turvallisuustason kattamiseksi riittäisi, vaikka OCSP stapling olisi pois päältä. Muut julkaisijat olivat tästä huolimatta sen kannalla, että OCSP stapling -lisäosan tulisi olla päällä (McKay & Cooper, 2019, s. 23; Mozilla, 2020b; SSL Labs, 2020).

Vaikka yleinen suositus on pitää OCSP stapling päällä, silti suositut WWW-palvelimet, kuten Apache tai Nginx, eivät oletuksena ole asettaneet tätä päälle (Apache HTTP Server, ei pvm.; Nginx, ei pvm.). Microsoftin IIS-palvelimessa OCSP stapling on tietyillä asetuksilla oletuksena päällä, mutta he perustelevat omaa päätöstään jättää aktivoimatta OCSP stapling kaikissa tapauksissa potentiaalisella vaikutuksella palvelimen suorituskykyyn (Microsoft, 2021b).

3.6.6 Varmenteen käyttöiän pituus

TLS mahdollistaa identiteetin todistamisen X.509-varmenteella, josta kerrottiin aikaisemmin tutkimuksen teoriaosuudessa (LUKU 3.3). Varmenteen voimassaoloajalla tarkoitetaan, kuinka pitkäksi ajaksi varmenne on asetettu voimaan, ennen kuin se vanhenee.

SSL Labs (2020) ja Mozilla (2020b) antoivat varmenteen voimassaoloajan suositukseksi yhden vuoden, kun taas NIST (McKay & Cooper, 2019, s. 11) oli sitä mieltä, että varmenteen voisi allekirjoittaa jopa kolmeksi vuodeksi. Hollannin Kyberturvallisuuskeskus ei erikseen ottanut tähän suositukseen mitään kantaa. Konsensukseksi laskettiin siis yksi vuosi, mutta käytännössä varmen-

teiden liikkeellelaskijat allekirjoittavat varmenteen hieman yli vuodeksi liittyen yhteisiin sopimuksiin.

Varmenteiden voimassaoloajasta on pidetty useita eri äänestyksiä, joissa on haettu yhteisiä sääntöjä sen suhteen. CA/Browser Forum piti vuonna 2017 äänestyksen, jossa varmenteiden liikkeellelaskijat ja kuluttajat äänestivät yhdessä varmenteiden enimmäisajaksi 825 päivää (CA/Browser Forum, 2017). Kahta vuotta myöhemmin tästä seurasi toinen äänestys, jossa ehdotettiin varmenteiden enimmäisajan muuttamista 825 päivästä 398 päivään (CA/Browser Forum, 2019). Kaikki kuluttajat, kuten Apple, Chrome ja Mozilla, olivat sitä mieltä, että varmenteiden käyttöiän pituutta tulisi lyhentää 398 päivään. Varmenteiden allekirjoittajista 60 % oli kuitenkin sitä vastaan, minkä vuoksi äänestys ei mennyt läpi (CA/Browser Forum, 2019.). Äänestystuloksesta huolimatta 3.3.2020 Apple ilmoitti, että 1.9.2020 lähtien he poistavat tuen varmenteista, joiden käyttöiän pituus on pidempi kuin 398 päivää (Apple, 2020). Applen tiedotteen seurauksena myös Mozilla ja Chrome ilmoittivat lyhentävänsä varmenteiden voimassaoloajan 398 päivään (Ben Wilson, 2020; Chromium, ei pvm.).

Edellä mainittujen syiden takia varmenteen käyttöiän pituutta tarkennettiin tasan vuodesta päiväkohtaiseksi. Tämän tutkimuksen suositukseen asetettiin, että varmenteen käyttöiän pituus tulee olla tasan tai alle 398 päivää.

3.6.7 Varmenteen voimassaolo

Verkkosivu lasketaan voimassa olevaksi, kun varmennetta käytetään asianmukaisesti. X.509-varmenteen voimassaoloon vaikuttaa, onko varmenne voimassa tai sisältääkö se verkkosivun verkkotunnuksen. Jokaiseen X.509-varmenteeseen on merkitty alkamis- ja päättymispäivä, joiden sisällä varmenne on voimassa. Varmenteen tuli olla voimassa hetkellä, kun työkalu suoritti verkkosivun haun. Lisäksi X.509-varmenteen tuli myös sisältää tieto verkkosivun verkkotunnuksesta eli yleisestä nimestä (engl. common name). Tällä tavoin varmenteen allekirjoittaja vahvistaa, että verkkotunnus todella kuuluu sivuston haltijalle. Varmenteiden toiminnasta on kirjoitettu teoriaosuuden luvussa 3.3.

3.6.8 Varmenteen avaimen koko

X.509-varmenne pitää sisällään julkisen avaimen, jota käytetään hyväksi salatussa verkkoyhteydessä (Kuhn ym., 2001, s. 22). Julkinen avain voi olla esimerkiksi RSA- tai ECDSA-avain, ja merkittävänä parametrina toimii avaimen pituus (Dutch National Cyber Security Center, 2020, s. 9).

RSA- ja ECDSA-avaimet eivät ole vertailukelpoisia keskenään pelkän avaimen koon perusteella. Kummatkin näistä pohjautuvat eri matemaattisiin algoritmeihin, joista elliptisiin käyriin perustuva ECDSA saa tuotettua pienemmällä avaimen koolla saman salaustason. Salauksen tasoa tuleekin mitata sillä, kuinka paljon työtä salauksen murtaminen vaatii. Salaustason voimakkuutta mitataan biteissä. Esimerkiksi jos tarkoituksena on päästä 112 -bittiseen salaustason voimakkuuteen, RSA:n tapauksessa tämä vaatii 2048-bittisen avai-

men koon, kun taas ECDSA-avaimella sama salaustaso saavutetaan jo pelkällä 224–255 bittiä pitkällä avaimella. Alla esitetään taulukkomuodossa (TAULUKKO 4) eri salausten voimakkuuksien tasot ja niitä vastaavat avaimen koot RSA- ja ECDSA-algoritmeilla. (NIST, 2012b, ss. 51–53.)

TAULUKKO 4 Vertailussa eri algoritmien salauksen tasot (NIST, 2012b, s. 53)

Salauksen taso	Symmetrinen avain	RSA	ECDSA
≤ 80	2TDEA	1024	160–223
112	3TDEA	2048	224–255
128	AES-128	3072	256–383
192	AES-192	7680	384–511
256	AES-256	15360	512+

Kaikki neljä toimijaa olivat yksimielisiä siitä, että RSA-avaimen koon tulisi olla 2048 bittiä ja ECDSA avaimen koko tulisi olla 256–383 bittiä (Dutch National Cyber Security Center, 2020, s. 18; McKay & Cooper, 2019, s. 11; Mozilla, 2020b; SSL Labs, 2020). Toimijat asettivat samalla ECDSA:n minimivaatimuksen korkeammalle, koska sen salauksen voimakkuuden taso oli 128 bittiä, kun taas RSA-avaimelle riitti 112-bittinen voimakkuus.

3.6.9 HSTS

HSTS-suojauksen toimintaperiaatteet käytiin aikaisemmin läpi luvussa 3.4. Hollannin Kyberturvallisuuskeskus ja NIST (McKay & Cooper, 2019) eivät erikseen ottaneet kantaa HSTS:n käyttöön. HSTS ei kuulu itse TLS-protokollaan, vaan se on ainoastaan asiakaspäätteisiin, mikä on saattanut vaikuttaa siihen, miksi Hollannin Kyberturvallisuuskeskus ja NIST (McKay & Cooper, 2019) eivät ottaneet tähän kantaa. SSL Labs (2020) ja Mozilla (2020b) kuitenkin suosittelivat HSTS:n käyttöönottamista, joten suositusten konsensuskin vaatii sen käyttöönottamista.

Suosituista WWW-palvelimista Nginx ja IIS eivät oletusasetuksilla tue HSTS:n käyttöä, vaan se pitää erikseen asettaa päälle (Owen Garrett, 2016; Shi ym., 2020).

3.6.10 Yhteenveto SSL/TLS-salauksen suosituksista

Tässä luvussa vedetään yhteen edellisissä luvuissa läpikäydyt suositukset, jotka muodostavat yhteisen konsensuksemme TLS-salauksen suosituksista. Näissä kerätyissä suosituksissa on otettava huomioon, että suositukset on tehty tämänhetkisten tietojen mukaisesti, ja annetut suositukset vanhenevat ajan kanssa tietokoneiden laskentatehon kasvaessa. Taulukkoon on kerätty yhteenvetona kaikki toimijoiden antamat suositukset, ja viimeisessä sarakkeessa näiden muodostama konsensus. (TAULUKKO 5.)

TAULUKKO 5 Suosituksien yhteenveto ja yhteinen konsensus

Julkaisija	Hollannin Kyberturvallisuuskeskus	NIST	SSL Labs	Mozilla	Suositusten konsensus
Päivitetty	19.01.2021	01.08.2019	15.01.2020	27.07.2020	
SSL/TLS versiot	TLSv1.2 / TLSv1.3	TLSv1.2 / TLSv1.3	TLSv1.2 / TLSv1.3	TLSv1.2 / TLSv1.3	TLSv1.2 / TLSv1.3
Salaussarjat	Liitteenä	Liitteenä	Liitteenä	Liitteenä	
TLS-pakkaus	POIS	POIS	POIS	-	POIS
0-RTT	POIS	POIS	-	-	POIS
OCSP stapling	PÄÄLLÄ/POIS	PÄÄLLÄ	PÄÄLLÄ	PÄÄLLÄ	PÄÄLLÄ
Varmenteen koko (RSA)	112 bittiä	112 bittiä	112 bittiä	112 bittiä	112 bittiä
Varmenteen koko (ECDSA)	128 bittiä	-	-	128 bittiä	128 bittiä
Varmenteen käyttöiän pituus	-	Kolme vuotta	Yksi vuosi	Yksi vuosi	Yksi vuosi (398 päivää)
HSTS	-	-	PÄÄLLÄ (1 vuosi)	PÄÄLLÄ (2 vuotta)	PÄÄLLÄ

Yllä olevien suosituksien hyödyntämisen voisi mieltää yksinkertaisuudessaan auton katsastamiseen. Auton katsastuksessa tarkistetaan jokainen auton osio ja katsotaan, täyttääkö auto asetetut katsastuskriteerit. Jos jostakin auton osasta löytyy vikaa, auto ei mene katsastuksesta läpi ja auto hylätään. Tällöin myös auton tietoihin merkataan, mistä syystä auton katsastus epäonnistui. Samalla tavalla tässäkin tutkimuksessa verkkosivujen TLS-salausta verrataan yllä oleviin suosituksiin ja tarkistetaan, läpäisikö verkkosivu kaikki asetetut kategoriat. Jos verkkosivusto ei läpäise jotakin kategoriaa, se ei täytä asetettuja suosituksia. Tuloksissa nämä kaikki kategoriat esitetään siinä muodossa, kuinka monta verkkosivua täytti asetetun kategorian ja kuinka monta alitti asetetun kategorian. Lopuksi esitetään lukumäärä verkkosivuista, jotka täyttivät kaikki asetetut suositukset tai alittivat suositukset yhdessä tai useammassa kategoriassa.

Suosituksissa ei erikseen otettu kantaa varmenteen yleiseen nimeen, koska se ei varsinaisesti ole asetus, joka laitetaan päälle tai pois, vaan sen pitää olla kunnossa, jotta salaukseen voidaan luottaa. Toimivan varmenteen tulee pitää sisällään verkkosivun verkkotunnus, josta kerrottiin aikaisemmin luvussa 3.3. (Davies, 2010, s. 223.) Tämä varmenteen yleinen nimi otetaan tuloksissa yhdeksi tarkastettavaksi kategoriaksi, jotta tiedetään käytettävän varmenteen olevan kunnossa.

SSL Labs -sivusto julkaisee maailman tunnetuimpien verkkosivujen SSL/TLS-salauksesta tilastoja, joista saadaan käsitys nykyisestä SSL/TLS-salauksen tasosta. SSL Labsin julkaisemat tilastot eivät kuitenkaan ole kaikilta osin vertailukelpoisia tämän tutkimuksen kanssa, koska he eivät ole julkaisseet kaikkia tilastoja riittävän tarkalla tasolla tai keränneet niitä ollenkaan. Esimerkiksi salaussarjoista SSL Labs antaa vain arvosanan, mutta ei suoraan kerro, mitä salaussarjoja verkkosivut ovat käyttäneet. Näin ollen täydellistä vertailua

tämän tutkimuksen suositusten ja SSL Labsin julkaisemien tilastojen välillä ei päästä tekemään.

Alla olevassa taulukossa (TAULUKKO 6) tehtiin vertailu SSL Labsin (2021) tilastojen ja tässä tutkimuksessa kerättyjen TLS-salauksen suositusten osalta niissä kategorioissa, joissa se oli mahdollista tehdä. Vertailusta jouduttiin jättämään pois salaussarjat ja varmenteen käyttöiän pituus. Tutkimuksen ajankohtana tuorein SSL Labsin tilasto oli kerätty 20.09.2021 ja siinä oli mukana yhteensä 123 384 sivustoa (SSL Labs, 2021).

TAULUKKO 6 SSL Labs -tilastot 20.09.2021 (SSL Labs, 2021)

Julkaisija	**Alitti suositukset**
SSL/TLS-versiot	59359 (48,1 %)
TLS-pakkaus	135 (0,1 %)
0-RTT	463 (0,4 %)
OCSP stapling	71420 (57,9 %)
Varmenteen koko	0 (0,0 %)
HSTS	86485 (70,1 %)

3.7 Yhteenveto HTTPS-luvusta

Tämän luvun alussa kerroimme, miksi HTTPS-salausta tulisi käyttää ja mitä vastaan salauksella voidaan suojautua. HTTPS on suunniteltu turvaamaan verkkoliikenne salakuuntelulta ja aktiiviselta hyökkäykseltä. Verkkoliikenteen salaamisen edut voidaan tiivistää kolmeen eri kohtaan, jotka ovat salakuuntelemisen estäminen, verkkoliikenteen eheyden varmistaminen ja verkkosivun identiteetin todentaminen. HTTPS-salaus toteutetaan SSL/TLS-protokollalla, johon pääasiassa keskityttiin tässä luvussa. Luvussa käytiin läpi, kuinka SSL/TLS-protokolla on historiassa kehittynyt, ja avasimme myös sen käytännön toimintaa. Luvussa nostettiin myös esille julkisen avaimen varmenteiden tärkeys osana verkkosivujen salausta. HSTS-luvussa kerroimme turvallisuusmekanismista, jolla käyttäjät saadaan pakotettua käyttämään pelkästään salattua verkkoyhteyttä. Tämän jälkeen esittelimme heikkouksia ja haavoittuvuuksia, joita SSL/TLS-salausprotokollista on löytynyt.

Luvussa 3.6 keräämiämme TLS-salauksen suosituksia pystytään hyödyntämään verkkosivujen salauksia tehdessä tai myöhemmin jatkotutkimuksia varten. Suositukset koottiin neljältä eri toimijalta, joiden suosituksista kerätty konsensus määritteli verkkosivujen salauksen vähimmäistason tässä tutkimuksessa. Toimijoiden suositukset olivat myös melko yhdenmukaiset, joten jo yksittäisiäkin toimijoita seuraamalla pääsisi jo lähelle samaa lopputulosta.

Luvussa 3.6.10 nostimme esille SSL Labsin (2021) julkaiseman tilaston, jossa kävi ilmi, että suurin osa maailman yleisimmin käytetyistä sivustoista ei täyttänyt edes vähimmäistasoa TLS-salauksen suosituksista. Myöhemmin tässä tutkimuksessa tulemme toteuttamaan TLS-salauksen testit Helsingin pörssiyhtiötä

vasten kehittämällämme työkalulla. SSL Labsin (2021) tilastoista voi päätellä, että myös Helsingin pörssiyhtiöiden verkkosivujen salauksessa voisi olla parannettavaa.

Tähän päättyy tutkimuksen teoriaosuus ja tästä siirrytään tutkielman tutkimusosuuteen. Seuraavassa luvussa esitellään tutkimuksessa käytettävä konstruktiivinen tutkimusote ja kuinka tutkimus on kokonaisuudessa toteutettu.

4 TUTKIMUSMENETELMÄN ESITTELY

Tutkimukseen valittu näkökulma ja ratkaistava tutkimusongelma määrittelevät tutkimuksessa käytettävän tutkimusmenetelmän, ja tutkimusmenetelmä määrittelee tutkimuksen prosessin vaiheet (Oyegoke, 2011, s. 574). Tutkielman tutkimusmenetelmäksi valittiin konstruktiivinen tutkimusote ja sitä hyödynnetään siltä osin kuin se soveltuu valittuihin tutkimustavoitteisiin.

Tässä luvussa tutustutaan konstruktiivisen tutkimusotteen teoriaan sekä siihen, miten konstruktiivista tutkimusotetta käytetään ja sovelletaan tässä tutkimuksessa. Luvussa myös käsitellään luotavan konstruktion toteuttamista sekä konstruktion testausta ja validointia. Luvun lopussa kerrotaan tutkimuksen aineiston keräämisestä ja käsittelystä. Seuraavassa alaluvussa kerrotaan meidän kontribuutiomme tähän tutkimukseen liittyen.

4.1 Tutkijoiden kontribuutio

Toteutimme tutkimuksen parityönä, johon liittyen luvussa käsittelemme, miten tutkimuksessa olevat työkohdat jakautuivat meidän kesken. Toteutimme tutkimuksemme aikavälillä joulukuu 2020 - toukokuu 2022.

Teimme yhdessä Tiivistelmän ja Johdannon (LUKU 1). Johdannossa kirjoitimme ja kommentoimme jokaista alalukua, jotta johdantoon saatiin kaikki oleellinen tieto tutkimusta varten. Näin myös varmistimme, että meillä molemmilla oli yhdenmukainen käsitys siitä, mitä tutkimuksessa tehdään ja miksi.

Teimme yhdessä kirjallisuuskatsauksen, ja sen yhteydessä kirjoitimme myös tutkimuksen teoriaosuuden, jonka työtehtävät jakautuivat seuraavasti. Soivi kirjoitti luvun kaksi, jossa käsiteltiin muun muassa TCP/IP-protokollaa, verkkotunnuksia ja HTTP-protokollaa. Kiuru puolestaan kirjoitti pääasiallisesti luvun kolme, joka syventyi SSL/TLS-salaukseen ja SSL/TLS-salauksen suosituksissa käytettäviin parametreihin. Soivi kirjoitti lukuun kolme SSL/TLS-protokollan historia -luvun sekä suurimmaksi osaksi SSL/TLS-protokollan heikkoudet -luvun.

Tutkimusmenetelmä-luvun (LUKU 4) tekeminen jakautui niin, että Soivi kirjoitti konstruktiivisesta tutkimusotteesta sekä miten valittua tutkimusmenetelmää sovellettiin tässä tutkimuksessa. Kiuru kirjoitti konstruktion toteutukseen, testaukseen ja tutkimusaineiston keruuseen liittyvät luvut. Tutkijoiden kontribuutio -luvun kirjoitimme yhdessä.

Suunnittelimme ja määrittelimme konstruktion yhdessä, mutta Kiuru toteutti työkalun ohjelmoinnin (LIITE 4; LIITE 5; LIITE 6) sekä yksittäisten vaiheiden testaamisen. Työkalun toteuttamisen jälkeen katselmoimme työkalun ohjelmakoodin sekä testauksessa muodostuneen aineiston, jotta saimme varmistettua, että työkalu täytti ne vaatimukset, jotka määrittelyssä sovimme.

Kiuru suoritti syyskuussa 2021 työkalun ajamisen. Ajossa kerättiin kohteet ja tehtiin SSL/TLS-salauksen testit Helsingin pörssiyhtiöiden verkkosivuja vasten. Tulokset tallennettiin yhteiselle jaetulle tietokantapalvelimellemme. Soivi toteutti tuloksien keräämistä varten MongoDB-kyselyt (LIITE 7 – LIITE 18), joilla tutkimustulokset saatiin kerättyä tietokantapalvelimelta. Soivi dokumentoi kyselyistä saadut tulokset viidenteen lukuun. Kiuru tarkasti, että kyselyt hakivat haluttua tietoa sekä vahvisti, että kyselyistä saadut tulokset olivat oikeita.

Teimme pohdintaosuuden yhdessä. Luvussa kirjoitimme ja otimme kantaa tutkimuskysymyksiin, tutkimuksen reliabiliteettiin ja validiteettiin sekä pohdiskelimme tutkimuksen kontribuutiota. Tämä tehtiin yhdessä siksi, että saimme molempien perspektiivin tutkimukseen liittyen ja pohdittua laajemmin koko tutkimusta. Tutkimuksen lopuksi kirjoitimme tutkimuksen Johtopäätökset, jossa Kiuru teki tutkimuksen yhteenvedon, sekä yhteistyössä pohdimme ja dokumentoimme mahdolliset jatkotutkimusaiheet.

Teimme yhteistyötä kaikissa tutkimusraportin kohdissa. Luimme aina toisen kirjoittaman tekstin, sekä luvuista käytiin aina keskustelu ennen ja jälkeen sen kirjoittamisen. Kommentoimme luettua tekstiä ja tarvittaessa sen pohjalta teimme korjauksia jo kirjoitettuun tekstiin. Tällä tavoittelimme sitä, että tutkimukseen saatiin kaikki oleellinen tieto ja että tutkimusmateriaali on meidän molempien hyväksymää. Näin myös pysyimme molemmat koko ajan tietoisina, miten tutkimuksemme etenee ja varmistimme, että tutkimuksemme etenee kohti yhteisesti asetettua tavoitettamme.

Seuraavaksi perehdytään tutkimuksessa käytettävään tutkimusmenetelmään.

4.2 Konstruktiivinen tutkimusote

Konstruktiivisessa tutkimusotteessa keskeisessä asemassa on reaalimaailman ongelma ja sen ratkaisemiseksi kehitetty konstruktio. Konstruktiivisessa tutkimusotteessa ongelmaksi voidaan valita melkein mikä tahansa tosielämän asia, joka koetaan tarpeellisena ratkaista. Tämä reaalimaailman ongelma voi olla esimerkiksi sairaus, johon ei ole vielä löydetty hoitoa, yrityksen prosessi, jota voisi tehostaa tietoteknisellä ratkaisulla, tai jokin reaalimaailman toiminnon

mittaaminen jollakin uudella mittaustavalla. Näihin reaalimaailman ongelmiin haetaan ratkaisua kehitettävällä konstruktiolla. (Lukka, 2001.)

Konstruktiolla tarkoitetaan abstraktia asiaa, joka voi olla lähes mitä vain, mikä ratkaisee reaalimaailman ongelman ja millä tuotetaan kontribuutiota tieteenalalle (Uusitalo & Kohtamäki, 2011, ss. 281–282). Konstruktioille ominaista on, että ne eivät ole luonnon muodostamia, vaan ne ovat jotain tarkoitusta varten ihmisen keksimiä ja kehittämiä (Lukka, 2001). Konstruktiona voidaankin käytännössä pitää kaikkea ihmisen luomaa, kuten diagrammeja, esineitä, organisaatiorakenteita ja tietojärjestelmiä (Kasanen ym., 1993, s. 245). Konstruktio on aina uusi asia, joka kehitetään tutkimuksessa, eikä aiemmin kehitettyjä konstruktiota sovellettuna uuteen ongelmaan pidetä konstruktiivisen tutkimusotteen toteutuksena (Lukka, 2001).

On hyvä huomioida se, että konstruktiivisessa tutkimusotteessa hyväksytään se, ettei konstruktio ratkaise asetettua ongelmaa. Epäonnistunut konstruktio ei automaattisesti tarkoita, että itse tutkimus olisi epäonnistunut. Jokaisella epäonnistuneella konstruktiolla luodaan uutta tietoa ongelmasta ja se lisää teoreettista ymmärrystä, miten ongelma ratkaistaan. Tiedon lisääntyminen ongelmaan liittyen ja ymmärrys sopivista ratkaisuista edistävät ongelman ratkaisua ja sopivan konstruktion toteuttamista. Epäonnistunutkin konstruktio voi siis tulevaisuudessa antaa ratkaisun avaimet onnistuneelle konstruktiolle, joka ratkaisee asetetun ongelman. Jokainen onnistunut ja epäonnistunut konstruktio antaa tietopohjaa luomaan uusia konstruktioita, jotka kehittävät todellisuuttamme, eli maailmaa kuten me sen näemme ja ymmärrämme. (Lukka, 2001.)

Konstruktiivisen tutkimusotteen on tarkoitus luoda yhteys käytännön ja teorian välille sekä tuoda hyötyä molempiin maailmoihin. Konstruktion luomisessa hyödynnetään jo olemassa olevaa teoriapohjaa, ja konstruktio tulee tuottamaan reaalimaailmaan ratkaisun, jonka tarkoitus on tuoda myös oma kontribuutio teoriamaailmaan. (Uusitalo & Kohtamäki, 2011, s. 284.)

Konstruktiivinen tutkimusote kuuluu yhdeksi tavoiksi toteuttaa tapaustutkimus ja se on myös rinnastettavissa etnografiseen tutkimukseen, grounded theory -tutkimukseen, teoriaa havainnollistavaan case-tutkimukseen, teoriaa testaavaan case-tutkimukseen ja toimintatutkimukseen (Lukka, 1999, s. 144). Konstruktiivisessa tutkimusotteessa korostetaan työkalun kehittämistä ja uuden konstruktion toteutusta, kun taas toimintatutkimuksessa tutkija osallistuu enemmän itse organisaation toiminnan kehittämiseen (Uusitalo & Kohtamäki, 2011, s. 283). Oyegoken (2011, s. 579) mukaan tutkimus ei noudata konstruktiivista tutkimusotetta, jos tutkimuksen tarkoituksena on enemmänkin tarkkailla ja analysoida. Konstruktiivisessa tutkimusotteessa on tarkoituksena löytää ongelma ja vaikuttaa siihen (Oyegoke, 2011, s. 579).

Konstruktiivinen tutkimusote on käytännönläheinen ja se keskittyy erityisesti tutkimukseen, joka tähtää työkalun kehittämiseen ja käytännön lisäarvon tuottamiseen (Uusitalo & Kohtamäki, 2011, s. 283). Konstruktiiviseen tutkimusotteeseen kuuluu myös se, että konstruktion toimivuutta testataan käytännössä ratkaistavaan ongelmaan, jotta tiedetään, ratkaiseeko konstruktio käytännössä

sen ongelman, jonka sen oli tarkoitettu ratkaisevan (Uusitalo & Kohtamäki, 2011, s. 288).

Konstruktiivisessa tutkimusotteessa ei rajoiteta tutkimuksen menetelmiä tai tekniikoita, joilla tutkimus toteutetaan, vaan siinä voidaan soveltaa sellaisia, jotka ovat toimivia (Oyegoke, 2011, ss. 591–592). Syy, miksi menetelmiä ja tekniikoita ei tarkasti määritellä on se, että vaikka konstruktiivinen tutkimusote kehitettiin liiketaloustieteeseen (Lukka, 2001), voidaan kyseistä tutkimusmenetelmää myös soveltaa useilla eri tieteenaloilla, kuten tietojärjestelmätieteissä (Oyegoke, 2011, s. 579), lääketieteessä (Kasanen ym., 1993, s. 245) ja kasvatustieteissä (Vaso, 1998). Tästä syystä tarkkoja raameja konstruktiiviselle tutkimusotteelle on lähes mahdotonta asettaa (Oyegoke, 2011, ss. 591–592). Konstruktiivinen tutkimusote antaakin suhteellisen vapaat kädet ongelman asettamisessa, ratkaisun kehittämisessä ja ratkaisun validoinnissa, eikä tarkasti määrittele, miten nämä kohdat pitäisi toteuttaa (Lukka 2001). Konstruktiivisessa tutkimusotteessa tärkeintä on käytännön ja teorian kontribuutio (Lukka 1999, ss. 141-142).

Konstruktiivinen tutkimusote valittiin tähän tutkimukseen, koska se sopi parhaiten tämän kaltaiseen tutkimukseen, jossa tutkijoilla on aktiivinen rooli ongelman määrittelyssä, konstruktion luomisessa sekä ongelman ratkaisemisessa. Valittu menetelmä soveltui erityisesti tutkimukseen, joka tähtää työkalun kehittämiseen ja käytännön lisäarvon tuottamiseen. Tästä tutkimuksesta siis löytyy lähes kaikki konstruktiivisen tutkimuksen piirteet, minkä takia tutkimuksessa noudatetaan konstruktiivista tutkimusotetta soveltuvin osin.

Uusitalo ja Kohtamäki (2011, s. 282) pitää konstruktiivinen tutkimusotetta tuoreena menetelmänä ja heidän mielestään se vielä vaatii kehittämistä. Tässä tutkimuksessa pyrimme testaamaan ja soveltamaan konstruktiivista tutkimusotetta, joten tutkimuksemme avulla potentiaalisesti luodaan lisää ymmärrystä tutkimusmenetelmästä ja sen käyttömahdollisuuksista muita tutkimuksia varten. Konstruktiivista tutkimukseen kuuluu yleensä seitsemän eri tutkimusprosessin vaihetta (Lukka, 2000, ss. 116–120), joihin perehdytään seuraavassa luvussa. Lisäksi perehdytään siihen, miten konstruktiivista tutkimusotetta sovelletaan tässä tutkimuksessa.

4.3 Konstruktiivinen tutkimusote prosessina

Tässä luvussa kerrotaan, miten tutkimuksessa on toteutettu konstruktiivisen tutkimusotteen prosessin vaiheet ja miten konstruktiivista tutkimusotetta on sovellettu tässä tutkimuksessa. Lukka (2000, ss. 116–120) määrittelee konstruktiiviseen tutkimusotteen seitsemän eri vaihetta, jotka ovat:

1. Etsi käytännössä relevantti ongelma, jossa on mahdollisuus myös teoreettiseen kontribuutioon
2. Selvitä mahdollisuudet pitkän aikavälin tutkimusyhteistyöhön kohdeorganisaation kanssa
3. Hanki syvällinen tutkimusaiheen tuntemus sekä käytännöllisesti että teoreettisesti

4. Innovoi ja kehitä ongelman ratkaiseva konstruktio, jolla voisi olla myös teoreettista kontribuutiota
5. Toteuta ratkaisu ja testaa sen toimivuus
6. Pohdi ratkaisun soveltamisalaa
7. Tunnista ja analysoi teoreettinen kontribuutio

Alaluvuissa kerrotaan tarkemmin nämä seitsemän eri prosessinvaihetta, miten ne on määritelty ja toteutettu tässä tutkimuksessa, sekä mitkä luvut käsittelevät näitä eri prosessin vaiheita. Tämä luku on toteutettu, jotta pystyttäisiin validoimaan tutkimuksessa tehdyt valinnat ja sen toteutuksen sekä halutessaan toistamaan tutkimuksesta saadut tulokset. Seuraavassa luvussa kerrotaan, miten tämän tutkimuksen toteuttamiseen päädyttiin.

4.3.1 Etsi käytännössä relevantti ongelma, jossa on mahdollisuus myös teoreettiseen kontribuutioon

Alkuperäisenä tutkimuskohteenamme oli tutkia suomalaisten yhtiöiden verkkosivujen TLS-salauksen tasoa, koska sen tasosta ei ole aikaisempaa tietoa, sillä kukaan ei ole sitä tiettävästi tutkinut. TLS-salauksen tutkiminen nähtiin tarpeellisena, koska se luo perustan turvalliselle Internetin käytölle. SSL Labsin (2021) tilastojen mukaan 52,8 prosenttia verkkosivuista käyttää riittämätöntä salausta, minkä takia haluttiin tarkemmin selvittää, mikä on suomalaisten yhtiöiden tilanne.

Tutkimusta suunniteltaessa huomasimme muutamia eri haasteita, jotka meidän pitäisi ratkaista, jotta tutkimus olisi realistista toteuttaa. Suomalaisia yhtiöitä on rekisteröity yli 600 tuhatta (Patentti- ja rekisterihallitus, 2022), mutta tutkimukseen sopivaa julkista listaa yhtiöistä ja niiden verkkosivuista ei löytynyt kovinkaan helposti. Halusimme saada laajan määrän verkkosivuja, jotta ne kattaisivat mahdollisimman laajalti erilaisia suomalaisia yhtiöitä. Siltikään tutkittavien verkkosivujen määrä ei saanut kasvaa liian suureksi, koska se olisi voinut asettaa haasteen aineiston keräämisessä, ja tutkimusaineisto haluttiin kerätä kohtuullisen ajan puitteissa.

Valitsimme tutkittaviksi kohteiksi Helsingin pörssin päälistan yritykset, koska niistä saimme tutkimuksen kannalta parhaimman läpileikkauksen Suomessa toimivista yhtiöistä, sillä ne edustavat eri toimialojen ja kokoluokan yrityksiä (LIITE 1). Näistä yhtiöistä pystyimme keräämään tarvittavat tiedot, joita tutkimuksessa tulisimme käyttämään. Tutkittavien yhtiöiden lukumäärä (133 yhtiötä) koettiin sopivaksi, koska isommassa yritysten määrässä olisi voinut tulla aineiston keräämisen kanssa vaikeuksia; aineiston keräämiseen olisi saattanut esimerkiksi kulua liikaa aikaa, mikä olisi vaikuttanut työmme edistymiseen. Pörssiyhtiöiltä voisi myös olettaa löytyvän riittävästi resursseja ylläpitämään verkkosivujensa turvallisuutta. Näiden perusteluiden ja tehdyn pohjatyön perusteella otimme Helsingin pörssiyhtiöt tutkimuksemme kohteiksi.

Tutkimusta ja sen tavoitteita määriteltäessä havaitsimme muutaman ongelman, jotka meidän pitäisi ratkaista, jotta pystyisimme toteuttamaan tutki-

muksemme. Ensimmäisenä ongelmana oli kerätä tarvittavat verkkosivut, joiden TLS-salauksesta voisi kerätä dataa. Toisena ongelmana oli kerätä löydettyjen verkkosivujen TLS-salauksesta tietoa. Näiden ongelmien ratkaisemiseen emme löytäneet valmista tapaa, minkä takia tutkimuksen onnistumisen kannalta konstruktion toteuttaminen on välttämätöntä. Havaitsemiemme ongelmien takia ja konstruktion puutteen takia näimme oleellisena toteuttaa työkalun, jolla pystyttäisiin keräämään tarvittavat verkkosivut sekä niiden TLS-salauksen asetukset.

Tämä työkalu tulee itsessään ratkaisemaan havaitsemamme reaalimaailman ongelman ja se myös julkaistaan muiden tutkijoiden käyttöön. Työkalun toteuttamisen myötä tutkimuksesta saadaan myös teoreettinen kontribuutio, kun selvitämme, mikä on Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso sekä ymmärrämme paremmin työkalun kehityskohdat ja kyvykkyydet, joita tulevaisuuden tutkimuksissa pystytään hyödyntämään.

Tutkielman tutkimusmenetelmäksi valittiin konstruktiivinen tutkimusote, koska edellä mainittujen ongelmien ratkaisemiseksi päädyimme hyödyntämään sellaista tutkimusmenetelmää, joka toimisi parhaiten tutkimukseen, jossa tuotetaan teoreettista kontribuutiota sekä itse konstruktio.

4.3.2 Selvitä mahdollisuudet pitkän aikavälin tutkimusyhteistyöhön kohdeorganisaation kanssa

Konstruktiiviseen tutkimusotteessa konstruktio kuuluisi toteuttaa tutkimusyhteistyössä kohdeorganisaation kanssa. Tähän Lukka (2000, s. 117) mainitsee syyksi sen, että ilman kohdeorganisaatiota tutkijan konstruktion toteuttaminen jää helposti toteuttamatta. Emme kuitenkaan nähneet tätä ongelmana, koska tahtotilanamme oli toteuttaa tutkimus ja siinä syntyvä konstruktio.

Sovellamme tutkimuksessa konstruktiivista tutkimusotetta niin, ettemme valinneet tutkimukseen tutkimusyhteistyöorganisaatiota, jonka tarpeisiin konstruktio tultaisiin suunnittelemaan. Tutkimuksen ja konstruktion toteutuksessa on hyödynnetty yhteistyössä tahoja Jyväskylän Yliopistolta sekä Suomen Kyberturvallisuuskeskukselta, mutta konstruktiota ei toteuteta kummankaan toimeksiantona. Tutkimuksen kohdeyleisönä on tiedeyhteisö sekä konstruktion toteutuksen kohteena ovat tahot, kuten tutkijat ja yritykset, jotka tarvitsevat tämänkaltaista työkalua tarpeisiinsa.

Miksi emme ottaneet tutkimukseen tiettyä kohdeorganisaatiota, jolle konstruktio toteutetaan, johtuu siitä, että havaitsimme ongelman, jonka haluamme ratkaista. Koimme kohdeorganisaation olevan enemmän riski kuin mahdollisuus. Lukka (2000, ss. 116–117) mainitseekin useista eri ongelmista, joita voi syntyä kohdeorganisaation kanssa: Kohdeorganisaatio saattaa hidastaa tai haitata tutkimuksen tekemistä muun muassa erilaisilla sopimusehdoilla, olemalla sitoutumaton tai jättämällä tutkimuksen kesken (Lukka, 2000, ss. 116–117). Kohdeorganisaatio voi myös ohjata tutkimusta suuntaan, joka sopisi paremmin kohdeorganisaatiolle, mutta ei olisi tutkimuksen kannalta mieluisa. Pahimmassa tapauksessa kohdeorganisaatio saattaisi yrittää kieltää konstruktion ja tutkimuksen julkaisemisen (Lukka, 2000, s. 117).

Julkaisemme valmiin konstruktion avoimeksi kaikkien käyttöön, jolloin halukkaat tahot pystyvät hyödyntämään ja muokkaamaan konstruktiotaan omiin tarpeisiinsa ja täten luoda uusia konstruktioita, jotka ratkaisevat uusia ongelmia ja luovat uutta todellisuutta, mikä onkin yksi tärkeimmistä piirteistä konstruktiivisissa tutkimuksissa. Konstruktion julkaiseminen mahdollistaa myös sen, että tutkimuksen toistaminen ja tulosten vahvistaminen on mahdollista, mikä on tärkeä ominaisuus tieteellisissä tutkimuksissa.

4.3.3 Hanki syvällinen tutkimusaiheen tuntemus sekä käytännöllisesti että teoreettisesti

Tutkimuksen johdannossa käsiteltiin aiempia tutkimuksia, jotta ymmärrettiin syvällisemmin, mitä aiheesta jo tiedetään. Aiempien tutkimusten etsimiseen hyödynnettiin eri tieteellisten artikkeleiden julkaisijoita, kuten ACM (Association for Computing Machinery) sekä IEEE (Institute of Electrical and Electronics Engineers).

Teoriakatsauksessa kerättiin teoreettista tuntemusta tutkimusaiheesta ja perehdyttiin aihealueeseen liittyvään kirjallisuuteen, jotta saatiin yleisnäkemystä aihealueesta. Aikaisemmassa tutkijoiden kontribuutio tutkimukseen -luvussa (LUKU 4.1) käsiteltiin, miten tutkimuksen eri työkohdat jakautuivat meidän kesken. Luvun yhtenä tarkoituksena oli osoittaa, että pohjatyö on tehty mahdollisimman laadukkaasti ja läpinäkyvästi, ja että kirjoitettu teksti on meidän molempien validoimaa. Aikaisempien tutkimuksien ja teoriakatsauksen avulla pystytään pohjaamaan tutkimus jo olemassa olevaan tietoon aiheesta sekä havaitsemaan tutkimuksesta saatava teoreettinen kontribuutio.

Konstruktion kehitysvaiheessa perehdytään käytännössä itse ongelmaan, jotta voidaan innovoida toimiva konstruktio. Konstruktion kehitystä varten testasimme useita erilaisia ohjelmistoja, joita voisimme mahdollisesti hyödyntää omassa konstruktiossamme. Näistä oleellisimpia ohjelmistoja käsitellään konstruktion kehittämisessä (LUKU 4.4), jossa myös perustelemme tekemämme valinnat. Perehdyimme myös useisiin eri julkisiin lähteisiin selvittäessämme, mistä julkisista lähteistä löytyy konstruktiota ja tutkimusta varten tarvitsemamme tieto. Näitäkin käsitellään tarkemmin myöhemmässä konstruktion kehitysvaiheessa.

4.3.4 Innovoi ja kehitä ongelman ratkaiseva konstruktio, jolla voisi olla myös teoreettista kontribuutiota

Konstruktion toteutus ja testaus -luvussa (LUKU 4.4) toteutetaan konstruktio, jonka tarkoituksen on ratkaista asetettu ongelma; työkalu, jolla voidaan kerätä laaja määrä verkkosivuja sekä tietoa niiden TLS-salauksesta. Työkalulla myös kerätään aineisto tätä tutkimusta varten. Pohjatyön perusteella emme löytäneet yhtäkään sellaista työkalua, jolla tämänkaltaisen tutkimuksen olisi voinut toteuttaa, minkä takia tulemme innovoimaan ja toteuttamaan konstruktion

tutkimuksessa. Tämä puute koettiin koskevan myös tiedeyhteisöä, minkä takia kehitettävälle työkalulle koettiin tarvetta.

Tutkimuksessa tuotettava konstruktio tuottaa jo itsessään teoreettista kontribuutiota tieteenalalle, vaikka se ei ratkaisisikaan ongelmaa, koska emme saaneet selville, että vastaavaa ongelmaa olisi aikaisemmin ratkaistu tai edes yritetty ratkaista. Konstruktion innovoiminen ja kehittäminen luo ymmärrystä asetetusta ongelmasta sekä tavoista, miten ongelma voidaan ratkaista tai miten se voitaisiin tulevaisuudessa ratkaista paremmin.

Konstruktion lisäksi tiedon lisääntyminen Helsingin pörssiyhtiöiden TLS-salauksen tasosta tulee tuottamaan teoreettista kontribuutiota tieteenalalla, koska sitä ei ole koskaan aikaisemmin tutkittu. Tarkoituksenamme on myös julkaista toteutettu konstruktio, jolloin se potentiaalisesti auttaa tutkijoita tulevaisuudessa keräämään dataa vastaavan kaltaisiin tutkimuksiin.

4.3.5 Toteuta ratkaisu ja testaa sen toimivuus

Konstruktio toteutetaan konstruktion toteutus ja testaus -luvussa (LUKU 4.4), jossa kerrotaan tarkemmalla tasolla konstruktion toiminnasta, toteuttamisesta ja testaamisesta. Kyseisessä luvussa valitaan konstruktion jokainen sen osa perustellusti, ja konstruktion keräämää dataa validoidaan satunnaisotannalla sen luomisen eri vaiheissa. Konstruktio toteutetaan käyttäen ketterää kehitystä: Konstruktio jaettaan pienempiin osiin, eli eri vaiheisiin, jotka toteutetaan ja testataan itsenäisesti. Jos jossakin vaiheessa löytyy ongelma, se korjataan ja vaihe ajetaan uudelleen. Tätä toistetaan niin kauan, kunnes vaihe menee yksinään onnistuneesti läpi. Kun kaikki vaiheet on yksinään testattu ja todettu toimiviksi, siirrytään ohjelman kokonaisvaltaiseen testaamiseen.

Kokonaisvaltainen konstruktion testaus toteutetaan konstruktion valmistumisen jälkeen. Kokonaisvaltaisessa testauksessa ajetaan konstruktion kaikki vaiheet läpi ja näistä syntyy tutkimuksen tulokset, joihin kootaan konstruktion keräämä aineisto analysointia varten. Konstruktion toiminnasta ja sen käyttämisestä luodaan myös selkeä ohjeistus, jotta ulkopuolisetkin tahot pystyvät halutessaan hyödyntämään konstruktiotamme. Ohjeistus on liitettynä konstruktion ohjelmakoodin rinnalle ja löytyy GitHubista (https://github.com/kiuru/pyTLScanner/blob/main/README.md).

4.3.6 Pohdi ratkaisun soveltamisalaa

Konstruktiivisen tutkimusotteen kuudennen vaiheen tarkoituksena on pohtia toteutetun konstruktion soveltamisalaa (Lukka, 2000, s. 119). Konstruktion toteuttamisen lisäksi tutkimuksen päätavoitteena on tutkia Helsingin Pörssi yhtiöiden verkkosivujen TLS-salauksen tasoa, minkä takia tutkimuksen käytännön hyötyä analysoidaan laajemmin, eikä pelkästään konstruktion näkökulmasta. Analysoimme Pohdinnassa (LUKU 6), mitä käytännön hyötyjä tutkimuksestamme on, esimerkiksi mihin eri käyttötarkoituksiin työkalua voisi käyttää, mi-

ten TLS-salauksen tasoa ja siihen liittyvää tietoa voidaan soveltaa, sekä mitä rajoitteita työkalusta löytyy.

4.3.7 Tunnista ja analysoi teoreettinen kontribuutio

Konstruktiiviseen tutkimusotteen viimeisessä vaiheessa otamme etäisyyttä tutkimukseemme ja pohdimme tutkimuksen teoreettista kontribuutiota reflektoiden sitä aiempaan tutkimustietoon (Lukka, 2000, s. 119). Konstruktiivisesta tutkimusotteesta mainittiin yllä, että sillä yritetään löytää uusia tapoja ratkaista reaalimaailman ongelma ja tuottaa teoreettista kontribuutiota tieteenalalle (Lukka, 1999, ss. 141–142).

Tutkimusprosessin seitsemättä vaihetta voisi pitää konstruktiivisen tutkimusotteen kannalta tärkeimpänä vaiheena, koska konstruktiivisessa tutkimusotteessa konstruktio ei aina välttämättä ratkaise asetettua ongelmaa, mutta tutkimuksen tulisi aina tuottaa teoreettista kontribuutiota tieteenalalla. Tämä viimeinen vaihe toteutetaan tutkimuksen Pohdinnassa (LUKU 6), jossa analysoidaan konstruktion ja koko tutkimuksen teoreettista kontribuutiota.

4.4 Konstruktion toteutus ja testaus

Tiedeyhteisöstä havaittiin puute, ettei tiettävästi ole olemassa konstruktiota, jolla voitaisiin kerätä laaja määrä verkkosivuja sekä tietoa niiden TLS-salauksesta. Tutkimuksessa otettiin alatavoitteeksi toteuttaa tämänkaltainen konstruktio ja testata sen toimivuus käytännössä.

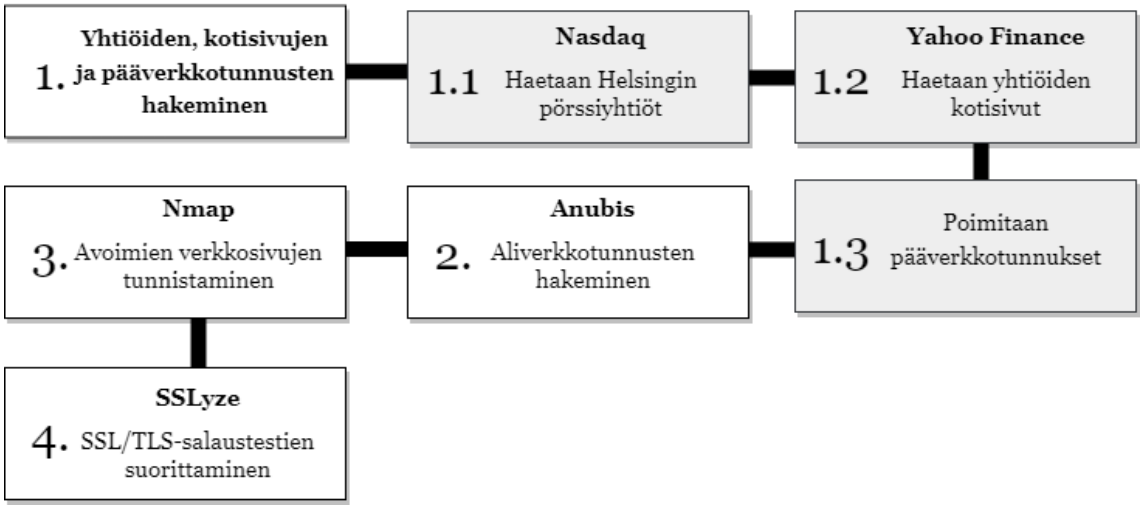
Tutkimuksessa kehitettiin konstruktio, jonka toimivuutta testattiin selvittämällä Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso, jonka selvittäminen asetettiin tutkimuksen päätavoitteeksi. Konstruktio toteutettiin kehittämällä työkalu, joka keräsi varsinaisen tutkimusaineiston päätavoitetta varten. Tässä luvussa kerrotaan, kuinka työkalu toteutettiin ja kuinka sen toimivuus testattiin.

Tutkimuksessa kehitetty työkalu etsii Helsingin pörssiyhtiöiden verkkotunnuksia julkisista lähteistä ja tekee TLS-salauksen tason tunnistavat testit niitä vasten. Heti tutkimuksen alusta lähtien oli odotuksena, että kohteita tulee löytymään tuhansia, joten konstruktion rakentamisen edellytyksenä oli, että se on hoidettava ohjelmallisesti, koska datan kerääminen manuaalisesti ei olisi ollut mitenkään mahdollista kohtuullisen ajan puitteissa.

Työkalu toteutettiin Python-ohjelmointikielellä ja se koostui neljästä eri vaiheesta: Helsingin pörssiyhtiöiden pääverkkotunnuksien hakeminen, aliverkkotunnusten noutaminen, avointen verkkosivujen tunnistaminen ja SSL/TLS-testien suorittaminen. Työkalun tuottamat tulokset tallennettiin MongoDB-tietokantaan, jonka avulla kerättyjä SSL/TLS-salauksen asetuksia saatiin vertailtua eri toimijoiden suositusten konsensukseen. Työkalun lähdekoodit

ovat julkaistu saataville GitHubissa (https://github.com/kiuru/pyTLScanner), josta löytyy samalla koko työkalun käytännön suoritusketju.

Seuraavassa kuviossa (KUVIO 8) esitellään työkalun suoritusketjun kokonaisuudessaan. Jokainen yksittäinen kohta kuvastaa yhtä työkalun osa-aluetta, jotka käydään erikseen vielä seuraavissa luvuissa läpi.



KUVIO 8 Työkalun suoritusketju.

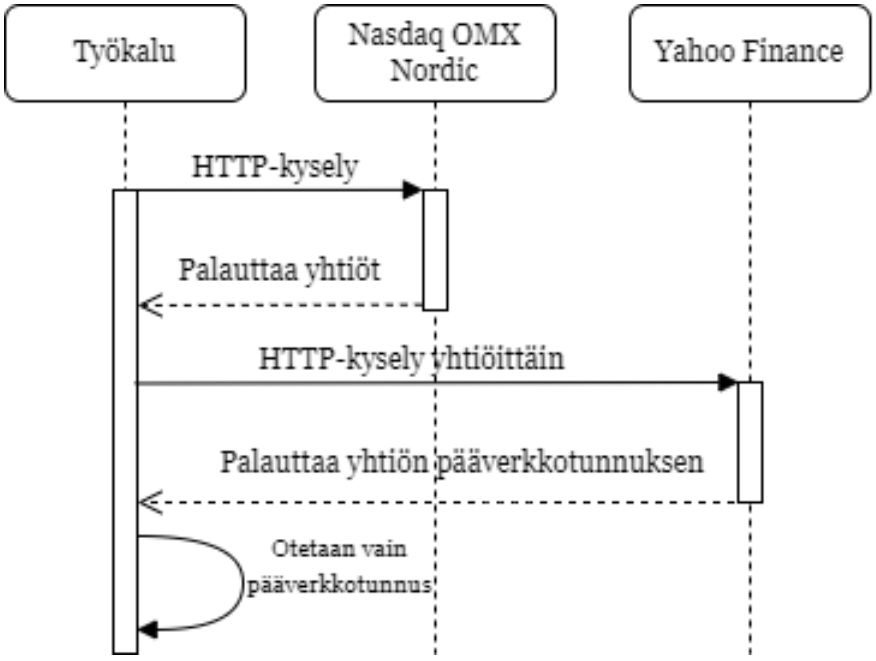
4.4.1 Yhtiöiden, kotisivujen ja pääverkkotunnusten hakeminen

Työkalun suoritusketjun ensimmäisessä vaiheessa haettiin Helsingin pörssiyhtiöt Nasdaq OMX Nordicin kotisivuilta. Pörssiyhtiöt löytyivät osakelajeittain, mikä tarkoittaa sitä, että osa yhtiöistä saattoi löytyä useamman kerran Nasdaq OMX Nordicin kotisivuilta. Löydetyt kaksoiskappaleet kuitenkin poistettiin myöhemmässä työkalun ajon vaiheessa.

Nasdaq OMX Nordicin sivustolta ei välttämättä aina löytynyt yhtiöiden kotisivuja, minkä takia yhtiöiden kotisivut haettiin Yahoo Finance -rajapinnasta. Nasdaq OMX Nordicin sivu piti sisällään yhtiön kaupankäyntitunnuksen, jonka avulla Yahoo Finance -rajapinnasta sai haettua yhtiölle kuuluvan kotisivun. Nasdaq OMX Nordicin sivustolta noudetun yhtiölistan avulla saatiin Yahoo Finance -rajapinnasta haettua Helsingin pörssiyhtiöille kotisivut.

Yhtiöiden kotisivujen verkkotunnuksessa käytetään joskus aliverkkotunnusta, kuten www-alkuista osoitetta. Kotisivujen verkkotunnuksesta karsittiin pois kaikki aliverkkotunnukset, jolloin jäljelle jäi pelkästään yhtiölle kuuluva pääverkkotunnus. Koska osakelajeista saattoi löytyä sama yhtiö useamman kerran, tässä vaiheessa jätettiin jäljelle vain uniikit pääverkkotunnukset, jottei samoja pääverkkotunnuksia löytyisi useampia.

Seuraavassa kuviossa (KUVIO 9) havainnollistetaan, kuinka työkalu kokonaisuudessaan nouti pääverkkotunnukset, joita hyödynnetään työkalun seuraavissa vaiheissa.



KUVIO 9 Työkalun pääverkkotunnusten haku

4.4.2 Aliverkkotunnusten hakeminen

Työkalun toisessa vaiheessa haettiin julkisista lähteistä yhtiölle kuuluvia aliverkkotunnuksia. Aliverkkotunnusten hakukriteerinä toimi yhtiöiden kotisivusta poimittu pääverkkotunnus. Tutkimuksen kohteiksi oli rajattu ainoastaan sellaiset verkkotunnukset, jotka pitävät sisällään Yahoo Finance -rajapinnassa ilmoitetusta kotisivusta löytyvän pääverkkotunnuksen. Todellisuudessa yhtiöt omistavat useita eri pääverkkotunnuksia, esimerkiksi fi-, com- tai net-päätteisiä verkkotunnuksia. Eri verkkotunnusten omistustietoja ei kuitenkaan ole mistään julkisesti saatavilla ja näin ollen ei ole olemassa mitään luotettavaa keinoa osoittaa, mitä eri pääverkkotunnuksia yhtiö omistaa. Tässä tutkimuksessa tehtiin oletus, että yhtiö omistaa sen verkkotunnuksen, jonka kotisivu löytyy Yahoo Finance -rajapinnasta.

Aliverkkotunnusten hakeminen hoidettiin Anubis-työkalulla. Anubis (Anubis, 2022) on alkuaan JonLuca DeCaron kehittämä työkalu, joka löytää eri aliverkkotunnuksia hyödyntäen julkisia lähteitä, muun muassa X509-varmenteen SAN-tietueita (engl. subject alternative name), DNSSEC:n (Domain Name System Security Extension) NSEC-vastaus (engl. next secure record) ja DNS-vyöhykesiirto (engl. zone transfer). Tutkimuksessa päädyttiin käyttämään Anubista, koska se osasi samalla hyödyntää useita muita tunnettuja verkkopalveluita, ja testeissä saatiin siten tuotettua kattava määrä aliverkkotunnuksia. Muihin verkkopalveluihin lukeutui muun muassa DNSDumpster, crt.sh, HackerTarget, Shodan, Censys, Sublist3r, Netcraft ja AnubisDB (Anubis, 2022). Anubiksen käytännön suoritustapa on kuvattu työkalun README.md-tiedostossa, joka löytyy työkalun lähdekoodista GitHub-sivuilta (https://github.com/kiuru/pyTLScanner/blob/main/README.md).

Seuraavassa luvussa käydään läpi, kuinka työkalu tunnisti avoinna olevat aliverkkotunnukset.

4.4.3 Avoimien verkkosivujen tunnistaminen

Edellisen vaiheessa löydetyt verkkotunnukset saattoivat pitää sisällään myös paljon vanhoja verkkosivuja, jotka ovat saattaneet poistua käytöstä. Joukossa saattoi myös olla verkkosivuja, jotka olivat ainoastaan yrityksen sisäisessä käytössä, tai verkkotunnus, jossa ei palvele WWW-palvelin. TLS-testaus pystytään suorittamaan ainoastaan olemassa olevaan palvelimeen, jossa pyörii WWW-palvelin portissa 443, joka on salatun HTTP-yhteyden oletusportti. Lopulliset avoinna olevat verkkosivut saatiin rajattua verkkoskannaamalla kaikki verkkotunnukset läpi ja testaamalla, kuuntelevatko ne porttia 443.

Tämän työkalun suorituskohdan jälkeen edettiin seuraavaan vaiheeseen, jossa lähdettiin suorittamaan varsinaisia SSL/TLS-salauksen testejä kerättyjä verkkosivuja vasten.

4.4.4 SSL/TLS-salaustestien suorittaminen

Työkalun viimeisessä vaiheessa suoritettiin SSL/TLS-salauksen testit verkkosivuja vasten. Työkalun ensimmäisessä versiossa testit suoritettiin Qualys Inc -yhtiön tarjoamalla SSL Labs -työkalulla, joka tekee tarvittavat SSL/TLS-salaustason mittaavat testit ja palauttaa niistä saadut tulokset. SSL Labs on myös ollut useassa aikaisemmassa verkkosivujen salauksen tason testausta toteuttavassa tutkimuksessa käytössä, minkä takia sen käyttäminen olisi soveltunut myös tähän tutkimukseen. Keskeiseksi ongelmaksi SSL Labsin kanssa kuitenkin ilmeni, että yksittäisen verkkosivujen ajaminen saattoi kestää jopa viisi minuuttia, jolloin koko ajo olisi saattanut kestää yli kuukauden. Tutkimuksen alkuvaiheessa ehdittiin myös huomata useaan kertaan, että SSL Labsilla oli rajattu prosessimäärä, jotka suorittavat TLS-salauksen testejä yhtäaikaisesti, mikä esti testauksen skaalautumisen isommalle kohdemäärälle. Kaikkien prosessien ollessa varattuina jouduttiin ensiksi odottamaan niiden valmistumista. Jo tutkimuksen alusta lähtien oli odotettavissa, että verkkosivuja tulee olemaan tuhansia, minkä vuoksi SSL Labs ei ollut riittävän suorituskykyinen ja skaalautuva tähän tutkimukseen.

Kun oli selvää, että tutkimuksen toteuttamiseen tarvittiin oma työkalu, lähdettiin ensiksi etsimään valmiita vaihtoehtoja. Tutkimuksen testit päädyttiin ajamaan Alban Diquetin aloitteesta kehitetyllä avoimen lähdekoodin projektilla nimeltään SSLyze, joka mahdollisti testien ajamisen suurelle määrälle verkkotunnuksia. Työkalun valinnassa painotettiin sitä, että sitä pystyttiin helposti kutsumaan Pythonilla, joka mahdollisti sen liittämistä osaksi tämän tutkimuksen konstruktion suoritusketjua. Valintaa tuki myös Manfredi ym. (2021) tekemä tutkimus, jossa SSLyze sai hyvät arvostelut.

4.4.5 Konstruktion validointi

Konstruktion toteuttamisen aikana ja sen jälkeen tutkimuksessa tehtiin työkalun validointia, jotta voitiin varmistua sen tulosten oikeellisuudesta. Työkalun validointia pystyttiin tekemään erikseen jokaisen neljän vaiheen kohdalla. Ensimmäisessä vaiheessa haettavat Helsingin pörssiyhtiöt oli mahdollista validoida yksitellen, koska yhtiöitä oli ainoastaan 133 kappaletta. Validoinnissa pystyttiin toteamaan, että kaikille yhtiöille löytyi sopiva kotisivu Yahoon Finance -rajapinnasta.

Toisessa vaiheessa haettiin aliverkkotunnukset Anubis-ohjelmalla, jonka tarkoitus oli löytää mahdollisimman paljon aliverkkotunnuksia. Kaikkia aliverkkotunnuksia ei voitu tarkistaa manuaalisesti, mutta aliverkkotunnuksia silmäillen oli mahdollista päätellä, että data oli validia. Aliverkkotunnuksia kokeiltiin selaimella satunnaisotannalla, ja tarkastettavia aliverkkotunnuksia oli noin kaksisataa. Määrä valittiin sen perusteella, että löydetyistä aliverkkotunnuksista siihen lukeutui noin kaksi prosenttia, joka koettiin riittäväksi määräksi tarkastettavia kohteita. Kaikki tarkastetut aliverkkotunnukset kuuluivat tarkastettavalle yhtiölle.

Anubiksen tuottamat aliverkkotunnukset pitivät sisällään myös toimimattomia tai vanhaksi jääneitä verkkosivuja, minkä vuoksi kolmannessa vaiheessa testattiin, mitkä verkkosivut vastaavat jotain. Tätä testaustyötä helpotettiin ajamalla verkkoskannaus 443-porttiin, jolla tunnistettiin, vastaako WWW-palvelimen oletusportista jokin palvelu. Tässäkin satunnaisotannalla tehdyt tarkastukset osoittivat lopullisen datan olevan validia. Satunnaisotanta tehtiin kuten edellisessä vaiheessa, ja tutkittavia kohteita oli noin kaksi sataa, joka oli toimivista verkkosivuista noin viisi prosenttia kohteista.

Viimeisessä neljännessä vaiheessa teimme testit avoinna oleviin verkkosivuihin. Satunnaisotannalla tarkastettiin lukuisia (noin 100 kpl) yksittäisiä verkkosivua ja niistä saatuja TLS-salauksen testituloksia. Tässä vaiheessa käytettiin hyväksi myös muita työkaluja, kuten SSL Labs, jolloin saatiin vertailtua SSLyzen ja SSL Labsin tuloksia keskenään ja täten validoitua testituloksia. Satunnaisotannalla verratut testitapaukset täsmäsivät keskenään.

Yksittäisten testauksien perusteella työkalun jokainen vaihe todettiin toimivaksi, joten siirryimme ajamaan kaikki työkalun vaiheet ja keräämään tutkimuksessa tarvittavaa aineistoa. Seuraavassa luvussa käsitellään tarkemmin tutkimuksen kannalta oleellisimmat asiat aineiston keruusta ja käsittelystä.

4.5 Tutkimusaineiston keruu ja käsittely

Tutkimuksen luotettavuuden ja pätevyyden kannalta on tärkeää dokumentoida mahdollisimman täsmällisesti, miten tutkimuksen aineisto on kerätty ja käsitelty (Hirsjärvi ym., 2016, s. 232). Tässä luvussa kerrotaan, miten ja milloin data

kerättiin, miten se tallennettiin ja miten sitä käsiteltiin, jotta saatiin kerättyä tätä tutkimusta varten käytettävä aineisto.

Tutkimuksen testit ajettiin DigitalOcean:lta hankitulta virtuaalipalvelimelta, jossa oli yksi prosessori, 1 Gt RAM, 25 Gt kiintolevytilaa ja käyttöjärjestelmänä Ubuntu 20.04. Työkalun ajon tulokset tallennettiin paikalliseen MongoDB-tietokantaan JSON-formaatissa myöhempää käsittelyä varten.

Työkalu ajettiin ajanjaksolla 27.9.2021 – 3.10.2021. Testit suoritettiin hankitulta palvelimelta kohteena oleviin verkkotunnuksiin, joille ajettiin seuraavat testit: X509-varmenteen haku, palvelimen tuettujen SSL/TLS-versioiden noutaminen, käytettiinkö TLSv1.3 version yhteydessä 0-RTT:ä, tukiko palvelin TLS-verkkoliikenteen pakkausta, tukiko verkkosivu HSTS:ää ja mitä elliptisen käyrän algoritmeja palvelin tuki. Työkalun keräämää esimerkkiaineistoa löytyy liitteistä (LIITE 3).

SSLyze-tuloksia käsiteltiin siten, että niitä vertailtiin teoriassa luotuja SSL/TLS-suosituksia vasten. Käytännössä tämä tehtiin käsin kirjoittamalla MongoDB-kyselyitä, jotka palauttivat vertailun tulokset. Vähimmäissuositukset oli rakennettu suoraan MongoDB-kyselyihin, jotka osasivat palauttaa vastaukseksi puutteelliset sivustot. MongoDB-kyselyitä muokkaamalla voisi jatkossakin saada tuotua päivitetyt suositukset mukaan tulosten vertailuun. Kaikki aineiston käsittelyyn liittyvät MongoDB-kyselyt löytyvät tutkimuksen liitteistä (LIITE 8 – LIITE 18).

Seuraavassa luvussa esitetään tutkimuksen tulokset eli tässä luvussa kerätty aineisto, joka käsiteltiin käyttäen mainittuja MongoDB-kyselyitä.

5 TUTKIMUKSEN TULOKSET

Tässä luvussa käsitellään tutkimuksessa kerätyt aineistot. Ensimmäisessä alaluvussa esitetään työkalun ajon eri vaiheissa syntyneet tulokset. Ensimmäisen alaluvun tarkoituksena on toteuttaa tutkimuksen alatavoite, eli validoida työkalun toimiminen suuremmassa mittakaavassa, sekä kerätä aineisto ensisijaista päätavoitetta varten.

Toisessa alaluvussa esitetään, kuinka moni työkalun keräämä verkkotunnus läpäisi asetetut TLS-salauksen suositukset, mikä oli ensimmäisen päätavoitteen kolmas tutkimuskysymys. Toisessa alaluvussa avataan myös tarkemmin jokainen suosituskategoria, millä vastataan ensimmäisen päätehtävän alakysymykseen siitä, mikä on yleisin syy, etteivät verkkotunnukset läpäisseet asetettuja TLS-salauksen suosituksia.

5.1 Työkalun ajon tulokset

Työkalun ajaminen koostui neljästä eri vaiheesta, jotka olivat Helsingin pörssiyhtiöiden kotisivujen hakeminen, aliverkkotunnusten hakeminen, avointen verkkosivujen tunnistaminen ja SSL/TLS-testien suorittaminen. Tässä luvussa esitetään näistä vaiheista saadut tulokset. Luvun lopputuloksena saatiin verkkotunnukset, joita verrataan SSL/TLS-salauksen suosituksiin seuraavassa luvussa.

Työkalun ajon ensimmäisessä vaiheessa haettiin Helsingin pörssin kaikkien yhtiöiden osakelajit, joista osa oli samojen yhtiöiden eri osakelajeja. Eri osakelajeja oli yhteensä 140 kappaletta, joille kaikille löytyi kotisivu. Koska osalla yhtiöistä oli Helsingin pörssissä useampia osakelajeja, oli tuloksissa myös samoja kotisivuja useaan kertaan. Tuloksista karsittiin pois kaikki kaksoiskappaleet, jolloin jäljelle jäi yhteensä 133 uniikkia yhtiötä ja kotisivua. Lista kaikista pörssiyhtiöistä ja niiden pääverkkotunnuksista löytyy liitteistä (LIITE 20). Osa yhtiöistä käytti kotisivunaan aliverkkotunnuksia, minkä takia kotisivuista poistettiin verkkotunnusten alimmat tasot, jotta saatiin yhtiöiden käyttämät pää-

verkkotunnukset. Näitä yhtiöiden pääverkkotunnuksia oli yhteensä 133 kappaletta ja niitä hyödynnettiin työkalun ajon toisessa vaiheessa. (TAULUKKO 7.)

TAULUKKO 7 Työkalun ajon ensimmäisen vaiheen tulokset (LIITE 4)

Nimi	Määrä
Osakelajeja	140
Yhtiöitä	133
Pääverkkotunnuksia	133

Työkalun ajon toisessa vaiheessa haettiin kaikille 133 pääverkkotunnukselle kaikki löytyvät verkkotunnukset ja näiden verkkotunnusten IP-osoitteet. Työkalu löysi yhteensä 7946 eri verkkotunnusta ja 5099 uniikkia IP-osoitetta. Osa löydetyistä verkkotunnuksista ohjautui samoihin IP-osoitteisiin, minkä takia uniikkeja IP-osoitteita oli vähemmän kuin verkkotunnuksia. (TAULUKKO 8.)

TAULUKKO 8 Työkalun ajon toisen vaiheen tulokset

Nimi	Määrä
Verkkotunnuksia	7946
IP-osoitteita	5099

Työkalun ajon kolmannessa vaiheessa tarkastettiin, mitkä uniikeista IP-osoitteista kuuntelivat porttia 443 ja vastasivat, jos niihin yritettiin ottaa yhteyttä. Työkalulle vastasi yhteensä 2552 eri IP-osoitetta. Näin ollen aktiivisia käytössä olevia verkkotunnuksia löytyi 4782 kappaletta. (TAULUKKO 9.)

TAULUKKO 9 Työkalun ajon kolmannen vaiheen tulokset (LIITE 5)

Nimi	Määrä
IP-osoitteet, jotka kuuntelivat porttia 443	2552
Verkkotunnukset, jotka ohjautuivat porttia 443 kuuntelevaan IP-osoitteeseen	4782

Työkalun neljännessä vaiheessa suoritettiin SSL/TLS-salauksen testit näitä 4782:aa verkkotunnusta vasten. Tässä testissä tuli virheitä 351 verkkotunnuksesta. Virheen aiheuttaneet verkkotunnukset poistettiin lopullisista jäljelle jääneistä verkkotunnuksista. Näin ollen työkalu löysi yhteensä 4431 validia verkkotunnusta. (TAULUKKO 10.)

TAULUKKO 10 Työkalun ajon neljännen vaiheen tulokset (LIITE 6)

Nimi	Määrä
Verkkotunnukset, joissa tuli virheitä	351
Validit verkkotunnukset	4431

Edellisessä vaiheessa syntyneet virheet koostuivat kuudesta eri virheestä, jotka olivat “NoneType object is not iterable”, “certificate_info”, “could not resolve”,

"bug in sslyze", "connectivity issue" ja "client certificate needed". Näitä saatuja virheitä tarkastellaan myöhemmin pohdintaosiossa. (TAULUKKO 11.)

TAULUKKO 11 Työkalun ajossa tulleet virheet (LIITE 6)

Virheen nimi	**Määrä**
NoneType object is not iterable	212 (60,4 %)
Client certificate needed	80 (22,8 %)
Bug in sslyze	39 (11,1 %)
Connectivity issue	12 (3,4 %)
Certificate info	6 (1,7 %)
Could not resolve	2 (0,6 %)
Yhteensä	**351**

Lopputuloksena työkalu löysi yhteensä 4431 verkkotunnusta, joiden salauksen tasoa verrataan asetettuihin suosituksiin seuraavassa luvussa.

5.2 Suositukset

Teoriaosuudessa muodostettiin konsensus SSL/TLS-salauksen suosituksista hyödyntämällä eri julkaisijoiden suosituksia. Näitä eri suosituskategorioita oli yhteensä kymmenen erilaista ja niitä vasten testattiin edellisessä luvussa löydettyjä valideja verkkotunnuksia. Tässä luvussa esitetään tulokset liittyen näihin testeihin. Suositusten tulokset on saatu käyttäen MongoDB-kyselyitä, jotka löytyvät tutkimuksen liitteistä (LIITE 8 – LIITE 18).

Seuraavassa taulukossa on esitettynä, kuinka moni löydetyistä 4431 verkkotunnuksesta läpäisi SSL/TLS-salauksen suositukset. Alla olevaan taulukkoon on koottu jokainen suosituskategoria, ja jokaiselle kategorialle on kerrottuna lukumäärä, kuinka monta verkkosivua täytti kyseisen kategorian suositukset. Taulukon viimeisenä rivinä on kaikista suosituskategorioista koottu yhteenveto, joka kertoo, että 155 verkkotunnusta läpäisi kaikki asetetut suositukset, ja että loput verkkotunnuksista eivät täyttäneet yhtä tai useampaa näistä suosituksista. (TAULUKKO 12.)

TAULUKKO 12 Kaikkien suosituskategorioiden tulokset (LIITE 8; LIITE 19)

Suosituskategoria	Suositusten mukainen	Alitti suositukset
SSL/TLS-versio	2398 (54,1 %)	2033 (45,9 %)
Salaussarjat	1181 (26,7 %)	3250 (73,3 %)
TLS-pakkaus	4426 (99,9 %)	5 (0,1 %)
0-RTT	4324 (97,6 %)	107 (2,4 %)
OCSP stapling	1328 (30,0 %)	3103 (70,0 %)
Varmenteen avaimen koko	4427 (99,9 %)	4 (0,1 %)
Varmenteen voimassaolo	4222 (95,3 %)	209 (4,7 %)
Varmenteen käyttöiän pituus	3723 (84,0 %)	708 (16,0 %)
Varmenteen yleinen nimi	4009 (90,5 %)	422 (9,5 %)
HSTS	1472 (33,2 %)	2959 (66,8 %)
Täytti kaikki suositukset	**155 (3,5 %)**	**4276 (96,5 %)**

Yksikään verkkotunnus ei alittanut kaikkia kategorioita (LIITE 19). Seuraavissa alaluvuissa käydään yksityiskohtaisemmin läpi jokainen suosituskategoria. Ensimmäiseksi otetaan tarkasteluun SSL/TLS-versiot.

5.2.1 SSL/TLS-versio

Asetettujen suositusten ensimmäisenä kohtana oli TLS-versio. Suosituksissa sallittiin ainoastaan TLS 1.2- ja TLS 1.3 -versioiden käyttö, joten jos verkkotunnus hyväksyi vanhempien versioiden käyttämisen, ei se täyttänyt asetettua suositusta. Alla olevassa taulukossa on esitettynä suosituksen alittaneet verkkotunnukset ja eri variaatiot, miten ne tukivat eri SSL/TLS-versioita (TAULUKKO 13).

TAULUKKO 13 TLS-versiosuosituksen alittaneet verkkotunnukset (LIITE 9)

Tuetut versiot	Määrä
TLS 1.0, TLS 1.1 ja TLS 1.2	1365 (67,1 %)
TLS 1.0, TLS 1.1, TLS 1.2 ja TLS 1.3	320 (15,7 %)
SSL 3.0, TLS 1.0, TLS 1.1 ja TLS 1.2	130 (6,4 %)
TLS 1.1 ja TLS 1.2	103 (5,1 %)
TLS 1.1, TLS 1.2 ja TLS 1.3	74 (3,6 %)
TLS 1.0 ja TLS 1.2	13 (0,6 %)
TLS 1.0	11 (0,5 %)
SSL 2.0, SSL 3.0 ja TLS 1.0,	7 (0,3 %)
SSL 3.0 ja TLS 1.0	3 (0,1 %)
SSL 2.0, SSL 3.0, TLS 1.0 ja TLS 1.2	3 (0,1 %)
SSL 3.0 ja TLS 1.2	2 (0,1 %)
SSL 3.0, TLS 1.0 ja TLS 1.2	1 (< 0,1 %)
TLS 1.0 ja TLS 1.1	1 (< 0,1 %)
Alitti suositukset	**2033**

Yhteensä 2033 verkkotunnusta tuki jotain muitakin TLS-versioita kuin pelkästään TLS 1.2:sta ja TLS 1.3:sta. Alla olevassa taulukossa on esitettynä ne verkkotunnukset, jotka täyttivät asetetun suosituksen ja miten nämä suositusten mukaiset verkkotunnukset tukivat eri SSL/TLS-versioita (TAULUKKO 14).

TAULUKKO 14 TLS-versiosuositusten mukaiset verkkotunnukset (LIITE 9)

Tuetut versiot	**Määrä**
TLS 1.2	1482 (61,8 %)
TLS 1.2 ja TLS 1.3	914 (38,1 %)
TLS 1.3	2 (0,1 %)
Suositusten mukaiset	**2398**

Lopputuloksena oli, että 2033 verkkotunnusta alitti asetetut suositukset ja 2398 verkkotunnusta oli suositusten mukaisia (TAULUKKO 13; TAULUKKO 14). Seuraavassa luvussa käsitellään salaussarjoja.

5.2.2 Salaussarjat

Suositusten toisena kohtana oli salaussarjat. Yksi verkkotunnus pystyy tukemaan useampia eri salaussarjoja, ja tässä suosituksessa tarkasteltiin, että kaikki verkkotunnuksen tukemat salaussarjat löytyvät suositeltujen salaussarjojen listalta. Jos yksikin verkkotunnuksen tukema salaussarja ei löytynyt suositeltujen salaussarjojen listalta, verkkotunnus alitti suositukset.

Suosituksissa sallittuina salaussarjoina oli vain TLS 1.2- ja TLS 1.3 -versioiden tukemat salaussarjat. Tästä syystä verkkotunnukset, jotka käyttivät TLS 1.2- ja TLS 1.3 -versioiden lisäksi muita TLS-versioita eivät voineet täyttää tätä salaussarjojen suositusta. Tämä siis käytännössä tarkoittaa sitä, että edellisessä luvussa TLS-versiosuosituksen alittaneet 2033 verkkotunnusta eivät pystyneet täyttämään tätä suositusta, joten kyseiset verkkotunnukset alittivat tämän suosituksen. (TAULUKKO 13)

Kaksi verkkotunnusta täytti suoraan salaussarjojen suositukset, koska ne tukivat pelkästään TLS 1.3 -versiota (TAULUKKO 14). Tämä johtui siitä, että suositukset eivät rajanneet pois yhtäkään TLS 1.3 -version salaussarjaa.

Otetaan tarkempaan tarkasteluun jäljelle jääneet 2396 kappaletta verkkotunnusta, jotka tukivat TLS 1.2 -versiota, sekä niiden tukemat TLS 1.2 -version salaussarjat (TAULUKKO 14). Suositeltujen salaussarjojen listalta löytyi yhteensä 33 eri TLS 1.2 -version salaussarjaa, joista 25 eri salaussarjaa oli kerätyssä datassa käytössä. Kerätystä datasta löytyi yhteensä 60 erilaista salaussarjaa, joista 25 oli suositeltuja salaussarjoja ja loput 35 salaussarjaa eivät täyttäneet suositusta.

Alla olevassa taulukossa on koottuna, kuinka monta verkkotunnuksen käyttämää salaussarjaa ei kuulunut suositeltujen salaussarjojen listaan. Esimerkiksi jos verkkotunnus tuki viittä eri salaussarjaa ja niistä kaikki viisi löytyi suositeltujen salaussarjojen listalta, verkkotunnuksella ei tällöin ollut yhtään suositusten alittavaa salaussarjaa ja se täytti suosituksen. Jos taas verkkotunnuksen

tukemista viidestä salaussarjasta vain kaksi täytti suosituksen, silloin kolme salaussarjaa alitti suosituksen, jolloin verkkotunnus ei täyttänyt suositusta. (TAULUKKO 15.)

TAULUKKO 15 Suositusten alittaneiden salaussarjojen lukumäärä (LIITE 10)

Kuinka monta suositusten alittavaa salaussarjaa verkkotunnuksella käytössä	**Määrä**
0	1179 (49,2 %)
1–5	367 (15,3 %)
6–10	774 (32,3 %)
11–15	28 (1,2 %)
16–20	32 (1,3 %)
21	1 (< 0,1 %)
26	15 (0,6 %)
Yhteensä	**2396**

Tarkasteltavana olevista, TLS 1.2 -versiota käyttävistä verkkotunnuksista 1179 verkkotunnusta tuki vain salaussarjoja, jotka kuuluivat suositeltujen salaussarjojen listaan. Loput 1217 verkkotunnusta tuki salaussarjoja, joista yksi tai useampi salaussarja ei kuulunut suositeltuihin salaussarjoihin ja näin ollen alitti suositukset. Seuraavaksi kootaan tämän luvun suosituksen alittaneiden verkkotunnusten ja suositusten mukaisten verkkotunnusten lukumäärät.

Yhteensä 3250 verkkotunnusta alitti tässä luvussa asetetun salaussarjojen suosituksen. Näistä suosituksen alittavista verkkotunnuksista 2011 verkkotunnusta tuki muitakin TLS-versioita kuin vain TLS 1.2:sta ja TLS 1.3:sta. Verkkotunnuksista 1217 tuki salaussarjoja, jotka eivät kuuluneet suositeltujen salaussarjojen listaan. Loput 22 verkkotunnusta ei tukenut ollenkaan TLS 1.2- tai TLS 1.3 -versiota. (TAULUKKO 16.)

TAULUKKO 16 Salaussarjasuositukset alittaneet verkkotunnukset

Miksi alitti suosituksen	**Määrä**
Tuki muitakin versioita kuin TLS 1.2 tai TLS 1.3	2011 (61,9 %)
Tuki salaussarjaa, joka ei kuulunut suosituksiin	1217 (37,4 %
Ei tukenut TLS 1.2 tai TLS 1.3	22 (0,7 %)
Alitti suositukset	**3250**

Yhteensä 1181 verkkotunnusta oli suositusten mukaisia. TLS 1.2 -versiota käyttävistä verkkotunnuksista 1179 verkkotunnusta käytti vain salaussarjoja, jotka kuuluivat suositeltujen salaussarjojen listaan. Kaksi verkkotunnusta läpäisi suosituksen, koska ne tukivat vain TLS 1.3-versiota, jossa kaikki salaussarjat olivat suositusten mukaisia. (TAULUKKO 17.)

TAULUKKO 17 Salaussarjasuositusten mukaiset verkkotunnukset

Miksi täytti suositukset	Määrä
Tuki vain salaussarjoja, jotka kuuluivat suosituksiin	1179 (99,8 %)
Tuki vain TLS 1.3 versiota	2 (0,2 %)
Suositusten mukaiset	**1181**

Yhteensä 4431 verkkotunnuksesta 3250 verkkotunnusta alitti annetut salaussarjojen suositukset (TAULUKKO 16). Loput 1181 verkkotunnusta oli salaussarjoihin asetettujen suositusten mukaisia (TAULUKKO 17).

5.2.3 TLS-pakkaus

TLS-pakkaus oli suositusten kolmantena kohtana. TLS-pakkaus on vanhentunut lisäosa, joka ei ole enää tuettu teknologia TLS 1.3 -versiossa (E. Rescorla, 2018, s. 9). Tästä syystä verkkotunnukset, jotka tukivat vain TLS 1.3 -versiota täyttivät tämän suosituksen jo automaattisesti, koska niissä ei edes pystytty laittamaan TLS-pakkausta päälle. Näitä verkkotunnuksia oli kerätyssä datassa kaksi kappaletta (TAULUKKO 14).

TLS-pakkaus voi olla päällä tai pois päältä, mikä ilmaistiin kerätyssä datassa arvoilla true tai false. True tarkoitti, että TLS-pakkaus oli päällä, eli verkkotunnus ei täyttänyt suosituksia, kun taas false tarkoitti, että TLS-pakkaus oli poissa päältä, eli verkkotunnus täytti asetetun suosituksen. (TAULUKKO 18.)

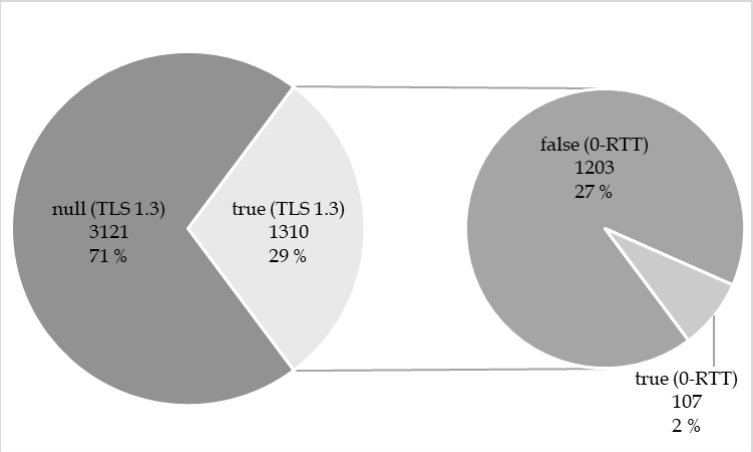
TAULUKKO 18 Verkkotunnuksien TLS-pakkauksen arvot (LIITE 11)

TLS-pakkauksen arvo	Määrä
false	4426 (99,9 %)
true	5 (0,1 %)
Yhteensä	**4431**

Toisien sanoen viisi verkkotunnusta oli asettanut TLS-pakkauksen päälle, eli alitti asetetun suosituksen. Loput 4426 verkkotunnusta oli asettanut TLS-pakkauksen tukemisen pois päältä, eli täytti asetetun suosituksen. (TAULUKKO 18.) Seuraava luku käsittelee 0-RTT-suositusta, joka tuotiin uutena lisäosana TLS 1.3 -versioon.

5.2.4 0-RTT

Suositusten neljäntenä kohtana oli 0-RTT, jonka pitäisi suositusten mukaan olla kytketty pois päältä. 0-RTT -lisäosa esiteltiin TLS 1.3 -versioon, minkä takia sen käyttöönottaminen vaatii sen, että verkkotunnus tukee TLS 1.3 -version käyttöä. Kerätyssä datassa oli 3121 verkkotunnusta, jotka eivät tukeneet TLS 1.3 -versiota (TAULUKKO 13; TAULUKKO 14). Jäljelle jäi 1310 verkkotunnusta, jotka tukivat TLS 1.3 -version käyttöä. Näistä 1310 verkkotunnuksesta 1203 verkkotunnuksella oli 0-RTT:n poissa päältä ja 107:llä oli se päällä (KUVIO 10).

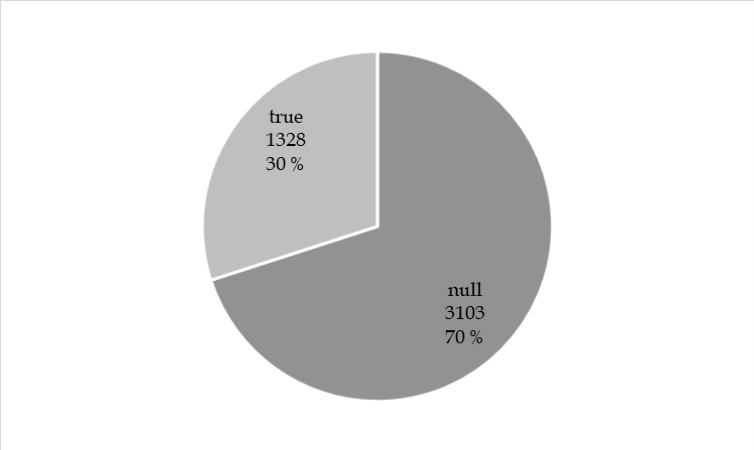


KUVIO 10 Kuinka moni verkkotunnus tuki TLS 1.3 -versiota ja oli kytkenyt 0-RTT:n päälle (LIITE 9; LIITE 12)

Kerätyistä verkkotunnuksista 107 verkkotunnusta alitti asetetun suosituksen, koska ne olivat asettaneet 0-RTT:n päälle. Loput 4324 verkkotunnusta oli suositusten mukaisia, koska 0-RTT oli poissa päältä. (KUVIO 10.)

5.2.5 OCSP stapling

Suositusten viidentenä kohtana oli OCSP stapling. Tässä suosituksessa OCSP stapling voi olla asetettu päälle, jolloin verkkotunnus on suositusten mukainen, tai OCSP stapling voi olla pois päältä, jolloin verkkotunnus ei täytä suosituksia. Kerätyssä datassa nämä arvot ovat true eli OCSP stapling on päällä tai null eli OCSP on poissa päältä. Alla olevassa kuviossa on esitettynä verkkotunnuksista kerätyt arvot kuten ne ovat datassa (KUVIO 11).



KUVIO 11 Verkkotunnuksien OCSP staplingin arvot (LIITE 13)

OCSP stapling -suosituksen täytti 1328 verkkotunnusta ja suosituksen alitti 3103 verkkotunnusta. Seuraavassa luvussa käsitellään verkkotunnusten varmenteiden avaimien kokoa.

5.2.6 Varmenteen avaimen koko

Suositusten kuudentena kohtana oli varmenteen avaimen koko. Suosituksissa asetettiin varmenteen vähimmäisvaatimukseksi RSA-algoritmia käyttävä 2048-bittinen avain, joka vastaa 224–255 bittisen elliptisen käyrän avainta (TAULUKKO 4). Kerätystä datasta neljä verkkotunnusta alitti asetetun suosituksen, koska ne tukivat RSA-algoritmia käyttäviä 1024-bittisiä avaimia (TAULUKKO 19).

TAULUKKO 19 Varmenteen avaimen koko -suosituksen alittaneet verkkotunnukset (LIITE 14)

Tuetut avaimet	**Määrä**
RSAPublicKey 1024 bit	4 (100 %)
Alitti suositukset	**4**

Suositusten mukaisia tai sitä vahvempia avaimia käyttäviä verkkotunnuksia oli yhteensä 4427. Verkkotunnukset voivat myös tarjoilla käyttäjilleen useampia varmenteita, jotka käyttävät eri avaimia. Osa suositusten mukaisista verkkotunnuksista tukikin useampia eri avaimia, mutta yksikään näistä käytetyistä avaimista ei alittanut avaimen koolle annettua suositusta. Taulukossa on esitettynä suositusten mukaiset verkkotunnukset ja niiden tukemat eri avainvariaatiot. (TAULUKKO 20.)

TAULUKKO 20 Varmenteen avaimen koko -suosituksen täyttäneet verkkotunnukset (LIITE 14)

Tuetut avaimet	**Määrä**
RSAPublicKey 2048 bit	3716 (83,9 %)
RSAPublicKey 4096 bit	424 (9,6 %)
EllipticCurvePublicKey 256 bit ja RSAPublicKey 2048 bit	192 (4,3 %)
EllipticCurvePublicKey 256 bit	42 (0,9 %)
EllipticCurvePublicKey 384 bit	15 (0,3 %)
RSAPublicKey 3072 bit	10 (0,2 %)
RSAPublicKey 4096 bit ja RSAPublicKey 4096 bit	10 (0,2 %)
RSAPublicKey 2048 bit ja RSAPublicKey 2048 bit	9 (0,2 %)
RSAPublicKey 8192 bit	8 (0,2 %)
RSAPublicKey 2048 bit, RSAPublicKey 2048 bit ja RSAPublicKey 2048 bit	1 (< 0,1 %)
Suositusten mukaiset	**4427**

Annetun varmenteen avaimen koko -suosituksen alitti neljä verkkotunnusta, ja 4427 verkkotunnusta oli suositusten mukaisia (TAULUKKO 19; TAULUKKO 20). Seuraavassa luvussa tarkastellaan verkkotunnusten käyttämien varmenteiden voimassaoloaikoja.

5.2.7 Varmenteen voimassaolo

Suositusten seitsemäntenä kohtana oli varmenteen voimassaolo, jolla tarkoitetaan sitä, oliko varmenne voimassa, kun tutkimusaineistot kerättiin. Varmenteen voimassaoloon kuuluu varmenteen alkamis- ja päättymisaika.

Varmenteen voimassa oleva alkamispäivä ei saa olla tulevaisuudessa, koska tällöin varmenne ei olisi vielä voimassa. Päivämäärä, johon alkamispäivää verrattiin, oli 3.10.2021, koska tuona päivämääränä kerättiin viimeisten verkkotunnusten tiedot. Alkamispäivä ei siis saa olla 3.10.2021 jälkeen, koska muuten alkamispäivä olisi ollut tulevaisuudessa ja varmenne ei olisi vielä astunut voimaan, eikä toisin sanoen täyttäisi voimassaoloajan suositusta.

Verkkotunnuksilla käytössä olevien varmenteiden alkamispäivämäärät asettuivat välille 27.12.2013 – 3.10.2021 (LIITE 15). Yhdenkään verkkotunnuksen varmenteen voimassaolo ei ollut tulevaisuudessa ja kaikkien verkkotunnusten varmenteet olivat astuneet voimaan, joten kaikki 4431 verkkotunnusta olivat suositusten mukaisia.

Varmenteen päättymispäivämäärä kertoo ajan, jonka jälkeen varmenne ei ole enää voimassa. Tämä päättymispäivämäärä ei saa olla menneisyydessä, koska tällöin varmenne olisi vanhentunut ja voimassaolo olisi päättynyt. Päivämäärä, johon päättymispäivämäärää verrattiin, oli 27.9.2021, koska sinä päivänä aloitettiin verkkotunnuksista tietojen kerääminen.

Verkkotunnusten päättymispäivämäärät asettuivat välille 10.05.2015 – 31.12.2039 (LIITE 15). Tästä päätellen osa verkkotunnuksien varmenteista oli ehtinyt vanhentua ennen 27.9.2021. Tästä syystä otetaan verkkotunnusten varmenteiden päättymispäivämäärät tarkempaan tarkasteluun, jotta saadaan selvitettyä, kuinka monen verkkotunnuksen varmenne oli vanhentunut, eli alitti asetetun suosituksen.

Yhdellä verkkotunnuksella voi olla käytössä useampia varmenteita, joita verkkotunnus tarjoilee käyttäjilleen. Kerätyssä datassa oli 211 verkkotunnusta, jotka tarjoilivat kahta eri varmennetta, sekä yksi verkkotunnus, joka tarjoili kolmea eri varmennetta. Lopuilla 4219 verkkotunnuksella oli vain yksi käytössä oleva varmenne. (LIITE 14.)

Voimassaoloajan suosituksessa on vaatimuksena, että verkkotunnuksen kaikki käytössä olevat varmenteet ovat voimassa. Jos verkkotunnuksen yksikin käytössä oleva varmenne ei ole voimassa, ei verkkotunnus täytä asetettua voimassaolon kriteeriä ja alittaa suosituksen. Esimerkiksi verkkotunnuksella, jolla on käytössä kaksi varmennetta, ja ensimmäisen varmenteen päättymisaika on 24.1.2022 ja toisen varmenteen päättymisaika on 10.2.2021, on päättymispäivä 10.2.2021 menneisyydessä, joten kyseinen verkkotunnus ei täyttäisi voimassaoloajan päättymisajan suositusta, koska toinen varmenteista on vanhentunut.

Alla olevassa taulukossa on listattuna kaikki verkkotunnukset, jotka eivät täyttäneet varmenteen voimassaoloajan päättymisajan kriteeriä. Taulukossa on esitettynä, kuinka kauan sitten verkkotunnuksen varmenne on mennyt vanhaksi, kun tarkasteluajankohta oli 27.9.2021. Jos verkkotunnuksella oli käytössä useampia varmenteita kuin yksi, taulukossa esitetään se, jonka päättymispäivä

on kauimpana menneisyydessä. Taulukossa vuoden mittana käytettiin 365 päivää, joka oli päivien lukumäärä vuonna 2021. Jos varmenteen päättymispäivämäärä oli päivänkin yli vuoden menneisyydessä, kuului verkkotunnus kahden vuoden kategoriaan; jos taas yli kaksi vuotta, se kuului kolmannen vuoden kategoriaan, ja niin edelleen. (TAULUKKO 21.)

TAULUKKO 21 Varmenteen päättymisajan alittaneet verkkotunnukset (LIITE 15)

Kuinka monta vuotta sitten verkkotunnuksen varmenne on vanhentunut	**Määrä**
1	98 (46,9 %)
2	54 (25,8 %)
3	31 (14,8 %)
4	13 (6,2 %)
5	6 (2,9 %)
6	5 (2,4 %)
7	2 (1,0 %)
Yhteensä	**209**

Yhteensä 209 verkkotunnusta ei täyttänyt varmenteen päättymispäivän suositusta, koska niiden voimassaoloaika oli päättynyt. Seuraavaan taulukkoon on listattu 4222 verkkotunnusta, jotka täyttivät tämän päättymispäivän suosituksen (TAULUKKO 22), eli verkkotunnukset, joiden kaikki käytössä olevat varmenteet olivat vielä voimassa 27.9.2021 tai sen jälkeen. Taulukossa on esitettynä, kuinka kaukana tulevaisuudessa verkkotunnuksen varmenteen päättymispäivämäärä on vuosissa. Jos varmenne oli voimassa 365 päivää tai alle, se kuului yhden vuoden kategoriaan; jos varmenne oli voimassa yli 365 päivää, se kuului kategoriaan kaksi vuotta, ja niin edelleen. Jos verkkotunnuksella oli käytössä useampia varmenteita, on taulukossa esitettynä se varmenne, jonka päättymispäivämäärään oli vähemmän aikaa.

TAULUKKO 22 Varmenteen päättymisajan täyttäneet verkkotunnukset (LIITE 15)

Kuinka monta vuotta verkkotunnuksen varmenne on vielä voimassa	**Määrä**
1	4012 (95,0 %)
2	188 (4,5 %)
3	1 (< 0,1 %)
4	5 (0,1 %)
5	3 (0,1 %)
6	4 (0,1 %)
8	2 (0,0 %)
9	3 (0,1 %)
10	3 (0,1 %)
19	1 (< 0,1 %)
Yhteensä	**4222**

Alla olevassa taulukossa on esitettynä omina kohtinaan voimassaoloajan alkamis- ja päättymispäivät, joita käsiteltiin tässä luvussa, sekä yhteenvetona, mitkä verkkotunnukset täyttivät ja alittivat voimassaoloajan suositukset TAULUKKO 23).

TAULUKKO 23 Kuinka moni verkkotunnus täytti voimassaoloajan suositukset (LIITE 15)

Suositusosio	**Suositusten mukainen**	**Alitti suositukset**
Varmenteen alkamispäivämäärä	4431 (100,0 %)	0 (0,0 %)
Varmenteen päättymispäivämäärä	4222 (95,3 %)	209 (4,7 %)
Täytti voimassaolonajan suositukset	**4222 (95,3 %)**	**209 (4,7 %)**

Kaikki verkkotunnukset olivat astuneet jo voimaan, joten kaikki 4431 verkkotunnusta täytti varmenteen alkamispäivämäärän suosituksen. Varmenteen päättymispäivämäärässä suosituksen alitti yhteensä 209 verkkotunnusta. Voimassaoloajan suositukset alitti 209 verkkotunnusta, ja loput 4222 verkkotunnusta täyttivät voimassaoloajan suositukset (TAULUKKO 23).

Seuraavassa luvussa otetaan tarkasteluun suositus varmenteen käyttöiän pituudesta, jonka laskemisessa hyödynnetään tässä luvussa käsiteltyjä varmenteen alkamis- ja päättymispäivämääriä.

5.2.8 Varmenteen käyttöiän pituus

Suositusten kahdeksantena kohtana oli varmenteen käyttöikä, eli kuinka kauan varmennetta voi käyttää. Käyttöikä laskettiin päivien lukumääränä varmenteen voimassaoloajan alkamispäivämäärästä varmenteen voimassaoloajan päättymispäivämäärään. Suosituksissa käyttöiän maksimipituudeksi asetettiin 398 päivää, jonka ylittävät varmenteet eivät täytä käyttöiän suosituksia. Tämän käyttöikäsuosituksen täytti yhteensä 3723 verkkosivua ja alitti yhteensä 708 verkkosivua (TAULUKKO 24).

TAULUKKO 24 Varmenteen käyttöikäsuosituksen alittaneet verkkotunnukset (LIITE 16)

Varmenteen käyttöikä päivissä	**Määrä**
398 päivää tai alle	3723 (84,0 %)
399–825 päivää	639 (14,4 %)
Yli 825 päivää	69 (1,6 %)
Alitti suositukset	**2281**

Yllä on esitetty verkkosivut, jotka noudattavat nykyistä CA/Browser Forumin (2019) äänestystulosta, joka oli 398 päivää, mutta siitä käy myös ilmi aikaisempi CA/Browser Forumin (2017) äänestystulos, jossa päätettiin 825 päivän enimmäisajasta. Verkkosivujen varmenteissa lyhyin käyttöikä oli 30 päivää ja pisin käyttöikä oli peräti yli 20 vuotta.

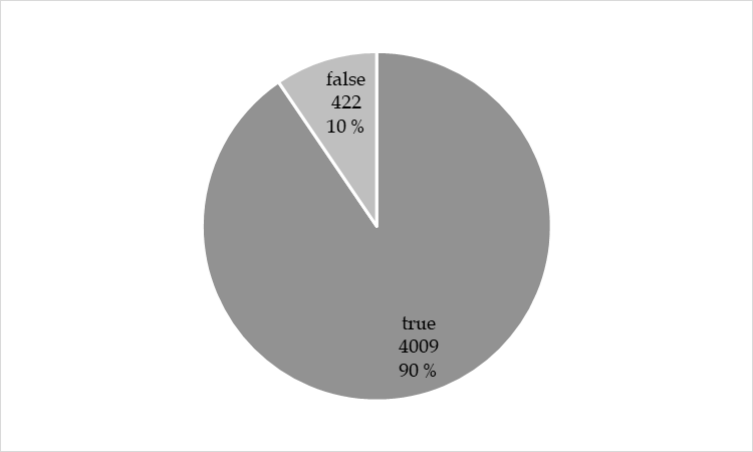
Jos verkkotunnuksella oli useampia varmenteita käytössä, tässä suosituksessa otettiin tarkasteluun verkkotunnuksen varmenteista se, jolla oli pisin käyttöikä. Mikäli verkkotunnuksella oli kaksi varmennetta, joista ensimmäisen käyttöikä oli 30 päivää ja toisen varmenteen käyttöikä oli 399 päivää, oli toisen varmenteen käyttöikä yli maksimipituuden, joten verkkotunnus ei täyttänyt asetettua suositusta.

Varmenteen käyttöiän pituus -suosituksen täytti 3723 verkkosivua ja alitti 708 verkkosivua (TAULUKKO 24). Seuraava luku käsittelee varmenteeseen liittyvää yleistä nimeä.

5.2.9 Varmenteen yleinen nimi

Varmenteen yleinen nimi oli suositusten yhdeksäntenä kohtana. Tässä suosituksessa tarkasteltiin sitä, että verkkotunnus käyttää varmennetta, joka on sille myönnetty. Eli esimerkiksi sivu.fi-verkkotunnuksen pitää käyttää varmennetta, joka on sivu.fi-verkkotunnuksen käyttöön tarkoitettu, eikä esimerkiksi sivun esimerkki.fi käyttöön tarkoitettu. Verkkotunnus ei täytä asetettua suositusta, jos se käyttää sille kuulumatonta varmennetta.

Datassa tämä tieto on esitetty true- tai false-arvolla. True-arvo tarkoittaa sitä, että verkkotunnuksen varmenne kuuluu sille. False-arvo tarkoittaa, ettei varmenne ole tarkoitettu kyseisen verkkotunnuksen käyttöön (KUVIO 12).



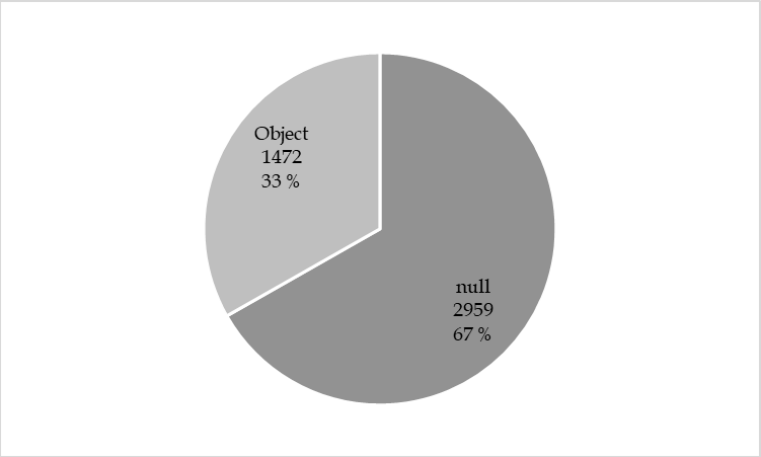
KUVIO 12 Täsmääkö verkkotunnuksen nimi varmenteen yleiseen nimeen (LIITE 17)

Kerätyssä datassa 422 verkkotunnusta ei täyttänyt asetettua suositusta, koska ne käyttivät varmenteita, joiden yleisenä nimenä oli käytetty jotakin muuta kuin kyseisen verkkotunnuksen nimeä. Suosituksen täyttivät 4009 verkkotunnusta, joilla verkkotunnus oli varmenteen yleisenä nimenä. (KUVIO 12.)

5.2.10 HSTS

Suositusten viimeisenä kriteerinä oli HSTS. Tässä suosituksessa katsottiin, löytyykö datasta HSTS:ään liittyvä objekti, jonka sisällä on HSTS:ään liittyviä eri

asetuksia. Suosituksissa ei otettu tarkemmin kantaa HSTS:n eri asetuksiin, joten tässä suosituksessa tarkasteltiin vain, onko verkkotunnuksella HSTS päällä, eli löytyykö HSTS:n objekti verkkotunnuksen datasta. Jos verkkotunnukselta ei löytynyt objektia, oli HSTS:n asetuksen arvo null eli HSTS-asetusta ei ole laitettu päälle. Alla olevasta kuviosta näkee, että HSTS:ään liittyviä objekteja löytyi yhteensä 1472 verkkotunnuksesta, eli HSTS oli asetettu päälle, ja null arvoja, eli HSTS oli kytkettynä pois päältä, 2959 verkkotunnuksesta (KUVIO 13).



KUVIO 13 Verkkotunnuksen HSTS:n arvo (LIITE 18)

Näin ollen 1472 verkkotunnusta oli suosituksen mukaisia ja 2959 verkkotunnusta alitti HSTS:ään liittyvän suosituksen (KUVIO 13). Seuraavassa luvussa tulkitaan tässä luvussa kerättyjä tuloksia sekä käsitellään tutkimusta kokonaisuutena.

6 POHDINTA

Tässä luvussa kootaan yhteen tutkimuksen tulokset, verrataan tuloksia aikaisempiin tutkimuksiin sekä käsitellään tutkimuksen päätavoite ja alatavoite. Päätavoitteelle ja alatavoitteelle oli asetettu omat tutkimuskysymyksensä, joihin kaikkiin vastataan omissa alaluvuissaan. Luvussa lisäksi arvioidaan tutkimuksen toteutusta, reliabiliteettia ja validiteettia. Lopuksi vielä käydään läpi, mitä käytännön ja tieteen kontribuutiota tässä tutkimuksesta saatiin aikaan.

Päätavoite edellytti alatavoitteen saavuttamista, mikä liittyy tutkimuksen yhteydessä kehitettyyn työkaluun, jolla pystyttiin keräämään tarvittavat verkkosivut ja suorittamaan TLS-salauksen testit suureen joukkoon verkkosivuja. Tutkimuksen alatavoitetta ja siihen liittyviä tutkimuskysymyksiä tarkastellaan seuraavassa luvussa.

6.1 Alatavoite

Tutkimuksen alatavoitteena oli toteuttaa erillinen työkalu, jolla pystytään keräämään tuhansia verkkosivuja ja tietoa niiden TLS-salauksesta, koska tämänkaltaista työkalua ei ole tiettävästi tiedeyhteisön käytössä. Tutkimuksen päätavoitteen selvittäminen vaati myös työkalun, jotta tutkimusaineisto saatiin kerättyä. Työkalun toimintaa arvioitiin seuraavista tulokulmista: onko työkalun keräämä data validia, mahdollistaako työkalu aineiston keräämisen tämän kaltaista tutkimusta varten ja miten työkalun toimintaa voitaisiin kehittää. Seuraavissa alaluvuissa vastataan näihin alatavoitteen tutkimuskysymyksiin.

6.1.1 Onko työkalun keräämä data validia?

Työkalun suoritusketju koostui neljästä eri vaiheesta, jotka olivat pääverkkotunnusten haku, aliverkkotunnusten noutaminen, avoimien sivujen tunnistaminen ja TLS-salauksen testien suorittaminen. Vaiheet olivat toisistaan riippu-

mattomia, joten niiden toimintaa pystyttiin arvioimaan itsenäisesti työkalun tekovaiheessa.

Työkalu löysi yhteensä 4431 avointa verkkosivua, jotka kerättiin 133 pörssiyhtiöltä. Verkkosivuja löytyi keskimäärin 33 kappaletta yritystä kohden. Työkalun suorituksen aikana havaittiin yhteensä 351 virheilmoitusta, joka oli 8 % testatuista verkkosivuista. Virheet jakautuivat kuuteen eri virheilmoitukseen, jotka olivat: "NoneType object is not iterable", "certificate_info", "could not resolve", "bug in sslyze", "connectivity issue" ja "client certificate needed". Osa virheilmoituksista oli tyypiltään sellaisia, mihin ei olisi pystynyt vaikuttamaan; esimerkiksi "could not resolve" viittasi siihen, että verkkotunnuksella ei ollut IP-osoitetta asetettu DNS-palvelimeen. Osa virheilmoituksista ("NoneType object is not iterable") oli tyypiltään sellaisia, johon olisi voinut löytyä ratkaisu työkalusta. Virheilmoitukset, joihin ei olisi pystynyt vaikuttamaan, ilmaantuivat, kun työkalu ei saanut yhteyttä verkkosivuun, tai sitten osa verkkosivuista saattoi vaatia verkkoselaimelta omaa varmennetta ("client certificate needed"), mikä esti osan haluttujen testien suorittamisesta. Osa virheilmoituksesta vaikutti viittaavan ongelmiin datan tallennuksessa ("NoneType object is not iterable") ja näihin työkalun kehityksessä olisi pystytty vaikuttamaan, mutta niiden korjaamista ja tulosten uudelleenkeruuta ei nähty relevanttina hyötyynsä nähden. Virheen tuottaneet verkkosivut muodostivat kokonaisuuden kannalta sen verran pienen osan, että niillä ei nähty olevan käytännön vaikutusta tuloksiin.

Työkalun suoritusketju oli mahdollista validoida vaihe vaiheelta, millä oli mahdollista vahvistaa datan validiteetti. Luvussa 4.4.5 käytiin läpi, kuinka tutkimusten testitulosten validiteetti on voitu varmistaa jokaisen vaiheen kohdalla.

Datan validoinnin perusteella voidaan todeta, että työkalu löysi Helsingin pörssiyhtiöille kuuluvia aliverkkotunnuksia, eikä TLS-salauksesta onnistuneesti kerätystä aineistosta löytynyt yhtäkään virhettä. Näin ollen aineisto voidaan olettaa virheettömäksi ja työkalun keräämä data validiksi.

6.1.2 Mahdollistaako työkalu aineiston keräämisen tämänkaltaista tutkimusta varten?

Tässä tutkimuksessa kehitetyllä työkalulla kerättiin TLS-salauksen tiedot 4431 verkkosivusta, ja koko tutkimusaineiston kerääminen tehtiin vain yhden viikon aikana. Vastaava aika ei testiemme mukaan olisi ollut mahdollista SSL Labs -verkkosivulla, joka oli suosiossa muissa tutkimuksissa. Tämän vuoksi nähtiin tarvetta kehittää työkalu, jota tutkijat voivat tulevaisuudessa käyttää. Useampi aikaisempi tutkimus oli tehty käyttäen hyväksi SSL Labs -verkkosivua (Ali & Murah, 2018; Silva & Fonte, 2019; Strzelecki & Rizun, 2020), mutta niissä verkkosivujen määrät olivat noin sata kappaletta, joka oli vielä sellaisissa rajoissa, että se pystyttiin toteuttamaan manuaalisesti ajettuna. Tuhansien verkkosivujen käsin ajaminen vaatisi liikaa aikaa, eikä olisi siten järkevää toteuttaa.

Työkalun tulokset voitiin todeta valideiksi, ja työkalun avulla saatiin kerättyä aineisto, joka esiteltiin luvussa 5. Nämä seikat osoittavat työkalun toimivuuden tämänkaltaisessa tutkimuksessa.

Vaikka työkalu mahdollisti tutkimusaineiston keruun tässä tutkimuksessa, se ei silti ole kaiken kattava. Työkaluun liittyvistä rajoitteista kerrotaan lisää seuraavassa alaluvussa.

6.1.3 Miten työkalun toimintaa voitaisiin kehittää?

Työkalu on ensisijaisesti suunniteltu tätä tutkimusta varten, mikä näkyy verkkosivujen haussa ja testien teossa. Työkalun toiminta koostui useasta eri vaiheesta, jotka on mahdollista irrottaa toisistaan, mikä mahdollistaa työkalun laajemman käytön muuhun tarkoitukseen. Esimerkiksi verkkotunnusten haussa olisi vapaasti käytettävissä mikä tahansa muu pääverkkotunnuksista koostuva lista, joka mahdollistaisi työkalun ajamisen muuta tarkoitusta varten. Sen sijaan ensimmäisen vaiheen saattaminen yleisluontoiseksi ratkaisuksi olisi erittäin haastavaa, koska verkkotunnusten yhtiösidonnaisuuksia on vaikea todistaa, kun taas tässä tutkimuksessa nojauduttiin Yahoo Finance -rajapinnasta löytyvään yhtiön kotisivuun. Tämän lisäksi ei ole mitään yhtä ja selkeää julkista lähdettä, josta kaikki pääverkkotunnukset voisi helposti hakea, vaan pääverkkolistauksen hakeminen pitää aina tehdä tapauskohtaisesti. Tämä haaste myös osoittaa, miksi tämänkaltaista työkalua ei löytynyt valmiina, vaan se piti kehittää tutkimustamme varten.

Työkalun pisin vaihe oli TLS-salausten tietojen noutaminen verkkosivuista, mikä tehtiin yksi sivu kerrallaan. Tätä vaihetta voisi parantaa ja tehdä verkkosivujen ajoja useampi rinnakkain, mikä mahdollistaisi suuremman tutkimusaineiston keräämisen samassa ajassa. Työkalun ajossa tuli myös virheitä, joihin voisi kohdentaa parempaa virheenhallintaa, jolla saisi työkalun ajossa tulevia virheitä vähennettyä.

Tässä tutkimuksessa toteutetun työkalun toiminnassa on hyvä huomioida, että konstruktiivinen tutkimusote ei oleta, että luotu konstruktio ratkaisisi kaikki mahdolliset variaatiot, vaan konstruktio luodaan sitä tarkoitusta varten, johon sitä tarvitaan. Tästä syystä työkalu toteutettiin täyttämään tämän tutkimuksen tarpeet, joihin se todistetusti toimi. Lukka (2001) mainitseekin, että innovoidun konstruktion laajentaminen normaalisti tarkoittaa uusien markkinatestien tekemistä ja niiden analysointia. Työkalun laajentamisessa olisi siis hyvä tarkemmin perehtyä tulevaisuuden tarpeisiin ja uusiin kehityskohtiin, jos sitä haluttaisiin kehittää. Tutkimuksen valmistumisen jälkeen työkalu julkaistaan, jotta kuka tahansa voi vahvistaa työkalun toiminnan ja tutkimuksen tulokset sekä kehittää ja hyödyntää työkalua omassa tutkimustyössään.

Seuraavassa luvussa käsitellään tutkimuksen päätavoite sekä siihen liittyvät tutkimuskysymykset.

6.2 Päätavoite

Tutkimuksen päätavoitteena oli selvittää, mikä on Helsingin pörssiyhtiöiden verkkosivujen salauksen taso. Vastataksemme tähän kysymykseen vaadittiin vastauksia päätavoitteeseen liittyvistä tutkimuskysymyksistä, joihin jokaiseen on erikseen vastattu omissa alaluvuissaan.

6.2.1 Miksi turvallisen TLS-salauksen käyttäminen on tärkeää?

TLS-salauksen merkittävyyttä käytiin läpi luvussa 3, jossa salatun verkkoliikenteen hyödyistä nostettiin esille kolme keskeisintä kohtaa, jotka olivat salakuuntelemisen estäminen, verkkoliikenteen eheyden varmistaminen ja verkkosivun identiteetin todentaminen. Riittämätön salaus asettaa verkkosivua käyttävät käyttäjät alttiiksi näille kolmelle keskeiselle kohdalle, joiden vuoksi TLS-salaus on luotu turvaamaan käyttäjiä.

Turvallinen TLS-salauksen käyttö lähtee liikkeelle verkkosivujen ylläpitäjistä, jotka voivat vaikuttaa verkkosivujen salauksen vahvuuteen. Tämä tarkoittaa sitä, että yrityksen on oltava aktiivinen osapuoli ja kovennettava verkkosivut vastaamaan nykypäivän vaatimuksia. Riittämätön salaus saattaa altistaa käyttäjät vaaraan ja aiheuttaa arkaluontoisen tiedon vuotamista, jolla voi olla yritykselle suoria liiketaloudellisia haittoja.

Aikaisemmassa tutkimuksessa on tehty havainto, että verkkosivun käyttäjät aliarvioivat salauksen merkitystä, mutta samalla verkkosivujen ylläpitäjät ovat hyvinkin tietoisia salauksen tärkeydestä ja mitä hyökkäyksiä vasten sillä suojaudutaan (Krombholz ym., 2019, s. 246). Tämän tiedon mukaan ylläpitäjien tulisi keskimääräisesti olla tietoisia siitä, miksi TLS-salauksen käyttäminen on tärkeää, ja syyt sen mahdolliselle laiminlyömiselle on johduttava jostain muusta kuin tietämättömyydestä.

Turvallinen TLS-salauksen käyttö ei ole pelkästään sitä, että salaus on asetettu päälle, vaan asianmukaiseen salaukseen kuuluu myös pitää huoli riittävästä salauksesta. Ajan myötä vanhoista salausprotokollista saattaa löytyä haavoittuvuuksia, jotka otetaan huomioon myöhemmässä kehityksessä. SSL/TLS-salauksesta löydettyjä heikkouksia läpikäytiin teoriaosuudessa (LUKU 3.5). Ensiksi esiteltiin mies välissä -hyökkäys, jossa hyökkääjä aktiivisena osapuolena pyrkii pääsemään selaimen ja WWW-palvelimen väliin, jolloin verkkoliikenne kulkee hänen kauttansa, ja hyökkääjä pääsee salakuuntelemaan ja myös muokkaamaan verkkoliikennettä haluamallaan tavalla. Lisäksi tarkasteltiin erilaisia heikkouksia kuvastamaan, minkä tyyppisiä heikkouksia SSL/TLS-protokollista on löytynyt vuosien varrelta. Nämä heikkoudet olivat POODLE, SLOTH, FREAK, LogJam ja CRIME.

TLS-salauksen suositukset elävät jatkuvasti. Tämä haaste on nostettu esille myös aikaisemmassa tutkimuksessa, jossa testiryhmään valitut opiskelijat olivat kritisoineet parhaiden käytänteiden puutetta ja valtavaa tarvetta taustatiedoille ennen salauksen käyttöönottamista (Krombholz ym., 2017, s. 1347). Tähän haas-

teeseen on tuotu ratkaisuja tässä tutkimuksessa, jonka teoriaosuudessa on perehdytty salauksen toimintaperiaatteisiin ja TLS-salauksen tämänhetkisiin suosituksiin.

Julkisen avaimen varmenteet ja siihen liittyvä infrastruktuuri ovat myös keskeisiä osia toimivaa salausta, jossa varmenteet mahdollistavat sivuston identiteetin todentamisen. Ilman varmenteita verkkosivu ei pystyisi todistamaan omaa identiteettiään, jolloin käyttäjä ei voisi olla varma, mitä sivustoa hän todellisuudessa käyttää. Varmenteista ja sen käytännön toiminnasta kerrottiin tarkemmin teoriassa (LUKU 3.3).

Edellä mainittujen asioiden perusteella saimme selville, että TLS-salauksella on tärkeä tehtävä suojata käyttäjiä Internetissä ja että turvallista TLS-salausta pitäisi käyttää kaikissa verkkosivustoissa. Seuraavassa pohdinnan alaluvussa käsitellään vaadittavaa TLS-salauksen vähimmäistasoa, joka auttaa hahmottamaan, mitkä WWW-palvelin konfiguraatiot tarvitaan, jotta sen salaus kattaisi vähimmäisvaatimukset.

6.2.2 Mikä on TLS-salauksen suositeltu vähimmäistaso?

TLS-salaus koostuu useasta eri asetuksesta, joiden suositukset elävät jatkuvasti. Suosituksilla on painetta aika ajoin vain kiristyä, koska tietokoneiden laskentateho kasvaa ja vanhoista algoritmeista voi löytyä haavoittuvuuksia. Luvussa 3.8.8 avattiin, miten salausalgoritmien voimakkuutta mitataan biteissä, ja luvussa 3.7 tuotiin esille, millaisia haavoittuvuuksia SSL/TLS-protokollasta on menneisyydessä löytynyt. Tutkimuksessa todettiin, että vaikka verkkosivu käyttää SSL/TLS-salausta, voi suojaus silti olla heikko tai helposti murrettavissa.

Tutkimuksen TLS-salauksen vähimmäistaso muodostettiin konsensusmenettelyllä, johon valittiin mukaan neljä erityyppistä toimijaa, joiden luomien suositusten pohjalta muodostettiin salauksen vähimmäistaso. Toimijat olivat Hollannin Kyberturvallisuuskeskus, NIST, SSL Labs ja Mozilla-säätiö. Suositusten keräämisen haasteena oli, että toimijat antoivat suositukset vaihtelevissa muodoissa. Tämän takia tutkimuksessa jouduttiin ottamaan kantaa eri sanamuotoihin, jotta saatiin erotettua riittävä ja riittämätön sekä löydettyä salauksen vähimmäistaso.

Helsingin pörssiyhtiöiden verkkosivuista kerättiin tietoa liittyen kymmeneen eri suosituskategoriaan. TLS-salauksen suosituskategoriat olivat SSL/TLS-versio, salaussarjat, TLS-pakkaus, 0-RTT, OCSP stapling, varmenteen avaimen koko, varmenteen voimassaolo, varmenteen käyttöiän pituus, varmenteen yleinen nimi ja HSTS. Lista hyväksytyistä salaussarjoista löytyy liitteistä (LIITE 2), ja loput parametrit on koottu taulukkoon (TAULUKKO 5) luvussa 3.6.10.

Tässä tutkimuksessa kuitenkin havaittiin, että tuoreet suositukset olivat hyvinkin yhdenmukaisia, ja suurimmat erot ilmenivät käytännössä salaussarjoissa. Jokainen toimija ei kuitenkaan ottanut kantaa kaikkiin etsittyihin kymmeneen suosituskategoriaan, joten useamman toimijan käyttäminen on välttämätöntä. Tässä tutkimuksessa vedettiin yhteen kerätyt suositukset, jotka olivat arvokasta tietoa jo itsessään (LUKU 3.6.).

Verkkosivut voivat asettaa itselleen erilaisia vaatimuksia turvallisuuden suhteen. Ne verkkosivut, jotka kokevat tarvitsevansa korkeaa turvallisuutta, eivät välttämättä hyväksy turvallisuustasoa, joka on juuri ja juuri riittävä. Tästä syystä TLS-salauksen riittävä taso on vaikeaa määritellä, mutta sen vähimmäistaso sen sijaan pystytään kertomaan. Vähimmäistasolla tarkoitetaan tilannetta, jossa vähimmäistason alapuolelle jäävät asetukset eivät ole hyväksyttäviä millään verkkosivuilla riippumatta siitä, onko kyseessä korkeaa turvallisuutta vaativa verkkosivu tai ei. Kaikkia verkkosivuja koskeva vähimmäisvaatimustaso voidaan tällöin aina määritellä, mikä tehtiinkin tässä tutkimuksessa.

6.2.3 Kuinka moni Helsingin pörssiyhtiöiden verkkosivu läpäisi asetetut TLS-salauksen suositukset?

Työkalun ajamisen aikana Helsingin pörssiyhtiössä oli yhteensä 133 eri yhtiötä. Näiden yhtiöiden pääverkkotunnuksia vasten löytyi 7946 verkkotunnusta, joista 4782 kuunteli HTTPS-oletusporttia 443. Työkalun ajon aikana 351:ssa eri verkkotunnuksessa ilmeni virheitä, jotka kaikki poistettiin tuloksista, jolloin lopulliset testitulokset kerättiin 4431 verkkotunnuksesta.

Kaikki vähimmäissuositukset täytti ainoastaan 155 verkkosivua, joka oli 3,5 prosenttia kaikista testatuista verkkosivuista. Heikkoa tulosta voi osakseen selittää se, että joukossa on saattanut olla paljon vanhoja tai epäaktiivisesti ylläpidettyjä verkkosivuja. Osasyynä voi myös olla WWW-palvelimien ylläpitäjien osaamattomuus salauksen suhteen. Heikkoa tulosta voi myös selittää se, että monelle WWW-palvelimien ylläpitäjille riittää, että palvelu vain ylipäätänsä toimii salatusti, eivätkä he tunnista eroa vahvan ja heikon salauksen välillä. Tämäkin voi liittyä ylläpitäjien osaamattomuuteen, tai se voi olla myös piittaamattomuutta.

WWW-palvelimien TLS-salauksen oletusasetukset eivät aina tarjoa riittävää turvallisuustasoa, jolloin järjestelmän ylläpitäjän tulisi osata tehdä tarvittavat kovennukset asetuksiin. Oletusasetukset ovat myös riippuvaisia WWW-palvelimen ohjelmistoversiosta, ja kehittäjät oletettavasti koventavat oletusasetuksia ajan saatossa, jolloin ajantasaiset ohjelmistoversiot nousevat myös tärkeään rooliin.

Tutkimuksen teoriassa (LUKU 3.2.3) nostettiin esille aikaisempi tutkimus (Kotzias ym., 2018), jossa oli tehty havainto sovellusten taaksepäin yhteensopivuuteen liittyen. Verkkoselainten ylläpitäjillä on kannustin tukea vanhoja salauksia, jotta he eivät riko käyttäjien vanhoja verkkosivuja. Vanhat verkkoselaimet johtavat siihen, että WWW-palvelimen ylläpitäjät eivät välttämättä uskalla päivittää palvelujaan, jotta he eivät estäisi vanhojen verkkosivujen saatavuutta. Tuki taaksepäin yhteensopivuudelle voi siis olla osaltaan syypää siihen, miksi verkkosivut eivät läpäisseet vähimmäisvaatimuksia.

Tutkimuksen tuloksia vertailtiin myös SSL Labsin tilastoihin, jotka nostettiin esille SSL/TLS-salauksen suosituksissa (LUKU 3.6). SSL Labsin tilastot pitivät sisällään maailman suosituimpien verkkosivujen SSL/TLS-salauksen tason, joten näitä pystyttiin soveltuvin osin vertailemaan Helsingin pörssiyhtiöistä

saatuihin tuloksiin (TAULUKKO 25). Vertailusta käy ilmi, että saadut tulokset ovat isoilta linjoiltaan hyvinkin samankaltaiset.

TAULUKKO 25 SSL Labs verrattuna Helsingin pörssiyhtiöihin

Julkaisija	SSL Labsin tilastossa suositukset alittaneet verkkosivut	Suositukset alittaneet Helsingin pörssiyhtiöiden verkkosivut	Ero %-yks.
SSL/TLS versiot	48,1 %	45,9 %	2,2
TLS-pakkaus	0,1 %	0,1 %	0,0
0-RTT	0,4 %	2,4 %	-2,0
OCSP stapling	57,9 %	70,0 %	-12,1
Varmenteen koko	0,0 %	0,1 %	-0,1
HSTS	70,1 %	66,8 %	3,3

6.2.4 Mikä on yleisin syy sille, että verkkosivut eivät läpäisseet asetettuja TLS-salauksen suosituksia?

Yleisimmät syyt sille, että verkkosivut eivät läpäisseet asetettuja TLS-salauksen suosituksia olivat salaussarjat (73,3 %), OCSP stapling (70,0 %) ja HSTS (66,8 %). Nämä kolme kategoriaa erottuivat selkeästi muista kategorioista, koska muissa kategorioissa yli puolet verkkosivuista läpäisi suositukset. Tuloksissa on kuitenkin huomioitava, että TLS-salauksen asetukset eivät ole tasavertaisia keskenään, eli yhdellä suosituskategorialla voi olla enemmän merkitystä turvallisuuden kannalta kuin toisella. Nämä kaikki kategoriat käsiteltiin kuitenkin yhtä arvokkaina, koska suositusten laatijatkaan eivät erikseen olleet priorisoineet näitä kategorioita.

Edellisessä luvussa (LUKU 6.2.3) mainitut juurisyyt sille, miksi pörssiyhtiöiden verkkosivut eivät yleisesti läpäisseet TLS-salauksen suosituksia selittävät myös sen, miksi salaussarjat, OCSP stapling ja HSTS asetukset eivät olleet suosituksien mukaiset. Kerätystä datasta löytyi yhteensä 60 erilaista TLS 1.2 -protokollaversion salaussarjaa, joista ainoastaan 25 oli suositusten mukaisia. Tämä tarkoittaa sitä, että yli puolet verkkosivujen käyttämistä salaussarjoista ei yltänyt asetettuihin suosituksiin. Salaussarjoja on paljon, joten verkkosivujen ylläpitäjille jää vastuu niiden koventamisesta. Lisäksi OCSP staplingin ja HSTS:n kohdalla voitiin jo luvussa 3.6 todeta, että suosituimpien WWW-palvelimien oletusasetukset eivät olleet riittävät. Salaussarjojen kanssa samaa asiayhteyttä oletusasetuksiin ei pystytty osoittamaan, koska tässä tutkimuksessa ei tehty vertailua, kuinka WWW-palvelimien salaussarjojen oletusasetukset vertautuvat suositusten kanssa. Tulokset kuitenkin puhuvat sen puolesta, että WWW-palvelimien oletusasetukset eivät olisi riittävät. SSL Labs (2021) vahvistaa käsitystä siitä, että ongelma ei ole ainoastaan Helsingin pörssiin kuuluvien yhtiöiden verkkosivuissa, sillä OCSP stapling ja HSTS ovat samalla tasolla maailmanlaajuisesti eri verkkosivuilla. Myös salaussarjat näyttävät heikoilta SSL Labs:n oman arviointiasteikon mukaisesti. (SSL Labs, 2021.)

Aikaisemmassa tutkimuksessa on tehty havainto, että TLS-salauksen käyttöönotto WWW-palvelimessa on hankalaa jopa asiantunteville tekijöille (Krombholz ym., 2017, s. 1347). Tutkimuksessa nostettiin esille tämän johtuvan muun muassa parhaiden käytänteiden puutteesta, WWW-palvelimien heikoista oletusasetuksista, konfiguraatiotiedostojen sekavasta rakenteesta ja monimutkaisesta konfiguraatioprosessista sekä siitä, että käyttöönotto vaatii liian paljon taustatietoa. Nämä löydöt selittävät sitä, miksi kaikki suosituskategoriat eivät olleet asianmukaisesti käyttöön otettuja.

Salaussarjoja on paljon erilaisia, ja suositusten antajatkaan eivät olleet täysin yksinmielisiä siitä, mitkä salaussarjat täyttäisivät minimivaatimuksen. Salaussarjat ovat myös suhteellisen monimutkaisia ja ne vaativat perehtymistä, jotta niihin osaa ottaa kantaa. Salaussarjat ovat suositusten kategorioista hankalasti ymmärrettävin, ja niiden käytössä on eniten variaatiota, mikä voi osaltaan vaikuttaa siihen, että ylläpitäjät saattavat helposti jättää ne huomiotta.

6.2.5 Mikä on Helsingin pörssiyhtiöiden salauksen taso?

Päätutkimuskysymyksenämme oli selvittää, mikä on Helsingin pörssiyhtiöiden verkkosivujen salauksen taso. Edellisissä alaluvuissa käytyjen tulosten pohjalta voidaan todeta, että Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksessa olisi parannettavaa, koska 4431 toimivasta verkkosivusta vain 155 (3,5 %) verkkosivua täytti kaikki asetetut suositukset. Helsingin pörssiyhtiöistä saatuja tuloksia vertailtiin soveltuvin osin SSL Labsista (2021) saatujen tilastojen kanssa (LUKU 6.2.3) osoittamaan, miten Helsingin pörssiyhtiöt vertautuvat maailmanlaajuisesti eri verkkosivuihin. Tämän havainnon mukaan Helsingin pörssiyhtiöiden verkkosivujen salauksen taso on linjassa muiden verkkosivujen kanssa. Heikko tulos ei siis kohdistu pelkästään Helsingin pörssiyhtiöiden verkkosivuihin, vaan ongelma on maailmanlaajuinen.

Salauksen heikolla tasolla viitataan siihen, että suurin osa tutkituista sivuista eivät yltäneet salauksessaan suositellulle tasolle, eivätkä näin ollen olleet riittävän hyvin suojattuja. Huonosti suojattu verkkosivu altistaa sen käyttäjät niille hyökkäyksille, joita varten TLS-salaus on luotu käyttäjiä suojelemaan. Asianmukaisen TLS-salauksen käyttöönotto on yrityksen vastuulla, ja huonosti suojatulla verkkosivulla voi olla suoria liiketaloudellisia haittoja yritykselle. Ainoastaan riittävällä tasolla suojattu verkkosivu tarjoaa TLS-salauksen tuoman suojan.

Heikon TLS-salauksen syyt johtuvat useasta eri seikasta. Järjestelmien ylläpitäjät ovat tietoisia salauksen merkityksestä ja tärkeydestä, mutta salauksen käyttöönotto vaatii silti syvällisesti asiaan perehtymistä ja ammattitaitoa, jota voi olla mahdotonta vaatia jokaisen verkkosivun ylläpitäjältä (Krombholz ym., 2017, s. 1347, 2019, s. 246). Salauksen heikkoon tasoon pitäisi puuttua keskittymällä syihin, jotka ovat johtaneet heikkoon tulokseen. Tässä tutkimuksessa edesautettiin salaukseen liittyvien ongelmien ratkaisua usealla eri tavalla. Ensinnäkin loimme konsensuksen tämän hetken tuoreimmista TLS-salauksen suosituksista usean eri toimijan pohjalta, mistä on hyötyä eri tahoille TLS-salausta

käyttöön ottaessaan. Toisekseen tutkimuksessa tuotiin esille TLS-salauksen tärkeys kaikissa tilanteissa, eikä vain niissä tapauksissa, kun verkkosivulla käsitellään arkaluonteista tietoa. Lisäksi tutkimuksenne tuotti tietoutta TLS-salauksen käytännön toimivuudesta, mikä voi parantaa järjestelmien ylläpitäjien tietotaitoa ja täten vahvistaa verkkosivujen salauksen tasoa.

Keskeisin havainto tutkimuksessa oli, että Helsingin pörssistä vain 3,5 % verkkosivuista läpäisi asetetut TLS-salauksen suositukset. Suosituksista luotiin neljän eri toimijan pohjalta konsensus, joka antaa hyvän pohjan hyväksytystä vähimmäistasosta.

Seuraavaksi käsitellään tutkimuksen toistettavuutta ja luotettavuutta, joita on tärkeää arvioida tutkimuksen tieteellisyyden kannalta.

6.3 Tutkimuksen reliabiliteetti ja validiteetti

Tässä luvussa käsitellään tutkimuksen toistettavuutta (reliabiliteetti) ja pätevyyttä (validiteetti). Tutkimuksen toistettavuutta arvioidaan sillä, miten hyvin tutkimuksen mittaustulokset pystytään toistamaan. Pätevyydellä puolestaan tarkoitetaan tutkimuksen luotettavuutta ja tutkimuksessa käytettyjen mittareiden kykyä mitata, mitä niillä halutaan mitata. (Hirsjärvi ym., 2016, s. 231.)

Tutkimuksen aineisto on kerätty tutkimusta varten rakennetulla työkalulla, joka löytyy tutkimuksen liitteistä (LIITE 4; LIITE 5; LIITE 6). Suorittamalla uudestaan työkalun suoritusketju on mahdollista toistaa tutkimusaineiston kerääminen. Lopuksi tutkimustulosten keräämiseen käytettiin hyväksi tietokantaa vasten suoritettavia tietokantakyselyjä, jotka löytyvät liitteistä (LIITE 8 – LIITE 18).

Tutkimusta toistettaessa on huomioitava, että verkkosivut muuttuvat jatkuvasti ajan kuluessa. Verkkosivujen ylläpitäjät voivat tehdä verkkosivuihinsa päivityksiä ja sitä mukaa myös parantaa tai huonontaa verkkosivujen TLS-salauksen tasoa. Yritykset julkaisevat uusia, poistavat vanhoja ja tekevät muutoksia olemassa oleviin verkkosivuihin, jolloin tulokset eivät pysy muuttumattomina. Muutoksia ilmenee myös Helsingin pörssissä, johon ajan myötä liittyy lisää yhtiöitä tai josta poistuu yhtiöitä. Tutkimuksessa määriteltiin, mikä on hyväksytty TLS-salauksen vähimmäistaso, eli koottiin TLS-salauksen suositukset. Nämä laaditut suositukset ovat TLS-salauksen suositukset tekohetkellä, mutta ajan saatossa teknologia kehittyy ja tieto lisääntyy, minkä myötä suositukset muuttuvat. Mittaustapa on silti vakioitu ja sillä pystytään hakemaan tietoa verkkosivujen TLS-salauksesta ja vertailemaan tehtyjen tietokantakyselyiden (LIITE 8 – LIITE 18) avulla tuloksia TLS-salauksen suosituksiin.

Tutkimuksessa on jaettu työkalu, jolla aineisto voidaan kerätä ohjelmallisesti, ja kaikki tietokantakyselyt, jotka tehdään haluttuun aineistoon. Tutkimuksesta löytyy myös TLS-salauksen suositukset, joita vasten kerättyä aineistoa voidaan verrata. Käytännössä siis tutkimus on mahdollista toistaa identtisesti, mutta yllä annettujen perusteluiden takia on erittäin epätodennäköistä, että saadut tulokset olisivat täysin yhteneväisiä tässä tutkimuksessa saatujen tulos-

ten kanssa. Tämä toki on ymmärrettävää tämänkaltaisessa tutkimuksessa, jossa tutkittavat kohteet muuttuvat ja teknologia kehittyy jatkuvasti.

Työkalussa käytetään muutamia eri ohjelmistoja, jotka voivat asettaa rajoituksia myöhemmin työkalun käyttöön ja tutkimuksen toistettavuuteen. Jos käytettyjen ohjelmistojen tuki lakkaa ja tekniikka kehittyy, eivät osa työkalun vaiheista välttämättä toimi oikein. Työkalussa myös käytetään muutamia julkisia lähteitä, jotka saattavat muuttua tai lopettaa toimintansa. Nämä seikat voisivat olla mahdollista korvata tulevaisuudessa muilla ohjelmistoilla ja julkisilla lähteillä, mutta nämä työkalua ja sen käyttöä koskevat rajoitteet on hyvä tiedostaa.

Tutkimuksen luotettavuutta parantaa tutkimuksen toteuttamisesta kertova tarkka selostus, joka koskee tutkimuksen kaikkia vaiheita (Hirsjärvi ym., 2016, s. 232). Tutkimuksessa on kiinnitetty tähän erityishuomiota, ja jokainen vaihe kuvailtu täsmällisesti, jotta tutkimuksen tulos ja siitä tehdyt johtopäätökset olisivat mahdollisimman päteviä.

Tutkimus on pohjustettu laajalla teoriataustalla, jonka tarkoituksena oli liittää tutkimus olemassa olevaan teoriapohjaan ja auttaa lukijaa ymmärtämään tutkimuksen sisältö. Lisäksi tällä osoitimme asiantuntijuutemme aihealueesta. Teoriassa myös käsiteltiin aikaisempia tutkimuksia ja kerrottiin perustellusti, miten TLS-salauksen suositukset luotiin. Lisäksi selostimme käytetyn tutkimusmenetelmän teorian ja kuvailimme, miksi kyseinen tutkimusmenetelmä valittiin ja miten sitä sovellettiin tässä tutkimuksessa. Käytetty tutkimusmenetelmä ei suoraan istunut tämänkaltaiseen tutkimukseen, minkä takia sitä piti jonkin verran soveltaa.

Tutkimuksen luotettavuuteen liittyen konstruktion toteutus, testaus ja aineiston validointi dokumentoitiin oleellisin osin. Konstruktiosta on hyvä huomioida se, että se kehitettiin keräämään aineistoa tätä tutkimusta varten. Konstruktiolla voidaan kerätä aineistoa Helsingin pörssiyhtiöiden verkkosivuista, mutta jos konstruktiota halutaan hyödyntää muihin tarkoituksiin, vaatii se mahdollista lisäkehitystä.

Tutkimuksen aineiston keräämisestä ja käsittelystä kerrottiin oleellinen tieto, ja tuloksissa avattiin tarkemmalla tasolla kaikki suosituskategoriat, jotta tulokset olisivat mahdollisimman läpinäkyviä ja helposti tulkittavissa. Pohdinnassa on arvioitu tutkimusta laajasti ja perusteltu, miksi tiettyihin päätelmiin on päädytty. Tämän lisäksi pohdinnassa on sidottu tutkimuksen löydökset aiempiin tutkimuslöydöksiin ja teoriataustaan. Tutkimuksen liitteisiin on lisätty tietoa, joka ei tutkimuksen ymmärtämisen kannalta ollut oleellista sijoittaa itse tekstiin, mutta nähtiin oleellisena liittää osaksi raporttia tutkimuksen toistettavuuden ja pätevyyden arvioinnin kannalta. Nämä kaikki seikat täsmällisen dokumentoinnin kera tukevat tutkimuksen luotettavuutta ja toistettavuutta.

Aikaisemmissa tutkimuksissa on ollut erilaisia variaatioita, joilla voi mitata TLS-salauksen tasoa, kuten valitsemalla tutkittavaksi vain muutamia eri kategorioita (Alashwali ym., 2019; Weerasinghe & Disanayake, 2018), tai kokoomalla eri kategorioista yksi arvosana ja jättämällä yksityiskohdat pois (Ali & Murah, 2018; Strzelecki & Rizun, 2020). Tässä tutkimuksessa haluttiin mitata laajemmin ja tarkemmin TLS-salauksen tasoa käyttäen useampia eri suosituska-

tegorioita, jotta saatiin selville, mihin TLS-salauksen asetuksiin olisi syytä kiinnittää enemmän huomiota.

Tässä tutkimuksessa käytetyt mittarit, eli TLS-salauksen suositukset, toimivat vertailuna kerättyä aineistoa vasten. TLS-salauksen suositukset kerättiin konsensusmenetelmällä neljältä eri toimijalta, jotka olivat tehneet julkaisun TLS-salauksen suosituksista. Valitut toimijat edustivat erityyppisiä organisaatioita, ja kaikki valitut suositukset olivat vuodelta 2019 tai uudempia, jotta ne olivat tarpeeksi tuoreita.

Tutkimuksessa päädyttiin konsensusmenettelyyn, koska vähimmäisasetuksiin ei ole oikeita tai vääriä arvoja, ja koska tavoitteena oli löytää yhteisymmärrys TLS-salauksen asetuksista. Jos tutkimuksessa olisi valittu vain yksi toimija, olisi kyseisen toimijan mielipiteelle annettu suuri painoarvo, eikä organisaatioiden joukossa sitä paitsi ollut yhtäkään toimijaa, joka olisi perustellusti noussut ylitse muiden. Tämän vuoksi konsensusmenettely nähtiin sopivana tapana luoda mittari tähän tutkimukseen.

Tutkimukseen olisi myös voitu valita enemmän toimijoita, mutta tehdyn pohjatyön perusteella osa suosituksista oli liian vanhoja ja osa toimijoista lisäksi hyödynsi suosituksissaan jo valittujen toimijoiden suosituksia. Näistä syistä päädyttiin neljään eri toimijaan. Valitut toimijat olivat myös erityyppisiä, mikä toi konsensukseen laajempaa näkökulmaa eri toimialoilta. Luodut suositukset ovat paras näkemys tämän hetken vähimmäisvaatimuksista, mutta niitä ei voi pitää yleispätevänä kaikkiin verkkosivustoihin, koska suojauksen taso voi vaihdella eri verkkosivustojen välillä ja eri tahot voivat painottaa eri asetuksien tärkeyttä eri tavoilla.

Luodussa mittarissa oli kymmenen eri kategoriaa, joita vasten verkkosivujen TLS-salauksen tasoa verrattiin. Eri kategorioiden avulla saatiin selville, täyttääkö verkkosivu asetetuttujen kategorioiden vähimmäisvaatimukset vai ei. Tämä toi läpinäkyvyyttä tuloksiin, koska niistä nähtiin, missä TLS-salauksen asetuksissa olisi vielä parannettavaa. Luotu mittari mittasi niitä asioita, joita suosituksia julkaisseet organisaatiot näkivät tarpeellisina huomioida TLS-salauksessa, minkä vuoksi ne valittiin myös tutkimuksen mittariksi.

Tutkimuksen aineistoa ei kerätty satunnaisotannalla erilaisista suomalaisista yhtiöistä, vaan valitut yhtiöt olivat kaikki Helsingin pörssin yhtiöt (133 kpl). Helsingin pörssiyhtiöt jakaantuivat suhteellisen tasaisesti eri kokoisiin ja eri toimialoja edustaviin yhtiöihin (LIITE 1). Tuloksia ei voida luotettavasti yleistää kaikkiin suomalaisten yhtiöiden verkkosivujen TLS-salauksiin, koska Suomeen on rekisteröity yhteensä yli 600 tuhatta suomalaista yhtiötä (Patentti- ja rekisterihallitus, 2022), ja valitut yritykset olivat vain pieni otanta siitä (noin 0,02 %).

Vaikkakin tutkimuskohteiksi valittiin kaikki Helsingin pörssiyhtiöt, on mahdoton sanoa, mikä osuus kyseisten yhtiöiden aliverkkotunnuksista löytyi tutkimusta varten. Tämä johtuu siitä, että Helsingin pörssiyhtiöiden verkkotunnuksiin liitettyjä aliverkkotunnuksia ei löydy luotettavasti mistään julkisesti. Pääverkkotunnuksiin liitettyjen aliverkkotunnusten suuri määrä (noin 33 kpl verkkosivua yhtä pääverkkotunnusta kohden) voisi viitata siihen, että iso osa

näistä julkisista verkkosivuista löytyi. Tätä tietoa ei pystytty tässä tutkimuksessa kuitenkaan vahvistamaan.

Tutkimuksessa tehtiin oletus, että Yahoo Finance -rajapinnasta saatu yhtiön pääverkkotunnus ja siihen liitetyt aliverkkotunnukset ovat yhtiölle kuuluvia. Kaikki yhtiöiden kotisivut ja pääverkkotunnukset tarkistettiin valideiksi, mutta täyttä varmuutta ei ollut mahdollista saada, koska verkkotunnuksien omistajista ei ollut saatavilla julkista tietoa luotettavasti. Kaikki löydetyt verkkosivut vahvistettiin satunnaisotannalla, jonka perusteella pääteltiin, että tutkittavat verkkosivut olivat valideja. Tämä ei siltikään poista sitä, etteikö joukossa voisi olla verkkosivuja, jotka eivät kuuluneet valituille yhtiölle. Jokaiselta yhtiöltä valittiin vain yksi pääverkkotunnus, vaikka yhtiöllä saattoi olla käytössä useampia pääverkkotunnuksia. Tämä rajoitus tehtiin sen takia, koska ei löytynyt luotettavaa tapaa hakea yhtiön kaikkia käytössä olevia pääverkkotunnuksia.

Tutkimuksessa tehtiin myös oletus, että yhtiöiden salatut julkiset verkkosivut toimivat vain portissa 443, koska se on HTTPS-yhteyden oletusportti, vaikka verkkosivun voisi vaihtaa toiseen porttiin. Tämä oletus tehtiin siksi, että kaikkien mahdollisten muiden porttien tarkastaminen WWW-palvelun varalta olisi pidentänyt ajoaikaa kohtuuttoman paljon. Muiden porttien skannaamista olisi voitu pitää myös laittomana. Koska todellista määrää yhtiön julkisista verkkosivuista ei ole tiedossa, ei myöskään pystytä luotettavasti yleistämään tutkimustuloksia Helsingin pörssiyhtiöiden verkkosivujen TLS-salaukseen liittyen. Huomioitavaa silti on, että tässä tutkimuksessa saadut tulokset (TAULUKKO 12) vaikuttaisivat noudattelevan SSL Labsin keräämän aineiston (TAULUKKO 6) kanssa. Tämä viittaisi siihen, että tutkimuksesta saadut tulokset olisivat päteviä.

6.4 Tutkimuksen kontribuutio

Lopuksi vielä pohdimme tulostemme pohjalta tutkimuksemme käytännön ja tieteellistä kontribuutiota seuraavissa kahdessa alaluvussa. Näiden huomioiminen on konstruktiivisen tutkimusotteen kuudes ja seitsemäs vaihe (Lukka, 2000, s. 119).

6.4.1 Käytännön kontribuutio

Tutkimus tuotti kokonaisuudessaan kolme konkreettista käytännön kontribuutiota, jotka ovat työkalu, TLS-salauksen suositukset sekä TLS-salauksen eri tasot. Tässä luvussa käsitellään nämä mainitut käytännön kontribuutiot tarkemmin.

Ensimmäisenä kontribuutiona syntyi työkalu, jolla pystytään massatestaamaan TLS-salauksen taso isosta määrästä verkkotunnuksia. Tutkimuksen jälkeen työkalu julkaistaan julkiseen käyttöön, jolloin se on minkä tahansa tahon, kuten muiden tutkijoiden tai yritysten, vapaassa käytössä. Tämä mahdollistaa esimerkiksi tutkimuksen toistamisen tai sen, että työkalua voidaan kehit-

tää uusiin tutkimuskohteisiin, kuten muiden verkkosivujen TLS-salauksen mittaamiseen, jolla voidaan luoda lisää käytännön ja tieteen kontribuutiota tulevaisuudessa.

Tutkimuksessa syntyi ajantasaista tietoa TLS-salauksen suosituksista. Tällä hetkellä verkkosivujen ylläpitäjien voi olla haastavaa löytää ajantasaista tietoa verkkosivujen TLS-salauksen vähimmäistasosta. Tämä juontaa juurensa siitä, että suositusten antajia on useita, suositukset voivat olla vanhentuneita, osa suosituksista ottaa kantaa vain muutamiin eri asetuksiin tai suosituksien tulkitseminen voi olla muuten vain haastavaa. Tässä tutkimuksessa luotiin TLS-salauksen suositukset, joita verkkosivujen ylläpitäjät voivat pitää ohjenuorana, kun he käyttöönottavat omien verkkosivujen TLS-salausta. Tutkimuksessa nostettiin esille tärkeimmät TLS-salauksen asetukset ja kerrottiin, mitä ne tarkoittavat ja missä asetuksessa ne olisi syytä olla, jotta TLS-salaus täyttäisi vähimmäisvaatimukset. Tuloksista nousi myös esille kolme eri suosituskategoriaa (salaussarjat, OCSP stapling ja HSTS), jotka kaikista yleisimmin alittivat suositukset ja joihin verkkosivujen ylläpitäjien olisi syytä kiinnittää huomiota, jos he haluavat täyttää asetetun vähimmäistason.

Tutkimuksessa tuotiin esille TLS-salauksen eri tasoja, missä on ymmärrettävä ero heikon ja vahvan salauksen välillä. Ei siis pelkästään riitä, että verkkosivu on salattu, vaan aina on pidettävä huoli hyvästä salauksesta. Heikko salaus mahdollistaa salauksen purkamisen. Tutkimuksessa käytiin läpi mahdollisia riskejä, jotka käyttäjien ja ylläpitäjien on hyvä tiedostaa, jos salausta ei ole tai se on toteutettu heikosti. Samalla tutkimuksessa myös korostettiin sitä, että vähimmäistaso ei välttämättä riitä kaikkiin verkkosivuihin, vaan verkkosivujen ylläpitäjien olisi hyvä myös tiedostaa, minkälainen TLS-salauksen taso on heidän verkkosivuilleen riittävä.

6.4.2 Tieteellinen kontribuutio

TLS-salausta voidaan pitää yhtenä verkkosivujen turvallisuuden kulmakivistä, joten kaikki tutkimukset TLS-salaukseen liittyen tuovat tärkeää lisätietoa siihen liittyen. Tämä tiedon lisääntymisen on toivottava tulevaisuudessa edesauttavan turvallisemman verkkoliikenteen ja täten turvallisemman verkkosivujen käytön toteuttamista. Tässä tutkimuksessa saatiin uutta tieteellistä kontribuutiota esimerkiksi aineiston keräämisestä, aineiston käsittelemisestä, tutkimustuloksista sekä tutkimusmenetelmästä, joista kerrotaan tässä luvussa.

Tämän tutkimuksen yhtenä tieteellisenä kontribuutiona on tieto, millä tasolla tutkittujen Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen taso on. Tutkimuksessa ainoastaan 3,5 prosenttia verkkosivustoista täytti asetetut suositukset, mitä voidaan pitää huonona tuloksena. Tällä tutkimuksella saatiin läpileikkaus useasta eri kokoisesta ja eri toimialalla toimivasta suomalaisesta yhtiöstä (LIITE 1), joita ei aikaisemmin olla tutkittu. Tämä tieto TLS-salauksen huonosta tasosta lisää ymmärrystä siitä, että TLS-salaukseen ja sen turvallisuuteen ei ole keskitytty riittävästi. Verkkosivujen TLS-salauksen tasoon olisi siis syytä kiinnittää enemmän huomioita.

Tutkimuksessa tuotiin myös esille TLS-salauksen tärkeys kaikissa tilanteissa, eikä vain niissä tapauksissa, kun verkkosivulla käsitellään arkaluonteista tietoa. Lisäksi tutkimuksessa avattiin kaikki suosituskategoriat, jotta saatiin tarkempi selvyys siitä, mitkä kategoriat jäävät ylläpitäjiltä yleisimmin koventamatta. Nämä eri kategoriat pitäisi ottaa huomioon, jotta TLS-salauksen tasoa saataisiin kohennettua.

Tutkimuksessa luotiin TLS-salauksen tason mittaamiselle uusi tapa, jossa huomioitiin useampia eri kategorioita, joita suositusten antajat pitivät tärkeänä. Yhdestäkään aikaisemmasta löydetystä tutkimuksesta ei löytynyt tämänkaltaista mittaustapaa, jossa olisi hyödynnetty alan toimijoiden antamien suosituksia. Tämä tuo verkkosivujen TLS-salauksen testaamiseen uuden tavan, jota voidaan hyödyntää muissakin tutkimuksissa. Suositusten luomisen myötä ymmärrettiin tarkemmin, ettei TLS-salauksen tason mittaaminen ole täysin yksiselitteistä, vaan siinä on useita tekijöitä, jotka on hyvä huomioida.

Nämä samat tekijät voivat vaikuttaa myös siihen, miksi TLS-salauksen taso on näin heikko. Verkkosivustojen ylläpitäjien ei ole täysin yksiselitteistä pitää TLS-salauksen turvallisuutta riittävän korkealla, koska siihen vaikuttaa moni eri tekijä, kuten parhaiden käytänteiden puute, WWW-palvelimien heikot oletusasetukset, konfiguraatiotiedostojen sekava rakenne, monimutkainen konfiguraatioprosessi ja se, että salauksen käyttöönotto vaatii liian paljon taustatietoa. Näiden lisäksi suositukset muuttuvat jatkuvasti, mikä vaatii ylläpitäjiltä jatkuvaa intoa, valveutuneisuutta ja aktiivista työotetta pitää verkkosivujensa TLS-salaus riittävällä tasolla. Näiden haasteiden ymmärtäminen herättää samalla kysymyksen, miksi riittävän turvallisuuden toteuttaminen on niin haastavaa, jos kerran TLS-salaus on verkkoliikenteen turvallisuuden yksi kulmakivistä. Toivottavaa olisi, että tulevaisuudessa näihin haasteisiin löytyisi helpompia ratkaisuja, jotta verkkosivujen TLS-salauksen turvallisuus pysyisi jatkuvasti riittävällä tasolla.

Yhtenä käytännön kontribuutiona mainittiin tutkimuksessa kehitetty työkalu, jota voidaan käyttää uusien tutkimusten toteuttamiseen. Työkalun toteuttaminen toi käytännön kontribuution lisäksi myös tieteellistä kontribuutiota. Tutkimuksessa käytiin kohta kohdalta läpi työkalun kehittämisen, testauksen ja aineiston keräämisen vaiheet. Tämä työkalun kehityksestä kerätty tieto luo tieteellistä kontribuutiota siitä, minkälaisia kohtia tulevaisuudessa työkalun kehittämisessä tai uuden työkalun luomisessa olisi syytä huomioida. Nostimme tutkimuksessamme esille oletuksia ja ongelmakohtia, joita ratkoimme työkalun toteutuksessa, kuten eri organisaatioiden pääverkkotunnusten hakeminen luotettavasti sekä kaikkien pääverkkotunnukseen liitettyjen aliverkkotunnusten noutaminen. Työkalun kehittämisessä löydettiin myös hyviä ratkaisuja useampiin tilanteisiin, joita muut kehittäjät voisivat potentiaalisesti kohdata, kuten miten TLS-salauksesta kerätään tietoa, miten kerätty aineisto tallennetaan ja miten kerättyä aineistoa käsitellään.

Tutkimus toi myös tieteellistä kontribuutiota käytettyyn tutkimusmenetelmään. Konstruktiivinen tutkimusote ei ole kovinkaan paljon käytetty tutkimusmenetelmä, mikä on nähtävissä siinä, että sitä hyödyntäviä tutkimuksia

löytyy vielä suhteellisen vähän muihin tunnetumpiin tutkimusmenetelmiin verrattaessa. Tässä tutkimuksessa sovellettiin konstruktiivista tutkimusotetta liittyen muun muassa konstruktioon ja kohdeorganisaatioon. Yleisesti konstruktiivisessa tutkimusotteessa itse konstruktion toteuttaminen on tutkimuksen päätavoite, mutta tutkimuksemme alatavoitteena oli konstruktio, jolla kerättiin aineisto päätavoitetta varten. Tutkimuksessa myös jätettiin kohdeorganisaatio pois, jotta tutkimuksen epäonnistumisen riskiä saatiin pienennettyä. Tutkimus oli onnistunut ja se lisää tietoa kyseisestä tutkimusmenetelmästä sekä tuo ymmärrystä uusista mahdollisista käyttökohteista. Uudenlainen tapa hyödyntää tutkimusmenetelmää voi mahdollistaa tulevaisuudessa sellaisia käyttötarkoituksia ja tutkimuksia, joihin tutkimusmenetelmää ei olisi välttämättä edes harkittu käytettäväksi.

Seuraava luku sisältää tutkimuksen yhteenvedon, jossa kiteytetään koko tutkimus lyhyesti yhteen lukuun, sekä esitetään jatkotutkimusaiheet.

7 JOHTOPÄÄTÖKSET

Tässä luvussa esitämme tutkimuksemme ja sen tulokset yhteenvetona. Näiden lisäksi pohdimme valitun tutkimusmenetelmän soveltuvuutta tämänkaltaiseen tutkimukseen ja annamme ehdotuksia jatkotutkimusaiheista.

7.1 Tutkimuksen yhteenveto

Tässä tutkimuksessa tutkittiin Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen tasoa. Päätavoitteenamme oli tarkastella, onko Helsingin pörssiyhtiöiden verkkosivut salattu riittävän vahvalla TLS-salauksella. Päätavoitteen lisäksi tutkimuksen alatavoitteena oli luoda työkalu, jolla pystytään testaamaan suuren verkkosivumäärän TLS-salauksen tasoa. Molemmat tavoitteet saavutettiin.

Tutkimustulokset osoittivat, että Helsingin pörssiyhtiöiden verkkosivuista ainoastaan 3,5 % ylsi riittävän vahvalle tasolle TLS-salauksessa. Kun tutkimuksessa saatuja tuloksia vertailtiin SSL Labsin (2021) julkaisemiin tuloksiin, ne olivat hyvin samankaltaiset, eli voidaan sanoa, että Helsingin pörssiyhtiöiden verkkosivut eivät tee eroa maailman suosituimpien verkkosivujen kanssa.

Tutkimustulosten kolme heikointa TLS-salauksen asetusta olivat salaussarjat, OCSP stapling ja HSTS. Heikon tuloksen syitä ei erikseen tutkittu, mutta aikaisempaan tutkimustietoon nojaten syiden ajatellaan johtuvan muun muassa parhaiden käytänteiden puutteesta, WWW-palvelimien heikoista oletusasetuksista, konfiguraatiotiedostojen sekavasta rakenteesta, monimutkaisesta konfiguraatioprosessista ja käyttöönoton vaativuudesta (Krombholz ym., 2019, s. 246).

Tutkimuksen alatavoitteen saavuttamiseksi toteutettu työkalu keräsi tutkimusaineistoa kaikista löydetyistä Helsingin pörssiyhtiöiden verkkosivuista, joita oli yhteensä 4431 kappaletta. Työkalun lähdekoodit julkaistiin tutkimuksen liitteissä (LIITE 4 — LIITE 6) kaikkien saataville ja myös GitHubissa (https://github.com/kiuru/pyTLScanner), jotta kuka tahansa pystyisi hyödyntämään sitä jatkossa vastaavanlaisessa tarkoituksessa.

Tutkimusmenetelmäksi valittu konstruktiivinen tutkimusote keskittyi reaalimaailman ongelman tunnistamiseen ja sen ratkaisemiseksi kehitettävään konstruktioon. Tutkimusmenetelmä pystyttiin sovittamaan tutkielmaan, mutta sen käyttöön liittyi haasteita, minkä vuoksi menetelmää jouduttiin hieman soveltamaan tätä tutkimusta varten. Tutkimusmenetelmä ei ollut kovinkaan käytetty entuudestaan, mikä tuotti heti ensimmäiseksi haasteita löytää käyttötapauksia, miten tutkimusmenetelmää on sovellettu aikaisemmissa tutkimuksissa. Yleisesti konstruktiivisessa tutkimusotteessa konstruktion toteuttaminen on tutkimuksen päätavoite, mutta tämän tutkimuksen alatavoitteena toteutettiin konstruktio, jolla kerättiin tutkimusaineisto päätavoitetta varten. Lisäksi yksi tutkimusmenetelmän keskeisistä kohdista, eli kohdeorganisaation hankkiminen, jätettiin pois, mutta valinta oli perusteltu, eikä sitä nähty tässä tapauksessa relevantiksi.

Tämä tutkimus toi esille kolme käytännön kontribuutiota, jotka olivat työkalu TLS-salauksen testausta varten, konsensus eri toimijoiden TLS-salauksen suosituksista ja erot TLS-salauksen vahvuustasoissa. Tutkimus toi lisäksi tiedeyhteisölle kolme kontribuutiota, jotka ovat Helsingin pörssiyhtiöiden verkkosivujen TLS-salauksen tasot tutkimuksen tekohetkellä, syitä painottaa TLS-salauksen tärkeyttä kaikissa verkkosivuissa ja tapa mitata TLS-salauksen vahvuustasoja.

Tutkimukselle annetut tavoitteet saavutettiin. Alatavoitteena onnistuimme kehittämään työkalun, jolla voidaan testata TLS-salausta suurta joukkoa verkkosivuja vasten. Työkalun avulla saimme päätavoitteenamme selvitettyä, että Helsingin pörssiyhtiöiden verkkosivuista ainoastaan 3,5 % ylsi riittävän vahvalle tasolle TLS-salauksessa. Tutkimustulos oli hyvinkin lähellä SSL Labsin (2021) julkaisemien tilastojen kanssa, joten Helsingin pörssi ei tee suurta eroa siinä.

TLS-salaus on tärkeä osa Internetiä, sillä sitä käytetään joka kerta, kun käyttäjä avaa verkkosivun salatulla yhteydellä. Pelkkä yhteyden salaus ei silti yksinään riitä, vaan verkkosivujen ylläpitäjien pitäisi ymmärtää turvallisen TLS-salauksen kriteerit ja noudattaa niitä. Tutkimuksessamme huomasimmekin, että verkkosivujen ylläpitäjät eivät ole tehneet asianmukaisia toimia, minkä vuoksi verkkosivujen TLS-salaukset eivät ole riittävällä tasolla ja niissä olisi parannettavaa. TLS-salauksen suuren käyttäjämäärän ja tärkeyden vuoksi olisi suotavaa, että TLS-salauksen turvallisuuteen kiinnitettäisiin enemmän huomiota jokaisen Internetin käyttäjän verkkoturvallisuuden takaamiseksi.

Lopuksi käymme vielä läpi tutkimuksemme pohjalta syntyneet jatkotutkimusaiheet.

7.2 Jatkotutkimusaiheet

Tutkimuksemme myötä saimme tuotettua uutta näkökulmaa TLS-salaukseen liittyen. Aihepiiri tarjoaa paljon kiinnostavia kysymyksiä jatkotutkimuksiin, koska TLS-salausta ja sen merkitystä ymmärretään koko ajan paremmin. Teknologia kehittyy myös jatkuvasti, mikä kasvattaa TLS-salauksen vahvuuden tär-

keyttä, koska heikosti toteutettu TLS-salaus ei anna riittävää luotettavuutta. Tässä tutkimuksessa ei ollut mahdollista tarkastella kaikkea TLS-salaukseen liittyen, koska aihealue olisi ollut liian laaja, mutta tutkimus tarjoaa useita eri jatkotutkimusaiheita, joita käsitellään tässä luvussa. Jatkotutkimusaiheemme on mietitty sen perusteella, mitä kysymyksiä tutkimuksemme herätti ja miten tässä tutkimuksessa nousseita ajatuksia voitaisiin jatkokehittää.

Tätä tutkimusta voisi laajentaa muutamalla eri tavalla. Yhtenä vaihtoehtona olisi käyttää jo haettua dataa ja tutkia aineistosta, vaikuttaako TLS-salauksen tasoon yritysten koko, toimiala tai jokin muu vastaava yhtiöön liittyvä ominaisuus. Tutkimuksen kohteita voitaisiin myös lisätä, jotta saataisiin monipuolisempaa tietoa siitä, vaikuttaako TLS-salaukseen jokin näistä yhtiön ominaisuuteen liittyvistä asioita. Uusien organisaatioiden lisääminen tutkimukseen olisi suhteellisen helppoa tutkimuksessa kehitetyllä työkalulla. Toisena vaihtoehtona voisi olla ajallinen laajennus: työkalun avulla tutkimusaineisto on helposti uudelleen noudettavissa, joten aineistoa voitaisiin hakea tietyn väliajoin uudelleen ja katsoa, miten TLS-salauksen taso kehittyy ajan myötä.

Yhtenä jatkotutkimuskohteena olisi myös itse työkalu. Tutkimuksessa kehitettyä työkalua voisi kehittää esimerkiksi geneerisemmäksi ja virhesietoisemmaksi. Työkalun ensimmäinen vaihe kohdistui Helsingin pörssiyhtiöihin, mutta työkalua voitaisiin kehittää siihen suuntaan, että ensimmäinen vaihe olisi geneerisempi, millä mahdollistettaisiin muun muassa uusien kohteiden kerääminen. Työkalun ajossa noin kahdeksassa prosentissa tuli jokin virhe, kun verkkosivun TLS-salausta testattiin. Työkalun virhesietoisuutta voitaisiin kehittää eri tilanteissa, jotta työkalun ajossa tulleiden virheiden määrää saataisiin vähennettyä. Tässä kehityskohdassa pitää huomioida se, että osalle ajossa tulleista virheistä ei voi tehdä mitään, koska ne johtuivat tutkituista verkkosivustoista, mutta osa virheistä olisi todennäköisesti ratkaistavissa.

Tutkielmassa selvisi, että isossa osassa (96,5 %) tutkittujen verkkosivujen TLS-salauksessa olisi jotain parannettavaa. Jatkotutkimuksessa voisi selvittää, mitkä syyt johtavat heikosti toteutettuun TLS-salaukseen. Tämän jatkotutkimuksen voisi toteuttaa haastattelututkimuksena, jossa haastateltaisiin kohdeyhtiöiden WWW-palvelimien ylläpitäjiä, joilta voisi kysyä, mistä syistä osa heidän verkkosivunsa TLS-salaus on toteutettu heikosti. Löytämällä syitä heikosti toteutettuun salaukseen voisi mahdollisesti löytää ratkaisuja, jotka auttaisivat saamaan TLS-salauksen tason vahvemmaksi ja turvalliseksi.

Verkkosivuista 73,3 prosentissa oli jotakin parannettava salaussarjoissa, ja salaussarjat olivatkin kaikista yleisin syy, miksi verkkosivu ei täyttänyt suosituksia. Tutkimuksessa tehtiin muutamia oletuksia, mistä tämä voisi johtua, kuten suositusten puutteesta, salaussarjojen laajasta valikoimasta ja siitä, että ne ovat hankalasti ymmärrettävissä. Olisikin kiinnostavaa selvittää, miksi salaussarjat ovat yleisin syy sille, että verkkosivu ei täytä TLS-salauksen suosituksia.

Tutkimuksen pohdinnassa kiinnitettiin huomiota siihen, että osa TLS-salauksen heikosta tasosta voisi selittyä WWW-palvelimien heikoista oletusasetuksista. Viimeisenä jatkotutkimusehdotuksenamme olisikin tutkia WWW-palvelimien oletusasetuksia. Tarkasteluun voisi ottaa yleisimmät WWW-

palvelimet ja niiden eri ohjelmistoversiot ja selvittää, kuinka hyvin ne täyttävät asetetut TLS-salauksen suositukset. Toiseksi WWW-palvelimien oletusasetuksia tutkittaessa voisi selvittää, kuinka moni verkkosivu käyttää WWW-palvelimien oletusasetuksia. WWW-palvelimet saattavat kertoa HTTP-vastauksessa käytetyn WWW-palvelimen ohjelmiston nimen ja -version. Tätä tietoa voisi hyväksikäyttää yhdessä TLS-salauksen asetuksien kanssa vertailemalla näitä keskenään. Vertailu paljastaisi, kuinka moni verkkosivu käyttää suoraan WWW-palvelimien oletusasetuksia ilman erillisiä kovennuksia.

LÄHTEET

A. Freier, P. K. (2015). The Secure Sockets Layer (SSL) Protocol Version 3.0. *RFC 6101*.

Acunetix. (ei pvm.). *CRIME SSL/TLS attack*. https://www.acunetix.com/vulnerabilities/web/crime-ssl-tls-attack/

Adrian, D., Bhargavan, K., Durumeric, Z., Gaudry, P., Green, M., Halderman, J. A., Heninger, N., Springall, D., Thomé, E., Valenta, L., VanderSloot, B., Wustrow, E., Zanella-Béguelin, S., & Zimmermann, P. (2015). Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice. *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*, 5–17. https://doi.org/10.1145/2810103.2813707

Alashwali, E. S., Szalachowski, P., & Martin, A. (2019). Does "Www." Mean Better Transport Layer Security? *Proceedings of the 14th International Conference on Availability, Reliability and Security*, 1–7.

Ali, A. A., & Murah, M. Z. (2018). Security Assessment of Libyan Government Websites. *2018 Cyber Resilience Conference (CRC)*, 1–4.

Anubis. (2022). *Anubis*. GitHub. https://github.com/jonluca/Anubis

Apache HTTP Server. (ei pvm.). *Apache Module mod_ssl*. https://httpd.apache.org/docs/trunk/mod/mod_ssl.html#sslusestapling

Apple. (2020). *About upcoming limits on trusted certificates*. https://support.apple.com/en-us/HT211025

Barnes, R., Thomson, M., Pironti, A., & Langley, A. (2015). *Deprecating Secure Sockets Layer Version 3.0*. IETF. https://tools.ietf.org/html/rfc7568

Ben Wilson. (2020). *Reducing TLS Certificate Lifespans to 398 Days*. Mozilla Security Blog. https://blog.mozilla.org/security/2020/07/09/reducing-tls-certificate-lifespans-to-398-days/

Beurdouche, B., Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Kohlweiss, M., Pironti, A., Strub, P.-Y., & Zinzindohoue, J. K. (2015). A Messy State of the Union: Taming the Composite State Machines of TLS. *2015 IEEE Symposium on Security and Privacy*, 535–552. https://doi.org/10.1109/SP.2015.39

Blake-Wilson, S., Bolyard, N., Gupta, V., Hawk, C., & Moeller, B. (2006). *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS) [RFC 4492]*. IETF. https://datatracker.ietf.org/doc/html/rfc4492

Bou Sleiman, M., & Gerdemann, S. (2021). Covid-19: a catalyst for cybercrime? *International Cybersecurity Law Review*, *2*(1), 37–45. https://doi.org/10.1365/s43439-021-00024-9

CA/Browser Forum. (2017). *Ballot 193 – 825-day Certificate Lifetimes*. https://cabforum.org/2017/03/17/ballot-193-825-day-certificate-lifetimes/

CA/Browser Forum. (2019). *Ballot SC22 – Reduce Certificate Lifetimes (v2)*. https://cabforum.org/2019/09/10/ballot-sc22-reduce-certificate-lifetimes-v2/

Chromium. (ei pvm.). *Certificate Lifetimes*. https://chromium.googlesource.com/chromium/src/+/HEAD/net/docs/certificate_lifetimes.md

Davies, J. (2010). Implementing SSL/TLS Using Cryptography and PKI. Teoksessa *Implementing SSL/TLS Using Cryptography and PKI*. https://doi.org/10.1002/9781118255797

De Santis, F., Schauer, A., & Sigl, G. (2017). ChaCha20-Poly1305 authenticated encryption for high-speed embedded IoT applications. *Proceedings of the 2017 Design, Automation and Test in Europe, DATE 2017*. https://doi.org/10.23919/DATE.2017.7927078

Dierks, T., & Rescorla, E. (2008). *The Transport Layer Security (TLS) Protocol Version 1.2 [RFC5246]*. IETF. https://datatracker.ietf.org/doc/html/rfc5246

Digicert. (2021). *POODLE (TLS)*. https://docs.digicert.com/certificate-tools/discovery-user-guide/tlsssl-endpoint-vulnerabilities/poodle-tls/

Durumeric, Z., Ma, Z., Springall, D., Barnes, R., Sullivan, N., Bursztein, E., Bailey, M., Halderman, J. A., & Paxson, V. (2017). *The Security Impact of HTTPS Interception*. https://doi.org/10.14722/ndss.2017.23456

Dutch National Cyber Security Center. (2020). *IT Security Guidelines for Transport Layer Security (TLS)*.

E. Rescorla. (2018). RFC 8446 - The Transport Layer Security (TLS) Protocol Version 1.3. Teoksessa *Internet Engineering Task Force (IETF)* .

Elisa. (2021). *Minkälaisen verkkotunnuksen voi hankkia?* https://yrityksille.elisa.fi/verkkotunnus

Global Web Index, & Datum Future. (2019). *The Data Confidence Index*. https://www.datumfuture.org/wp-content/uploads/2019/09/Data-Confidence-Index-Datum-Future-and-GWI-2019.pdf

Google Developers. (2022). *Secure your site with HTTPS*. https://developers.google.com/search/docs/advanced/security/https

Goralski, W. (2009). *The illustrated network: How TCP/IP works in a modern network*. Elsevier/Morgan Kaufmann Publishers.

Hämäläinen, V.-P. (2021). *Vastaamon entiset potilaat vaativat jopa 10 000 euron korvauksia tietomurrosta – konkurssipesä pitää 2 500:aa euroa ylärajana*. Yle. https://yle.fi/uutiset/3-12134525

Hill, M., & Swinhoe, D. (2021). *The 15 biggest data breaches of the 21st century*. CSO. https://www.csoonline.com/article/2130877/the-biggest-data-breaches-of-the-21st-century.html

Hirsjärvi, S., Remes, P., & Sajavaara, P. (2016). *Tutki ja kirjoita* (21. uud. p). Tammi.

Hodges, J., Jackson, C., & Barth, A. (2012). *HTTP Strict Transport Security (HSTS)*. IETF. https://tools.ietf.org/html/rfc6797

Jackson, C., & Barth, A. (2008). ForceHTTPS: protecting high-security web sites from network attacks. *Proceeding of the 17th International Conference on World Wide Web 2008, WWW'08*. https://doi.org/10.1145/1367497.1367569

Kasanen, E., Lukka, K., & Siitonen, A. (1993). The Constructive Approach in

Management Accounting Research. *Journal of Management Accounting Researc*, *5*(June 1991), 243–264.

Kauppalehti. (2020). *Muutoksia ICB-toimialaluokituksissa Nasdaqin Euroopan markkinoilla*. https://www.kauppalehti.fi/lehdistotiedotteet/muutoksia-icb-toimialaluokituksissa-nasdaqin-euroopan-markkinoilla/185a919d-ede2-3b6a-a126-c19aa79138b7

Kotzias, P., Paterson, K. G., Razaghpanah, A., Vallina-Rodriguez, N., Amann, J., & Caballero, J. (2018). Coming of age: A longitudinal study of TLS deployment. *Proceedings of the ACM SIGCOMM Internet Measurement Conference, IMC*. https://doi.org/10.1145/3278532.3278568

Krombholz, K., Busse, K., Pfeffer, K., Smith, M., & Von Zezschwitz, E. (2019). "If HTTPS Were Secure, i Wouldn't Need 2FA" - End User and Administrator Mental Models of HTTPS. *Proceedings - IEEE Symposium on Security and Privacy*, *2019-May*. https://doi.org/10.1109/SP.2019.00060

Krombholz, K., Mayer, W., Schmiedecker, M., & Weippl, E. (2017). "I have no idea what I'm doing" – on the usability of deploying HTTPS. *Proceedings of the 26th USENIX Security Symposium*.

Kuhn, D. R., Hu, V. C., Polk, W. T., & Shu-Jen, C. (2001). SP 800-32: Introduction to Public Key Technology and the Federal PKI Infrastructure. *National Institute of Standards and Technology*.

Kurose, J., & Ross, K. (2013). *Computer Networking* (6. p.). Pearson Education UK.

Kyberturvallisuuskeskus. (2019). *TLS 1.2 -salausprotokollassa haavoittuvuus*. https://www.kyberturvallisuuskeskus.fi/fi/tls-12-salausprotokollassa-haavoittuvuus

Kyberturvallisuuskeskus. (2021). *Kryptografiset vahvuusvaatimukset luottamuksellisuuden suojaamiseen - kansalliset suojaustasot*. 1–5.

Langley, A., & Benjamin, D. (2021). *Feature: TLS 1.0 and TLS 1.1 (deprecated)*. Chrome Platform Status. https://www.chromestatus.com/feature/5654791610957824

Let's Encrypt. (2022). *Let's Encrypt Stats*. https://letsencrypt.org/stats/

Loshin, P. (2003). *TCP/IP Clearly Explained* (4. p.). Morgan Kaufmann Publishers.

Lukka, K. (1999). Case/field-tutkimuksen erilaiset lähestymistavat laskentatoimessa. Teoksessa H. Hookana-Turunen (Toim.), *Tutkija, opettaja, akateeminen vaikuttaja ja käytännön toimija : professori Reino Majala 65 vuotta* (ss. 129–150). Turun kauppakorkeakoulu.

Lukka, K. (2000). The key issues of applying the constructive approach to field research. Teoksessa T. Reponen (Toim.), *Management expertise for the new millennium : in commemoration of the 50th anniversary of the Turku School of Economics and Business Administration* (ss. 113–128). Turku School of Economics and Business Administration.

Lukka, K. (2001). *Konstruktiivinen tutkimusote*. Metodix. https://metodix.fi/2014/05/19/lukka-konstruktiivinen-tutkimusote/

Manfredi, S., Ceccato, M., Sciarretta, G., & Ranise, S. (2021). Do Security Reports Meet Usability?: Lessons Learned from Using Actionable Mitigations for Patching TLS Misconfigurations. *ACM International Conference Proceeding Series*. https://doi.org/10.1145/3465481.3469187

McKay, K., & Cooper, D. (2019). *Guidelines for the Selection, Configuration, and Use of Transport Layer Security (TLS) Implementations*. https://doi.org/10.6028/NIST.SP.800-52r2

MDN Web Docs. (2021). *Strict-Transport-Security*. https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Strict-Transport-Security

Menezes, A. J., Van Oorschot, P. C., & Vanstone, S. A. (1996). Handbook of applied cryptography. Teoksessa *Handbook of Applied Cryptography*. https://doi.org/10.2307/2589608

Microsoft. (2021a). *Disabling TLS 1.0 and 1.1 for Microsoft 365*. https://docs.microsoft.com/en-us/microsoft-365/compliance/tls-1.0-and-1.1-deprecation-for-office-365?view=o365-worldwide

Microsoft. (2021b). *Transport Layer Security (TLS) registry settings*. https://docs.microsoft.com/en-us/windows-server/security/tls/tls-registry-settings

Möller, B., Duong, T., & Kotowicz, K. (2014). This POODLE Bites: Exploiting The SSL 3.0 Fallback. *OpenSSL*.

Mozilla. (2020a). *Firefox for Enterprise 78 - Release notes*. https://www.mozilla.org/en-US/firefox/78.0/releasenotes/

Mozilla. (2020b). *Security/Server Side TLS*. Mozilla Wiki. https://wiki.mozilla.org/Security/Server_Side_TLS

Nasdaq. (2020). *Companies listed on Nasdaq Helsinki*. http://www.nasdaqomxnordic.com/shares/listed-companies/helsinki

Nasdaq. (2021). *Muutoksia Nasdaqin pohjoismaisten pörssien markkina-arvoluokissa*. https://view.news.eu.nasdaq.com/view?id=b9c9372225fa9a93ea7af24b08d15a25c&lang=fi

National Security Agency. (2021). *Eliminating Obsolete Transport Layer Security (TLS) Protocol Configurations*. 1–6. https://media.defense.gov/2021/Jan/05/2002560140/-1/-1/0/ELIMINATING_OBSOLETE_TLS_UOO197443-20.PDF

Nginx. (ei pvm.). *Module ngx_http_ssl_module*. https://nginx.org/en/docs/http/ngx_http_ssl_module.html#ssl_stapling

Nidecki, T. A. (2020). *What Is the POODLE Attack?* Acunetix. https://www.acunetix.com/blog/web-security-zone/what-is-poodle-attack/

NIST. (2012a). *CVE-2012-4929*. https://nvd.nist.gov/vuln/detail/CVE-2012-4929

NIST. (2012b). Recommendation for Key Management - Part 1: General. *NIST Special Publication 800-57*, *Revision 3*(July).

Omar Santos. (2013). *BREACH, CRIME and Black Hat*. Cisco. https://blogs.cisco.com/security/breach-crime-and-blackhat

Oppliger, R. (2016). *SSL and TLS : theory and practice* (2. p.). Artech House.

Owen Garrett. (2016). *HTTP Strict Transport Security (HSTS) and NGINX*. Nginx. https://www.nginx.com/blog/http-strict-transport-security-hsts-and-nginx/

Oyegoke, A. (2011). The constructive research approach in project management research. *International Journal of Managing Projects in Business*, *4*(4), 573–595. https://doi.org/10.1108/17538371111164029

Paganini, P. (2015). *The FREAK Vulnerability: From Discovery to Mitigation*. Infosec. https://resources.infosecinstitute.com/topic/the-freak-vulnerability-from-discovery-to-mitigation/

Patentti- ja rekisterihallitus. (2022). *Yritysten lukumäärät kaupparekisterissä*. https://www.prh.fi/fi/kaupparekisteri/yritystenlkm/lkm.html

Pflug, K. (2015). *HTTP Strict Transport Security comes to Internet Explorer 11 on Windows 8.1 and Windows 7*. Windows Blogs. https://blogs.windows.com/msedgedev/2015/06/09/http-strict-transport-security-comes-to-internet-explorer-11-on-windows-8-1-and-windows-7/

Rashid, S. M. Z. U., Kamrul, M. I., & Islam, A. (2019). Understanding the Security Threats of Esoteric Subdomain Takeover and Prevention Scheme. *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*, 1–4.

Red Hat. (2016). *SLOTH: TLS 1.2 vulnerability (CVE-2015-7575)*. Red Hat Customer Portal. https://access.redhat.com/articles/2112261

Reis, C., Gribble, S. D., Kohno, T., & Weaver, N. C. (2008). Detecting in-flight page changes with web tripwires. *5th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2008*.

Rescorla, E. (2018). *The Transport Layer Security (TLS) Protocol Version 1.3*. IETF. https://tools.ietf.org/html/rfc8446

Rizzo, J., & Duong, T. (2012). *The CRIME attack*. https://docs.google.com/presentation/d/11eBmGiHbYcHR9gL5nDyZChu_-lCa2GizeuOfaLU2HOU/edit#slide=id.g1d134dff_1_147

Rooney, T. (2010). *Introduction to IP address management*. Wiley-IEEE Press.

Sanchez, W. G. (2016). *SLOTH Downgrades TLS 1.2 Encrypted Channels*. Trend Micro. https://www.trendmicro.com/en_us/research/16/b/sloth-downgrades-tls-1-2-encrypted-channels.html

Scheuerman, W. E. (2014). Whistleblowing as civil disobedience: The case of Edward Snowden. *Philosophy and Social Criticism*, *40*(7). https://doi.org/10.1177/0191453714537263

Shi, Y., Bradley, M., Schonning, N., & Wenzel, M. (2020). *HSTS Settings for a Web Site <hsts>*. Microsoft. https://docs.microsoft.com/en-us/iis/configuration/system.applicationhost/sites/site/hsts

Shinder, L., & Cross, M. (2008). Understanding Cybercrime Prevention. Teoksessa *Scene of the Cybercrime*. https://doi.org/10.1016/b978-1-59749-276-8.00012-1

Silva, J. M. C., & Fonte, V. (2019). Data Security and Trustworthiness in Online Public Services: An Assessment of Portuguese Institutions. *Proceedings of the 12th International Conference on Theory and Practice of Electronic Governance*, 348–353.

Souppaya, M. (2018). PCI Data Security Standard Requirements and Security Assessment Procedures. *October*, *3.2.1*(May).

SSL Labs. (2020). *SSL and TLS Deployment Best Practices*. GitHub. https://github.com/ssllabs/research/wiki/SSL-and-TLS-Deployment-Best-Practices

SSL Labs. (2021). *SSL Pulse*. https://www.ssllabs.com/ssl-pulse/

Strzelecki, A., & Rizun, M. (2020). Consumers' security and trust for online shopping after GDPR: examples from Poland and Ukraine. *Digital Policy, Regulation and Governance*, *22*(4), 289–305.

The Chromium Projects. (2021). *HTTP Strict Transport Security*. https://www.chromium.org/hsts

The MITRE Corporation. (2014). *CVE-2014-3566*. CVE. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2014-3566

Traficom. (2019a). *Muut kuin fi-verkkotunnukset*. https://www.traficom.fi/fi/viestinta/fi-verkkotunnukset/muut-kuin-fi-verkkotunnukset

Traficom. (2019b). *Näin hankit fi-verkkotunnuksen*. https://www.traficom.fi/fi/viestinta/fi-verkkotunnukset/nain-hankit-fi-verkkotunnuksen

Traficom. (2019c). *Tietoa fi-maatunnuksesta*. https://www.traficom.fi/fi/viestinta/fi-verkkotunnukset/tietoa-fi-maatunnuksesta

Turner, S., & Polk, T. (2011). *Prohibiting Secure Sockets Layer (SSL) Version 2.0*. IETF. https://tools.ietf.org/html/rfc6176

U.S Chief Information Officers Council. (ei pvm.). *Compliance Guide*. https://https.cio.gov/guide/

Uusitalo, K., & Kohtamäki, M. (2011). *Menetelmäviidakon raivaajat : perusteita laadullisen tutkimuslähestymistavan valintaan* (A. Puusa & P. Juuti (toim.); ss. 281–295). JTO.

Vaso, J. (1998). *Ammatillisen aikuiskoulutuksen laatu*. Tampereen yliopisto.

Weerasinghe, T. D. B., & Disanayake, C. (2018). Usage of RC4 Cipher in SSL Configurations in Web Portals of Sri Lankan Banking/Non-Banking Financial Institutes and Awareness Levels of Relevant Staff About It. *2018 National Information Technology Conference (NITC)*, 1–6.

Wilson, J. L. (2020). *How to Register a Domain Name for Your Website*. PCMag. https://uk.pcmag.com/web-hosting/86838/how-to-register-a-domain-name-for-your-website

Woodfield, M. (2015). *FREAK Attack: What You Need to Know*. DigiCert. https://www.digicert.com/dc/blog/freak-attack-need-know/

Xia, H., & Brustoloni, J. C. (2005). Hardening Web Browsers against Man-in-the-Middle and Eavesdropping Attacks. *Proceedings of the 14th International Conference on World Wide Web*, 489–498. https://doi.org/10.1145/1060745.1060817

Young, C. (2019a). *Introducing Zombie POODLE and GOLDENDOODLE*. Tripwire. https://www.tripwire.com/state-of-security/vulnerability-management/zombie-poodle-goldendoodle/

Young, C. (2019b). *What is Zombie POODLE?* Tripwire.

https://www.tripwire.com/state-of-security/vert/zombie-poodle/

Zalewski, M. (2012). *The tangled Web : a guide to securing modern Web applications*. No Starch Press.

LIITE 1 YHTIÖDEN TOIMIALAT JA MARKKINA-ARVOT

Helsingin pörssissä yhtiöt jaetaan 11 eri toimialaluokkaan (Kauppalehti, 2020). Yhtiölle määritellään toimialaluokka sen pääasiallisen liiketoiminnan ja muun julkisesti saatavilla olevan tiedon perusteella (Kauppalehti, 2020). Yhtiöt myös jaetaan markkina-arvon perusteella kolmeen eri markkina-arvoryhmään, jotka ovat suuret (yli miljardi euroa), keskisuuret (100 miljoonaa – yksi miljardia euroa) ja pienet yhtiöt (alle 100 miljoonaa euroa) (Nasdaq, 2021). Yhtiön markkina-arvo määräytyy kertomalla yhtiön osakkeiden lukumäärä osakkeen kurssilla (Nasdaq, 2021). Tutkimus hetkellä Helsingin pörssiin kuului 133 yhtiötä, joiden toimialaluokat ja markkina-arvoryhmät näet alla olevasta taulukosta.

TAULUKKO 26 Helsingin pörssiyhtiöiden jakautuminen eri toimialaluokkien ja markkina-arvoryhmien välillä (Nasdaq, 2020)

Toimialaluokka	**Suuret yhtiöt**	**Keskisuuret yhtiöt**	**Pienet yhtiöt**	**Yhteensä**
Teknologia	1	5	12	**18**
Tietoliikennepalvelut	3	0	1	**4**
Terveydenhuolto	2	3	2	**7**
Rahoitus	2	8	4	**14**
Kiinteistöyhtiöt	2	0	2	**4**
Kulutushyödykkeet	3	14	8	**25**
Peruskulutustuotteet	1	6	1	**8**
Teollisuustuotteet ja -palvelut	9	9	21	**39**
Perusteollisuus	6	1	4	**11**
Energia	1	0	0	**1**
Yleishyödylliset palvelut	1	1	0	**2**
Yhteensä	**31**	**47**	**55**	**133**

LIITE 2 SUOSITELLUT SALAUSSARJAT

TLS 1.3 -protokollan suositellut salaussarjat:

- TLS_AES_128_CCM_8_SHA256
- TLS_AES_128_CCM_SHA256
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384
- TLS_CHACHA20_POLY1305_SHA256

TLS 1.2 -protokollan suositellut salaussarjat:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_CCM
- TLS_DHE_RSA_WITH_AES_128_CCM_8
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_256_CCM
- TLS_DHE_RSA_WITH_AES_256_CCM_8
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM
- TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CCM
- TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256

LIITE 3 ESIMERKKI DATA

Tässä liitteessä on esimerkkiaineistoa, jota työkalu keräsi. Alla oleva objekti on esimerkki *nasdaq_helsinki*-taulusta:


```
{
    "name": "Yritys Oy",
    "symbol": "1234",
    "CCY": "EUR",
    "ISIN": "FI1245",
    "ICB": "1234",
    "employees": 1234,
    "industry": "example",
    "sector": "example",
    "website": "www.esimerkkisivu.fi"
}
```


Seuraava objekti on esimerkki *hosts*-taulusta:


```
{
    "domain": "www.esimerkkisivu.fi",
    "address": "1.2.3.4"
}
```


Alla esimerkki on auki olevasta 443-portista *nmap*-taulussa. Esimerkkiobjektista on siistitty pois ylimääräinen tieto tilan säästämisen vuoksi:


```
{
    "addresses": {
        "ipv4": "1.2.3.4"
    },
    "tcp": {
        "443": {
            "state": "open",
            "reason": "syn-ack",
            "name": "https",
            "product": "",
            "version": "",
            "extrainfo": "",
            "conf": "3",
            "cpe": ""
        }
    }
}
```


Esimerkkiobjektit kahdesta erilaisesta ajossa tulleesta virheestä *errors*-taulussa:


```
{
    "error_msg": "'NoneType' object is not iterable",
    "host": "www.esimerkkisivu.fi"
}

{
    "error_msg": "Could not resolve www.esimerkkisivu.fi",
    "host": "www.esimerkkisivu.fi"
}
```

Alla objektiesimerkki on *sslyze_helsinki*-taulusta. Tilan säästämisen takia esimerkkiin on poimittu vain oleellinen tieto, jota tutkimuksessa käytettiin:


```
{
    "scan_commands_results": {
        "tls_1_3_cipher_suites": {
            "tls_version_used": "TLS_1_3",
            "accepted_cipher_suites": [{
                "cipher_suite": {
                    "name": "TLS_CHACHA20_POLY1305_SHA256",
                    "is_anonymous": false,
                    "key_size": 256,
                    "openssl_name": "TLS_CHACHA20_POLY1305_SHA256"
                },
            }],
            "rejected_cipher_suites": [{
                "cipher_suite": {
                    "name": "TLS_AES_128_CCM_SHA256",
                    "is_anonymous": false,
                    "key_size": 128,
                    "openssl_name": "TLS_AES_128_CCM_SHA256"
                },
                "error_message": "TLS alert: handshake failure"
            }]
        },
        "tls_1_3_early_data": {
            "supports_early_data": false
        },
        "tls_1_2_cipher_suites": {...},
        "ssl_3_0_cipher_suites": {...},
        "ssl_2_0_cipher_suites": {...},
        "tls_1_1_cipher_suites": {...},
        "tls_1_0_cipher_suites": {...},
        "http_headers": {
            "strict_transport_security_header": {
                "max_age": 63072000,
                "preload": false,
                "include_subdomains": false
        }},
        "certificate_info": {
            "certificate_deployments": [{
                "received_certificate_chain": [{
                    "not_valid_before": "2020-12-08T00:00:00",
                    "not_valid_after": "2021-12-08T23:59:59",
                    "public_key": {
                        "algorithm": "_RSAPublicKey",
                        "key_size": 4096,
                        "rsa_e": 65537,
                        "ec_curve_name": null
                }}],
                "leaf_certificate_subject_matches_hostname": true,
                "ocsp_response_is_trusted": null
        }]}
    },
    "scan_commands_errors": {},
    "server_info": {
        "server_location": {
            "hostname": "www.esimerkkisivu.fi",
            "port": 443,
            "ip_address": "1.2.3.4"
        },
    },
}
```

LIITE 4 TYÖKALUN ENSIMMÄINEN VAIHE

Tähän liitteeseen on koottu työkalun ajon ensimmäiseen vaiheeseen liittyvät ohjelmakoodi. Alta löytyy *get_listed_companies.py*-ohjelma:

```
from bs4 import BeautifulSoup
from pymongo import MongoClient
import requests
from models.company import Company

client = MongoClient('mongodb://localhost:27017/')
db = client['jyu_tls_research']

def get_listed_companies(market, verbose=True):
    import yfinance as yf

    headers = {
         'User-Agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/86.0.4240.198 Safari/537.36'
        }
      r  =  requests.get("http://www.nasdaqomxnordic.com/osakkeet/listayhtiot/"+market,
headers=headers)
    soup = BeautifulSoup(r.text, 'html.parser')

    markets_suffix = {
        "DKK": ".CO",
        "EUR": ".HE",
        "ISK": ".IC",
        "SEK": ".ST",
    }

    companies = []
    rows = soup.find(id="listedCcompanies").find("tbody").find_all("tr")
    for row in rows:
        cells = row.find_all("td")
        name = cells[0].get_text()
         symbol  =  cells[1].get_text().replace('NDA FI', 'NDA-FI') # Nordea was "NDA
FI", but should be "NDA-FI"
        CCY = cells[2].get_text()
        ISIN = cells[3].get_text()
        #sector = cells[4].get_text()
        ICB = cells[5].get_text()
        if verbose == True: print(name, symbol)

        try:
            ticker = yf.Ticker(symbol + markets_suffix[CCY])
            employees = ticker.info["fullTimeEmployees"]
            industry = ticker.info["industry"]
            sector = ticker.info["sector"]
            website = ticker.info["website"]

               company  =  Company(name,  symbol,  CCY,  ISIN,  ICB,  employees,  industry,
sector, website)
            companies.append(company)
        except KeyError as err:
            print("KeyError: " + str(err))
        except ValueError as err:
            print("ValueError: " + str(err))
```

```
    return companies

def get_listed_companies_from_cache(market, verbose=True):
    collection = db['nasdaq_'+market]
    entries = collection.find({})
    companies = []
    for entry in entries:
        company = Company(entry["name"], entry["symbol"], entry["CCY"], entry["ISIN"],
entry["ICB"],       entry["employees"],       entry["industry"],       entry["sector"],
entry["website"])
        companies.append(company)
    return companies

if __name__ == '__main__':
    market = "helsinki"
    collection = db['nasdaq_'+market]

    companies = get_listed_companies(market)
    collection.delete_many({})
    for company in companies:
        collection.insert_one(company.__dict__)

    client.close()
```

Alta löytyy *models/company.py*-ohjelma:

```
class Company:
    name = ''
    symbol = ''
    CCY = ''
    ISIN = ''
    ICB = ''
    employees = ''
    industry = ''
    sector = ''
    website = ''
    ssllabs_result = ''

     def __init__(self, name, symbol, CCY, ISIN, ICB, employees, industry, sector,
website):
        self.name = name
        self.symbol = symbol
        self.CCY = CCY
        self.ISIN = ISIN
        self.ICB = ICB
        self.employees = employees
        self.industry = industry
        self.sector = sector
        self.website = website

    def __str__(self):
          return "%s, %s, %s, %s, %s, %s, %s, %s, %s" % (self.name, self.symbol,
self.CCY,   self.ISIN,   self.ICB,   self.employees,   self.industry,   self.sector,
self.website)
```

Alta löytyy *subdomains.py*-ohjelma:

```
#!/usr/bin/env python

from get_listed_companies import get_listed_companies_from_cache
```

```
import tldextract

def run_scan(market, debug):

    companies = get_listed_companies_from_cache(market, debug)
    domain_list = []
    for company in companies:
        ext_domain = tldextract.extract(company.website)
        domain = ext_domain.domain + "." + ext_domain.suffix
        domain_list.append(domain)

    domain_list = list(set(domain_list))
    for domain in domain_list:
        print(domain)

if __name__ == '__main__':
    run_scan('helsinki', True)
```

LIITE 5 TYÖKALUN KOLMAS VAIHE

Tähän liitteeseen on koottu työkalun ajon kolmanteen vaiheeseen liittyvät ohjelmakoodi. Alta löytyy *Tnmap_scanner.py*-ohjelma:

```
#!/usr/bin/env python

import nmap
from pymongo import MongoClient
from pprint import pprint
import tldextract
import copy
import argparse
import re

client = MongoClient('mongodb://localhost:27017/')
db = client['jyu_tls_research']
collection = db['nmap']

def scan():
    hosts_collection = db['hosts']
    entries = hosts_collection.find({},{"address":1})
    hosts = [e.get("address") for e in entries]
    hosts = list(set(hosts)) # Remove duplicate ip addresses

    for counter, host in enumerate(hosts):
        print("counter: %s\thost: %s" % (counter,host))
        run_scan(host)

def run_scan(host):
    nm = nmap.PortScanner()
    nm.scan(host, '80,443', '-n -Pn')
    for host in nm.all_hosts():
        host2 = copy.deepcopy(nm[host])
        for x in nm[host]["tcp"]:
            del host2["tcp"][x]
            host2["tcp"][str(x)] = nm[host]["tcp"][x]
        collection.insert_one(host2)

def empty_scan_results():
    collection.delete_many({})

def get_result_by_ip():
         entries    =    collection.find({    "$or":    [    {    "tcp.80.state":"open"},
{"tcp.443.state":"open"} ] }, {"addresses":1, "tcp.80.state":1, "tcp.443.state":1})
    for entry in entries:
         print('address: %s    \tport: 80\tstate : %s' % (entry["addresses"]["ipv4"],
entry["tcp"]["80"]["state"]))
        print('address: %s    \tport: 443\tstate : %s' % (entry["addresses"]["ipv4"],
entry["tcp"]["443"]["state"]))

def get_addresses_with_open_https_port():
          entries    =    collection.find({"tcp.443.state":"open"},    {"addresses":1,
"tcp.443.state":1})
    addresses = []
    for entry in entries:
        addresses.append(entry["addresses"]["ipv4"])
    return addresses
```

```
def get_addresses_with_open_http_port():
          entries    =    collection.find({"tcp.80.state":"open"},    {"addresses":1,
"tcp.80.state":1})
    addresses = []
    for entry in entries:
        addresses.append(entry["addresses"]["ipv4"])
    return addresses

def update_mongodb_host_list():
    from models.host import Host

    collection = db['hosts']
    collection.delete_many({})
    domains = get_domains_from_anubis_file()
    host_list = get_hosts_from_anubis_file()
    for x in host_list:
        line = x.replace('\n','')
        split = line.split(': ')
        domain = split[0]
        address = split[1]
        host = Host(domain, address)
        if is_valid_domain(domains, domain):
            collection.insert_one(host.__dict__)

def get_domains_from_anubis_file():
    domains = open('data/domains.txt', 'r').readlines()
    domains = [i.replace('\n','') for i in domains] # Remove \n character from every
end of line
    return domains

def get_hosts_from_anubis_file():
    file = open('data/anubis_result.txt', 'r').readlines()
    file = [i.replace('\n','') for i in file] # Remove \n character from every end of
line
    host_list = []
    for line in file:
           if  re.compile(r':').search(line)  and  not  re.compile(r'Subdomain  search
took').search(line) and not re.compile(r'Working on target').search(line):
            split = line.split(': ')
            address = split[1]
            if address:
                host_list.append(line)

    return host_list

def is_valid_domain(domains, subdomain):
    tldextract_domain = tldextract.extract(subdomain)
    if tldextract_domain.domain + "." + tldextract_domain.suffix in domains:
        return True
    else:
        return False

def get_result_by_host():
    collection = db['hosts']
    entries = collection.find({"address": {"$ne":""}})

    open_https_addresses = get_addresses_with_open_https_port()
    open_http_addresses = get_addresses_with_open_http_port()

    open = 0
    open_https = 0
    open_http = 0
    open_http_and_https = 0
    closed = 0
```

```
    for entry in entries:
            if entry["address"] in open_https_addresses and entry["address"] in
open_http_addresses:
                print('OPEN     -  address:  %s   \tport:80/443\tdomain:  %s  \t'  %
(format_address(entry["address"]), entry["domain"]))
            open=open+1
            open_http_and_https=open_http_and_https+1
        elif entry["address"] in open_https_addresses:
                print('OPEN     -  address:  %s   \tport:443\tdomain:  %s  \t'  %
(format_address(entry["address"]), entry["domain"]))
            open=open+1
            open_https=open_https+1
        elif entry["address"] in open_http_addresses:
                print('OPEN     -  address:  %s   \tport:80\t\tdomain:  %s  \t'  %
(format_address(entry["address"]), entry["domain"]))
            open=open+1
            open_http=open_http+1
        else:
                print('CLOSED   -  address:  %s   \tport:-\t\tdomain:  %s  \t'  %
(format_address(entry["address"]), entry["domain"]))
            closed=closed+1
    print('Result: Open: %s, Closed: %s' % (open, closed))
    print('Open http: %s, Open https: %s, Open both: %s' % (open_http, open_https,
open_http_and_https))

def format_address(address):
    if len(address) < 13:
        return address + "\t"
    return address

if __name__ == '__main__':
    parser = argparse.ArgumentParser(description='pyTLScanner - Nmap scanner')
    parser.add_argument('--scan', action="store_true", dest="scan", help="Scan hosts")
    parser.add_argument('--empty-results', action="store_true", dest="empty_results",
help="Empty old scan results")
      parser.add_argument('--result-ip',  action="store_true",  dest="result_by_ip",
help="Get results by ip address")
    parser.add_argument('--result-host', action="store_true", dest="result_by_host",
help="Get results by host")
       parser.add_argument('--update-db',   action="store_true",   dest="update_db",
help="Update MongoDB from domains.txt and hosts.txt")
    args = parser.parse_args()

    if args.update_db:
        update_mongodb_host_list()
    if args.empty_results:
        empty_scan_results()
    if args.scan:
        scan()
    if args.result_by_ip:
        get_result_by_ip()
    if args.result_by_host:
        get_result_by_host()

    client.close()
```

LIITE 6 TYÖKALUN NELJÄS VAIHE

Tähän liitteeseen on koottu työkalun ajon neljänteen vaiheeseen liittyvät ohjelmakoodi. Alta löytyy *pytlscanner.py*-ohjelma:

```
#!/usr/bin/env python

from pytlscanner import resultsFromCache
from pytlscanner import sslyze_scan
from pymongo import MongoClient
import time
import argparse
from pprint import pprint
from dataclasses import asdict
import json
import sslyze
from sslyze import ScanCommand
import requests
import copy

from datetime import datetime

client = MongoClient('mongodb://localhost:27017/')

list_of_market_choices = ["baltic", "copenhagen", "helsinki", "iceland", "stockholm",
"first-north", "first-north-premier"]
parser = argparse.ArgumentParser(description='pyTLScanner')
parser.add_argument("--market",   dest='market',   help="Select   a   target   market",
choices=list_of_market_choices)
parser.add_argument('--debug',    action="store_true",    dest="debug",    help="Debug
logging")
args = parser.parse_args()

def run_sslyze_scan(market, debug):
    """Run SSLyze scanner against selected market's websites

    Args:
        market ([type]): Target market (e.g. helsinki)
        debug ([type]): Debug SSLyze
    """
    db = client['jyu_tls_research']
    collection = db['sslyze_'+market]
    error_collection = db['errors']
    hosts = get_all_hosts(db)
    open_https_addresses = get_addresses_with_open_https_port(db)

    for index, host in enumerate(hosts):
        if host['address'] in open_https_addresses:
            domain = host['domain']
            current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
            print(f"{current_time} \t index: {index} \t domain: {domain}")
            try:
                scanner_results = scan(domain, debug)
                result_as_json = load_scan_result(scanner_results)
                #pprint(result_as_json)
                collection.insert_one(result_as_json)
            except TypeError as e:
                msg = error_message(e, host['domain'])
                error_collection.insert_one(msg)
```

```
            except KeyError as e:
                msg = error_message(e, host['domain'])
                error_collection.insert_one(msg)
            except sslyze.errors.ServerHostnameCouldNotBeResolved as e:
                msg = error_message(e, host['domain'])
                error_collection.insert_one(msg)

    client.close()

def error_message(error, domain):
    current_time = datetime.now().strftime("%Y-%m-%d %H:%M:%S")
    print(f"{current_time} Error with domain: {domain}")
    return { "error_msg": str(error) , "host": domain}

def scan(host, debug):
    scan_commands={
        ScanCommand.CERTIFICATE_INFO,
        ScanCommand.SSL_2_0_CIPHER_SUITES,
        ScanCommand.SSL_3_0_CIPHER_SUITES,
        ScanCommand.TLS_1_0_CIPHER_SUITES,
        ScanCommand.TLS_1_1_CIPHER_SUITES,
        ScanCommand.TLS_1_2_CIPHER_SUITES,
        ScanCommand.TLS_1_3_CIPHER_SUITES,
        ScanCommand.TLS_1_3_EARLY_DATA,
        #ScanCommand.HEARTBLEED,
        #ScanCommand.ROBOT,
        ScanCommand.ELLIPTIC_CURVES,
        ScanCommand.HTTP_HEADERS,
        ScanCommand.TLS_COMPRESSION,
        #ScanCommand.TLS_FALLBACK_SCSV,
        #ScanCommand.OPENSSL_CCS_INJECTION,
        ScanCommand.SESSION_RENEGOTIATION,
        ScanCommand.SESSION_RESUMPTION,
        #ScanCommand.SESSION_RESUMPTION_RATE,
    }
    return sslyze_scan(host, scan_commands, debug)

def load_scan_result(scanner_results):
    for scan_result in scanner_results:
                    result_as_json    =    json.loads(json.dumps(asdict(scan_result),
cls=sslyze.JsonEncoder))
        result_as_json2 = copy.deepcopy(result_as_json)

                                        for         i_deploy,         deploy         in
enumerate(result_as_json2['scan_commands_results']['certificate_info']['certificate_d
eployments']):
            if 'ocsp_response' in deploy and deploy['ocsp_response'] is not None:
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['ocsp_response']['serial_number']
            if 'received_certificate_chain' in deploy:
                for i_cert, certs in enumerate(deploy['received_certificate_chain']):
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['received_certificate_chain'][i_cert]['serial_number']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['received_certificate_chain'][i_cert]['public_key']['rsa_n']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['received_certificate_chain'][i_cert]['public_key']['ec_x']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['received_certificate_chain'][i_cert]['public_key']['ec_y']
                if 'path_validation_results' in deploy:
```

```
                    for i_path, path in enumerate(deploy['path_validation_results']):
                                        if 'verified_certificate_chain' in path and
path['verified_certificate_chain'] is not None:
                                                                for i_chain, chain in
enumerate(path['verified_certificate_chain']):
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['path_validation_results'][i_path]['verified_certificate_chain'][i_chain]
['serial_number']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['path_validation_results'][i_path]['verified_certificate_chain'][i_chain]
['public_key']['rsa_n']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['path_validation_results'][i_path]['verified_certificate_chain'][i_chain]
['public_key']['ec_x']
                                                                                    del
result_as_json['scan_commands_results']['certificate_info']['certificate_deployments'
][i_deploy]['path_validation_results'][i_path]['verified_certificate_chain'][i_chain]
['public_key']['ec_y']
    return result_as_json

def redirect_to_https(host):
    r = requests.get("http://"+host, allow_redirects=True)
    if 'https://' in r.url:
        return True
    else:
        return False

def get_all_hosts(db):
    collection = db['hosts']
    entries = collection.find({"address": {"$ne":""}})
    hosts = []
    for entry in entries:
        hosts.append(entry)
    entries.close()
    return hosts

def get_addresses_with_open_https_port(db):
    collection = db['nmap']
          entries    =    collection.find({"tcp.443.state":"open"},    {"addresses":1,
"tcp.443.state":1})
    addresses = []
    for entry in entries:
        addresses.append(entry["addresses"]["ipv4"])
    entries.close()
    return addresses

if __name__ == '__main__':
    run_sslyze_scan(args.market, args.debug)
```

Alta löytyy *pytlscanner/sslyze_scan.py*-ohjelma:

```
from sslyze import (
    ServerNetworkLocationViaDirectConnection,
    ServerConnectivityTester,
    Scanner,
    ServerScanRequest,
    ScanCommand,
)
from sslyze.errors import ConnectionToServerFailed
```

```
from pprint import pprint

def sslyze_scan(host, scan_commands, debug=False) -> None:
    # First validate that we can connect to the servers we want to scan
    servers_to_scan = []
    for hostname in [host]:
                                                        server_location             =
ServerNetworkLocationViaDirectConnection.with_ip_address_lookup(hostname, 443)
        try:
            server_info = ServerConnectivityTester().perform(server_location)
            servers_to_scan.append(server_info)
        except ConnectionToServerFailed as e:
                                                print(f"Error      connecting      to
{server_location.hostname}:{server_location.port}: {e.error_message}")
            return

    scanner = Scanner()

    # Then queue some scan commands for each server
    for server_info in servers_to_scan:
        server_scan_req = ServerScanRequest(
            server_info=server_info, scan_commands=scan_commands,
        )
        scanner.queue_scan(server_scan_req)

        if debug:
            print_results(scanner)

        return scanner.get_results()

def print_results(scanner):
    # Then retrieve the result of the scan commands for each server
    for server_scan_result in scanner.get_results():
                                                      print(f"\nResults             for
{server_scan_result.server_info.server_location.hostname}:")

        # Scan commands that were run with errors
        for scan_command, error in server_scan_result.scan_commands_errors.items():
            print(f"\nError when running {scan_command}:\n{error.exception_trace}")

def cipher_suites(server_scan_result, protocol):
    try:
        result = server_scan_result.scan_commands_results[eval(protocol)]
        print(f"\nAccepted cipher suites for {protocol}:")
        for accepted_cipher_suite in result.accepted_cipher_suites:
            print(f"* {accepted_cipher_suite.cipher_suite.name}")
    except KeyError:
        pass

def get_certificate_info(server_scan_result):
    try:
                                                        certinfo_result             =
server_scan_result.scan_commands_results[ScanCommand.CERTIFICATE_INFO]
        print("\nCertificate info:")
        for cert_deployment in certinfo_result.certificate_deployments:
                                                     print(f"Leaf        certificate:
\n{cert_deployment.received_certificate_chain_as_pem[0]}")
    except KeyError:
        pass

def get_http_headers(server_scan_result):
    try:
        result = server_scan_result.scan_commands_results[ScanCommand.HTTP_HEADERS]
        print("\nHTTP_HEADERS info:")
        pprint(result)
```

```
    except KeyError:
        pass

def get_common(server_scan_result, scan_literal):
    try:
        ec_result = server_scan_result.scan_commands_results[eval(scan_literal)]
        print(f"\n{scan_literal} info:")
        pprint(ec_result)
    except KeyError:
        pass

if __name__ == '__main__':
    scan_commands={
            ScanCommand.CERTIFICATE_INFO
        }
    sslyze_scan("cloudflare.com", scan_commands, True)
```

LIITE 7 TYÖKALUN AJON TULOKSET

Osakelajien määrä oli 140 kappaletta, joka oli *nasdaq_helsinki*-taulun objektien määrä. Pääverkkotunnuksien ja yhtiöiden määrä saatiin *nasdaq_helsinki*-tauluun tehdyllä kyselyllä:

```
[{$group: {_id: '$website'}}]
```

Löydettyjä verkkotunnuksia oli 7946 kappaletta, joka oli *hosts*-taulun objektien määrä. IP-osoitteiden yhteismäärä oli 5099 kappaletta, joka oli *nmap*-taulun objetien määrä. IP-osoitteet, jotka kuuntelivat porttia 443 saatiin seuraavalla kyselyllä *nmap*-tauluun:

```
[{$match: {'tcp.443.state': {$eq: 'open'}}}]
```

Verkkotunnukset, jotka ohjautuivat porttia 443 kuuntelevaan IP-osoitteseen olivat *error*-taulun 220 objektien ja *sslyze_helsinki*-taulun 4562 objetien yhteenlaskettu summa. Eli 4782. Virheet olivat *error*-taulun objekten sekä *sslyze_helsinki*-tauluun tehdyn kysely tuloksien yhteenlaskettu summa:

```
[{$match:
  {$or:[
    {'scan_commands_errors.ssl_3_0_cipher_suites':{$exists:true}},
    {'scan_commands_errors.http_headers':{$exists:true}},
    {'scan_commands_errors.ssl_2_0_cipher_suites':{$exists:true}},
    {'scan_commands_errors.tls_compression':{$exists:true}},
    {'scan_commands_errors.tls_1_0_cipher_suites':{$exists:true}},
    {'scan_commands_errors.tls_1_1_cipher_suites':{$exists:true}},
    {'scan_commands_errors.elliptic_curves':{$exists:true}},
    {'scan_commands_errors.session_resumption':{$exists:true}},
    {'scan_commands_errors.session_renegotiation':{$exists:true}},
    {'scan_commands_errors.certificate_info':{$exists:true}},
    {'scan_commands_errors.tls_1_2_cipher_suites':{$exists:true}},
    {'scan_commands_errors.tls_1_3_early_data':{$exists:true}},
    {'scan_commands_errors.tls_1_3_cipher_suites':{$exists:true}}
  ]}
]
```

Validit verkkotunnukset saatiin *sslyze_helsinki*-tauluun tehdyllä kyselyllä:

```
[{$match:
  {$and:[{scan_commands_errors:{$exists:true}},
    {'scan_commands_errors.ssl_3_0_cipher_suites':{$exists:false}},
    {'scan_commands_errors.http_headers':{$exists:false}},
    {'scan_commands_errors.ssl_2_0_cipher_suites':{$exists:false}},
    {'scan_commands_errors.tls_compression':{$exists:false}},
    {'scan_commands_errors.tls_1_0_cipher_suites':{$exists:false}},
    {'scan_commands_errors.tls_1_1_cipher_suites':{$exists:false}},
    {'scan_commands_errors.elliptic_curves':{$exists:false}},
    {'scan_commands_errors.session_resumption':{$exists:false}},
    {'scan_commands_errors.session_renegotiation':{$exists:false}},
    {'scan_commands_errors.certificate_info':{$exists:false}},
    {'scan_commands_errors.tls_1_2_cipher_suites':{$exists:false}},
    {'scan_commands_errors.tls_1_3_early_data':{$exists:false}},
    {'scan_commands_errors.tls_1_3_cipher_suites':{$exists:false}}
  ]}
}]
```

Validien verkkotunnusten kyselystä tehtiin oma näkymänsä nimeltä *view_sslyze_helsinki_no_errors*, johon tehtiin kaikki suosituksia koskevat kyselyt.

LIITE 8 SUOSITUSKATEGORIOIDEN KYSELYT

Tässä liitteessä olevat kyselyt ovat tehty niin, että jokaisen kyselyn lopputuloksena on verkkotunnukset, jotka alittivat asetetun suosituksen. Tämä siitä syystä, että lopuksi oli helppo tarkistaa verkkotunnukset, jotka eivät löytyneet mistään näistä kyselyistä. Jos verkkotunnusta ei löytynyt mistään kyselystä, täytti kyseinen verkkotunnus kaikki asetetut suositukset. Kaikki tässä liitteessä olevat kyselyt tehtiin *view_sslyze_helsinki_no_errors*-näkymään, joka määriteltiin aikaisemmassa liitteessä. Seuraavalla kyselyllä saadaan kaikki SSL/TLS-version alittaneet verkkotunnukset:

```
[
  {$match:{$or:[
    {'scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites':{$ne:[]}},
    {'scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites':{$ne:[]}},
    {'scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites':{$ne:[]}},
    {'scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites':{$ne:[]}}
  ]}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Suositellut salaussarjat olivat koottu *cipher_suites*-tauluun, jotta pystyttiin vertaamaan, kuinka monta verkkotunnuksen salaussarjaa alitti suositukset. Suositellut salaussarjat olivat tallennettu MongoDB:n seuraavassa muodossa:

```
{"_id": "TLS_AES_128_CCM_8_SHA256"}
```

Seuraavalla kyselyllä saadaan kaikki salaussarjat-suosituksen alittaneet verkkotunnukset, johon hyödynnettiin *cipher_suites*-taulua:

```
[
  {$lookup:{
    from:'cipher_suites',

localField:'scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites.cipher
_suite.name',
    foreignField:'_id',
    as:'valid_cipher_suites'}},
  {$project:{
    scan_commands_results:1,
    valid_cipher_suites:1,
    'server_info.server_location.hostname':1,
    accepted_cipher_suites_size:
      {$size:'$scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites'},
    valid_cipher_suites_size:{$size:'$valid_cipher_suites'}}},
  {$match:{$or:[
    {$expr:{$ne:['$accepted_cipher_suites_size','$valid_cipher_suites_size']}},

{'scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites':{$exists:true,$
ne:[]}},

{'scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites':{$exists:true,$
ne:[]}},

{'scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites':{$exists:true,$
ne:[]}},
```

```
{'scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites':{$exists:true,$
ne:[]}}]}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan TLS-pakkauksen alittaneet suositukset:

```
[
  {$match:{'scan_commands_results.tls_compression.supports_compression':{$eq:true}}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan kaikki verkkotunnukset, joilla oli 0-RTT päällä:

```
[

{$match:{'scan_commands_results.tls_1_3_early_data.supports_early_data':{$eq:true}}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan ne verkkotunnukset, joilla oli OCSP stapling oli poissa päältä:

```
[
  {$unwind: {
    path: '$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays: false}},
  {$match:
{'scan_commands_results.certificate_info.certificate_deployments.ocsp_response_is_tru
sted': {$ne: true}}},
  {$group: {_id: '$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan kaikki varmenteen avaimen koko -suosituksen alittaneet verkkotunnukset:

```
[
  {$unwind:{
    path:'$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':
      {$slice:

['$scan_commands_results.certificate_info.certificate_deployments.received_certificat
e_chain',1]},
    'server_info.server_location.hostname':1}},
  {$unwind:{

path:'$scan_commands_results.certificate_info.certificate_deployments.received_certif
icate_chain',
    preserveNullAndEmptyArrays:false}},
  {$match:{$and:[
{'scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.public_key.algorithm':
      {$eq:'_RSAPublicKey'}},
{'scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.public_key.key_size':
      {$lt:2048}}]}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan verkkotunnukset, joiden jokin varmenteista ei ollut datan keräyshetkellä voimassa:

```
[
  {$unwind:{path:
    '$scan_commands_results.certificate_info.certificate_deployments'
    ,preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':

{$slice:['$scan_commands_results.certificate_info.certificate_deployments.received_ce
rtificate_chain',1]},
    'server_info.server_location.hostname':1}},
  {$unwind:{

path:'$scan_commands_results.certificate_info.certificate_deployments.received_certif
icate_chain',
    preserveNullAndEmptyArrays:false}},
  {$addFields:{not_valid_before:{$toDate:
'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_before'}}},
  {$addFields:{not_valid_after:{$toDate:
'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_after'}}},
  {$match:{$or:[
    {not_valid_before:{$gt:ISODate('2021-10-03T00:00:00.000Z')}},
    {not_valid_after:{$lte:ISODate('2021-09-27T00:00:00.000Z')}}]}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan kaikki verkkotunnukset, joiden varmenne oli yli 398 päivää (34387200 sekunttia eli 398 pv * 24 tuntia * 60 min * 60 sec) voimassa:

```
[
  {$unwind:{path:
    '$scan_commands_results.certificate_info.certificate_deployments'
    ,preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':
      {$slice:[

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain',1]},
    'server_info.server_location.hostname':1}},
  {$unwind:{path:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain'
    ,preserveNullAndEmptyArrays:false}},
  {$addFields:{not_valid_before:{$toDate:
'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_before'}}},
  {$addFields:{not_valid_after:{$toDate:
'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_after'}}},
  {$project:{
    certificate_lifespan:{$dateDiff:
      {startDate:'$not_valid_before',endDate:'$not_valid_after',unit:'second'}},
    not_valid_after:1,
    not_valid_before:1,
    'server_info.server_location.hostname':1}},
```

```
  {$match:{certificate_lifespan:{$gte:34387200}}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavalla kyselyllä saadaan verkkotunnukset, joiden jokin varmenteista käytti yleisnimeä, joka ei täsmännyt käytetyn verkkotunnuksen kanssa:

```
[
  {$unwind:{path:
    '$scan_commands_results.certificate_info.certificate_deployments'
    ,preserveNullAndEmptyArrays:false}},
  {$match:{'
scan_commands_results.certificate_info.certificate_deployments.leaf_certificate_subje
ct_matches_hostname':
      {$eq:false}}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

Seuraavassa kyselyssä tulokseksi jää verkkotunnukset, jotka eivät olleet kytkeneet HSTS-ominaisuutta päälle:

```
[

{$match:{'scan_commands_results.http_headers.strict_transport_security_header':{$eq:n
ull}}},
  {$group:{_id:'$server_info.server_location.hostname'}}
]
```

LIITE 9 TLS-VERSIOIDEN KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saadaan kaikki eri TLS-versioiden variaatiot sekä lukumäärä, kuinka moni verkkosivu käytti kyseistä variaatiota:

```
[
  {$project:{
    'scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites',1]},
    'scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites',1]},
    'scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites',1]},
    'scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites',1]},
    'scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites',1]},
    'scan_commands_results.tls_1_3_cipher_suites.accepted_cipher_suites':

{$slice:['$scan_commands_results.tls_1_3_cipher_suites.accepted_cipher_suites',1]}}},

{$unwind:{path:'$scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},

{$unwind:{path:'$scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},

{$unwind:{path:'$scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},

{$unwind:{path:'$scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},

{$unwind:{path:'$scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},

{$unwind:{path:'$scan_commands_results.tls_1_3_cipher_suites.accepted_cipher_suites',
    preserveNullAndEmptyArrays:true}},
  {$group:{_id:{
    ssl_2_0:{$gt:

['$scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]},
    ssl_3_0:{$gt:

['$scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]},
    tls_1_0:{$gt:

['$scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]},
    tls_1_1:{$gt:
```

```
['$scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]},
    tls_1_2:{$gt:

['$scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]},
    tls_1_3:{$gt:

['$scan_commands_results.tls_1_3_cipher_suites.accepted_cipher_suites.cipher_suite',n
ull]}},
    count:{$sum:1}}}
]
```

LIITE 10 SALAUSSARJOJEN KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään hyväksikäyttäen *cipher_suites*- taulua (LIITE 8 kertoo tarkemmin tästä taulusta) saadaan selville kuinka monta suositusten alittavaa salaussarjaa verkkosivu tuki ja lukumäärä, monellako verkkosivulla oli saman verran alittavia salaussarjoja:

```
[
  {$match:{$and:[
    {'scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites':{$ne:[]}},
    {'scan_commands_results.tls_1_1_cipher_suites.accepted_cipher_suites':{$eq:[]}},
    {'scan_commands_results.tls_1_0_cipher_suites.accepted_cipher_suites':{$eq:[]}},
    {'scan_commands_results.ssl_3_0_cipher_suites.accepted_cipher_suites':{$eq:[]}},

{'scan_commands_results.ssl_2_0_cipher_suites.accepted_cipher_suites':{$eq:[]}}]}},
  {$lookup:{
    from:'cipher_suites',
    localField:

'scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites.cipher_suite.name
',
    foreignField:'_id',as:'valid_cipher_suites'}},
  {$project:{
    scan_commands_results:1,valid_cipher_suites:1,
    'server_info.server_location.hostname':1,
    accepted_cipher_suites_size:
      {$size:'$scan_commands_results.tls_1_2_cipher_suites.accepted_cipher_suites'},
    valid_cipher_suites_size:{$size:'$valid_cipher_suites'}}},
  {$group:
{_id:{$subtract:['$accepted_cipher_suites_size','$valid_cipher_suites_size']},count:{
$sum:1}}}
]
```

LIITE 11 TLS-PAKKAUKSEN KYSELY

Seuraavalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saa lukumäärän, kuinka moni verkkosivu tuki TLS-pakkausta ja kuinka moni ei tukenut:

```
[
  {$group:

{_id:'$scan_commands_results.tls_compression.supports_compression',count:{$sum:1}}}
]
```

LIITE 12 0-RTT KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saadaan selvitettyä, kuinka moni TLS 1.3-versiota käyttävä verkkosivu, käytti 0-RTT:tä ja kuinka moni ei käyttänyt:

```
[
{$match:{'scan_commands_results.tls_1_3_cipher_suites.accepted_cipher_suites':{$ne:[]
}}},

{$group:{_id:'$scan_commands_results.tls_1_3_early_data.supports_early_data',count:{$
sum:1}}}
]
```

LIITE 13 OCSP STAPLING KYSELY

Seuraavalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saadaan selvitettyä, kuinka moni verkkosivu käytti OCSP staplingia ja kuinka moni ei käyttänyt:

```
[
  {$unwind:{path:'
    $scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$group:{_id:{
    hostname:'$server_info.server_location.hostname',
    ocsp_response_is_trusted:
'$scan_commands_results.certificate_info.certificate_deployments.ocsp_response_is_tru
sted'}}},
  {$group:{_id:'$_id.ocsp_response_is_trusted',count:{$sum:1}}}
]
```

LIITE 14 VARMENTEEN AVAIMEN KOON KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saadaan kaikki eri varmenteiden käyttämien avaimien kokojen variaatiot sekä lukumäärä, kuinka moni verkkosivu käytti kyseistä variaatiota:

```
[
  {$unwind:{path:'$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':
      {$slice:['

$scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain',1]},
     'server_info.server_location.hostname':1}},
  {$unwind:{path:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain',
    preserveNullAndEmptyArrays:false}},
  {$project:{'server_info.server_location.hostname':1,

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain.public_key.algorithm':1,

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain.public_key.key_size':1}},
  {$sort:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain.public_key.algorithm':1}},
  {$group:{_id:'$server_info.server_location.hostname',certificate:{$push:'

$scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain.public_key'}}},
  {$group:{_id:'$certificate',count:{$sum:1}}}
]
```

LIITE 15 VARMENTEEN VOIMASSAOLOAJAN KYSELY

Seuraavalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään sai verkkotunnuksien kaikkien varmenteiden voimassaoloajan alkamis- ja päättymispäivämäärät:

```
[
  {$unwind:{
    path:'$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':{
    $slice:[

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain',1]},
    'server_info.server_location.hostname':1}},
  {$unwind:{

path:'$scan_commands_results.certificate_info.certificate_deployments.received_certif
icate_chain',
    preserveNullAndEmptyArrays:false}},
  {$addFields:{not_valid_before:{$toDate:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_before'}}},
  {$addFields:{not_valid_after:{$toDate:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_after'}}}
]
```

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään saatiin tulos, kuinka monta päivää verkkotunnuksien varmenne oli vielä voimassa tai monta päivää sitten se on vanhentunut, kun tarkasteluajankohta oli 27.9.2021:

```
[
  {$unwind:{path:
    '$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':
      {$slice:['

$scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain',1]},
    'server_info.server_location.hostname':1,
    today:ISODate('2021-09-27T00:00:00.000Z')}},
  {$unwind:{path:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain',
    preserveNullAndEmptyArrays:false}},
  {$addFields:{not_valid_after:{$toDate:
```

```
'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_after'}}},
  {$project:{
    day_diff:{$dateDiff:{startDate:'$today',endDate:'$not_valid_after',unit:'day'}},
    'server_info.server_location.hostname':1}},
  {$sort:{day_diff:1}},
  {$group:{_id:'$server_info.server_location.hostname',days:{$first:'$day_diff'}}},
  {$group:{_id:'$days',count:{$sum:1}}}
]
```

Aikaisemmassa kyselyssä otettiin tarkasteluun verkkosivun varmenne, joka meni aikaisimmin vanhaksi.

LIITE 16 VARMENTEEN KÄYTTÖIÄN PITUUDEN KYSELY

Seuraavalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään sai verkkotunnuksien varmenteiden käyttöiän sekunneissa, jos verkkotunnuksella oli käytössä useampia varmenteita, kyselyssä otettiin se varmenne, jolla oli pisin käyttöikä. Kyselyssä käytettiin sekunteja, koska sillä saatiin tarkemmin määriteltyä päivien lukumäärä.

```
[
  {$unwind:{path:
    '$scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$project:{

'scan_commands_results.certificate_info.certificate_deployments.received_certificate_
chain':
      {$slice:

['$scan_commands_results.certificate_info.certificate_deployments.received_certificat
e_chain',1]},
    'server_info.server_location.hostname':1}},
  {$unwind:{path:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain',
    preserveNullAndEmptyArrays:false}},
  {$addFields:{not_valid_before:{$toDate:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_before'}}},
  {$addFields:{not_valid_after:{$toDate:

'$scan_commands_results.certificate_info.certificate_deployments.received_certificate
_chain.not_valid_after'}}},
  {$project:{certificate_lifespan:
    {$dateDiff:{
      startDate:'$not_valid_before',endDate:'$not_valid_after',unit:'second'}},
    not_valid_after:1,not_valid_before:1,
    'server_info.server_location.hostname':1}},
  {$sort:{certificate_lifespan:-1}},

{$group:{_id:'$server_info.server_location.hostname',seconds:{$first:'$certificate_li
fespan'}}},
  {$group:{_id:'$seconds',count:{$sum:1}}}
]
```

LIITE 17 VARMENTEEN YLEISEN NIMEN KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään, saatiin selvitettyä, kuinka moni verkkosivu käytti yleisenä nimenään omaa verkkotunnusta ja kuinka moni ei käyttänyt:

```
[
  {$unwind:{path:'
    $scan_commands_results.certificate_info.certificate_deployments',
    preserveNullAndEmptyArrays:false}},
  {$sort:{

'scan_commands_results.certificate_info.certificate_deployments.leaf_certificate_subj
ect_matches_hostname':1}},
  {$group:{_id: '$server_info.server_location.hostname',
    subject_matches_hostname:{$first:

'$scan_commands_results.certificate_info.certificate_deployments.leaf_certificate_sub
ject_matches_hostname'}}},
  {$group:{_id:'$subject_matches_hostname',count:{$sum:1}}}
]
```

LIITE 18 HSTS KYSELY

Alla olevalla kyselyllä *view_sslyze_helsinki_no_errors*-näkymään, saatiin tulokseksi, kuinka moni verkkosivu käytti HSTS-ominaisuutta ja kuinka moni ei käyttänyt:

```
[
  {$group:{
    _id:'$scan_commands_results.http_headers.strict_transport_security_header',
    count:{$sum:1}}}
]
```

LIITE 19 TYÖKALUN AJON TULOKSET YHTEENSÄ

Työkalun ajon tulosten yhteensä-rivi tuotettiin erillisellä ohjelmalla, joka otti verkkosivut MongoDB-näkymistä ja laski ne yhteen. Alla olevalla *sslyze_results.py*-ohjelmalla saadaan laskettua tulokset yhteensä.

```
#!/usr/bin/env python
from pymongo import MongoClient
from pprint import pprint

client = MongoClient('mongodb://localhost:27017/')

recommended_ciphers = [
    'TLS_AES_128_CCM_8_SHA256',
    'TLS_AES_128_CCM_SHA256',
    'TLS_AES_128_GCM_SHA256',
    'TLS_AES_256_GCM_SHA384',
    'TLS_CHACHA20_POLY1305_SHA256',
    'TLS_DHE_RSA_WITH_AES_128_CBC_SHA',
    'TLS_DHE_RSA_WITH_AES_128_CBC_SHA256',
    'TLS_DHE_RSA_WITH_AES_128_CCM',
    'TLS_DHE_RSA_WITH_AES_128_CCM_8',
    'TLS_DHE_RSA_WITH_AES_128_GCM_SHA256',
    'TLS_DHE_RSA_WITH_AES_256_CBC_SHA',
    'TLS_DHE_RSA_WITH_AES_256_CBC_SHA256',
    'TLS_DHE_RSA_WITH_AES_256_CCM',
    'TLS_DHE_RSA_WITH_AES_256_CCM_8',
    'TLS_DHE_RSA_WITH_AES_256_GCM_SHA384',
    'TLS_DHE_RSA_WITH_CHACHA20_POLY1305_SHA256',
    'TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256',
    'TLS_ECDH_ECDSA_WITH_AES_256_GCM_SHA384',
    'TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256',
    'TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384',
    'TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA',
    'TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256',
    'TLS_ECDHE_ECDSA_WITH_AES_128_CCM',
    'TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8',
    'TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256',
    'TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA',
    'TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384',
    'TLS_ECDHE_ECDSA_WITH_AES_256_CCM',
    'TLS_ECDHE_ECDSA_WITH_AES_256_CCM_8',
    'TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384',
    'TLS_ECDHE_ECDSA_WITH_CHACHA20_POLY1305_SHA256',
    'TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA',
    'TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256',
    'TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256',
    'TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA',
    'TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384',
    'TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384',
    'TLS_ECDHE_RSA_WITH_CHACHA20_POLY1305_SHA256',
]

def get_ciphersuites(tls_version):
    aggregate_match                                                                 =
"scan_commands_results.{}.accepted_cipher_suites".format(tls_version)
    results                                                                         =
client['jyu_tls_research']['view_sslyze_helsinki_no_errors'].aggregate([
        {
```

```
            '$match': {
                '$and': [
                    {
                        aggregate_match: {
                            '$exists': True,
                            '$ne': []
                        }
                    }
                ]
            }
        },
        #{
        #    '$limit': 100
        #}
    ])

    weak_hosts = []
    for result in results:
        hostname = result['server_info']['server_location']['hostname']
        cipher_suites                                                                    =
result['scan_commands_results'][tls_version]['accepted_cipher_suites']
        for cipher in cipher_suites:
            if not cipher['cipher_suite']['name'] in recommended_ciphers:
                #print(hostname, cipher['cipher_suite']['name'])
                weak_hosts.append(hostname)

    weak_hosts = list(set(weak_hosts))
    pprint(weak_hosts)
    print('Count: ', len(weak_hosts))
    client.close()

def get_all_hosts():
    results                                                                          =
client['jyu_tls_research']['view_sslyze_helsinki_no_errors'].find({},{'server_info.se
rver_location.hostname': 1})
    hosts = []
    for result in results:
        hosts.append(result['server_info']['server_location']['hostname'])
    client.close()
    print('All hosts: ', len(hosts))
    return set(hosts)

def get_hosts_from_mongo_view(view):
    results = client['jyu_tls_research'][view].find()
    hosts = []
    for result in results:
        #print(result)
        hosts.append(result['_id'])
    client.close()
    print(view, len(hosts))
    return set(hosts)

if __name__ == '__main__':
    hosts = get_all_hosts()
    hosts = hosts - get_hosts_from_mongo_view('view_rec_tls_version')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_cipher_suites')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_tls_compression')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_0RTT')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_ocsp')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_certificate_valid_subject')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_certificate_lifespan')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_certificate_validity_period')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_certificate_keysize')
    hosts = hosts - get_hosts_from_mongo_view('view_rec_hsts')
    print('Meets all recommendations: ', len(hosts))
```

LIITE 20 YHTIÖT JA PÄÄVERKKOTUNNUKSET

Tästä liitteestä löytyy tutkimuksessa käytetyt Helsingin pörssin 133 yhtiötä ja niiden pääverkkotunnukset. Lista kaikista Anubiksen löytämistä aliverkkotunnuksista, jotka oli haettu pääverkkotunnuksen perusteella löytyy GitHubista https://github.com/kiuru/pyTLScanner/blob/main/data/anubis_result.txt.

Nro	Yhtiö	Pääverkkotunnus
1	Afarak Group Oyj	afarak.com
2	Aktia Bank Abp	aktia.fi
3	Ålandsbanken Abp	alandsbanken.fi
4	Alma Media Oyj	almamedia.fi
5	Anora Group Oyj	altiagroup.com
6	Apetit Oyj	apetit.fi
7	AS Tallink Grupp FDR	tallink.com
8	Aspo Oyj	aspo.com
9	Aspocomp Group Oyj	aspocomp.com
10	Atria Oyj	atria.fi
11	Basware Oyj	basware.com
12	Biohit Oyj	biohithealthcare.com
13	Bittium Oyj	bittium.com
14	Boreo Oyj	boreo.com
15	CapMan Oyj	capman.com
16	Cargotec Oyj	cargotec.com
17	Caverion Oyj	caverion.com
18	Citycon Oyj	citycon.com
19	Componenta Oyj	componenta.com
20	Consti Oyj	consti.fi
21	Digia Oyj	digia.com
22	Digitalist Group Oyj	digitalist.global
23	Dovre Group Oyj	dovregroup.com
24	EAB Group Oyj	eabgroup.fi
25	Eezy Oyj	eezy.fi
26	Elecster Oyj	elecster.fi
27	Elisa Oyj	elisa.com
28	Endomines	endomines.com
29	Enedo Oyj	enedopower.com
30	Enento Group Oyj	enento.com
31	Enersense International Oyj	enersense.com
32	eQ Oyj	eq.fi
33	Etteplan Oyj	etteplan.com
34	Evli Pankki Oyj	evli.com

35	Exel Composites Oyj	exelcomposites.com
36	Finnair Oyj	finnair.com
37	Fiskars Oyj Abp	fiskarsgroup.com
38	Fortum Oyj	fortum.com
39	F-Secure Oyj	f-secure.com
40	Glaston Oyj Abp	glaston.net
41	Gofore Oyj	gofore.com
42	Harvia Oyj	harvia.fi
43	HKScan Oyj	hkscan.com
44	Honkarakenne Oyj	honka.com
45	Huhtamäki Oyj	huhtamaki.com
46	Ilkka-Yhtymä Oyj	ilkka-yhtyma.fi
47	Incap Oyj	incapcorp.com
48	Innofactor Plc	innofactor.com
49	Investors House Oyj	investorshouse.fi
50	Kamux Oyj	kamux.com
51	Kemira Oyj	kemira.com
52	Keskisuomalainen Oyj	keskisuomalainen.com
53	Kesko Oyj	kesko.fi
54	Kesla Oyj	kesla.fi
55	Kojamo Oyj	kojamo.fi
56	KONE Oyj	kone.com
57	Konecranes Oyj	konecranes.com
58	Kreate Group Oyj	kreate.fi
59	Lassila & Tikanoja Oyj	lt.fi
60	Lehto Group Oyj	lehto.fi
61	Marimekko Oyj	marimekko.com
62	Martela Oyj	martela.com
63	Metsä Board Oyj	metsaboard.com
64	Metso Outotec Oyj	mogroup.com
65	Musti Group Oyj	mustigroup.com
66	Neles Oyj	neles.com
67	Neste Oyj	neste.com
68	Nixu Oyj	nixu.com
69	NoHo Partners Oyj	noho.fi
70	Nokia Oyj	nokia.com
71	Nokian Renkaat Oyj	nokiantyres.com
72	Nordea Bank Abp	nordea.com
73	Nurminen Logistics Oyj	nurminenlogistics.com
74	Olvi Oyj	olvi.fi
75	Oma Säästöpankki Oyj	omasp.fi
76	Optomed Oyj	optomed.com
77	Oriola Oyj	oriola.com
78	Orion Oyj	orion.fi
79	Orthex Oyj	orthexgroup.com

80	Outokumpu Oyj	outokumpu.com
81	Ovaro Kiinteistösijoitus Oyj	ovaro.fi
82	Panostaja Oyj	panostaja.fi
83	Pihlajalinna Oyj	pihlajalinna.fi
84	Ponsse Oyj	ponsse.com
85	PunaMusta Media Oyj	punamustamedia.fi
86	Puuilo Oyj	puuilo.fi
87	QPR Software Oyj	qpr.com
88	Qt Group Oyj	qt.io
89	Raisio Oyj	raisio.com
90	Rapala VMC Oyj	rapalavmc.com
91	Raute Oyj	raute.com
92	Reka Industrial Oyj	rekaindustrial.fi
93	Revenio Group Oyj	reveniogroup.fi
94	Robit Oyj	robitgroup.com
95	Rovio Entertainment Oyj	rovio.com
96	Saga Furs Oyj	sagafurs.com
97	Sampo Oyj	sampo.com
98	Sanoma Oyj	sanoma.com
99	Scanfil Oyj	scanfil.com
100	Sievi Capital Oyj	sievicapital.fi
101	Siili Solutions Oyj	siili.com
102	Sitowise Group Oyj	sitowise.com
103	Solteq Oyj	solteq.com
104	Soprano Oyj	soprano.fi
105	Sotkamo Silver AB	silver.fi
106	SRV Yhtiöt Oyj	srv.fi
107	SSAB	ssab.com
108	SSH Communications Security	ssh.com
109	Stockmann Oyj Abp	stockmanngroup.com
110	Stora Enso Oyj	storaenso.com
111	Suominen Oyj	suominen.fi
112	Taaleri Oyj	taaleri.com
113	Talenom Oyj	talenom.fi
114	Tecnotree Oyj	tecnotree.com
115	Teleste Oyj	teleste.com
116	Telia Company	teliacompany.com
117	Terveystalo Oyj	terveystalo.com
118	TietoEVRY Oyj	tietoevry.com
119	Tikkurila Oyj	tikkurilagroup.com
120	Tokmanni Group Oyj	tokmanni.fi
121	Trainers' House Oyj	trainershouse.fi
122	Tulikivi Oyj	tulikivi.com
123	United Bankers Oyj	unitedbankers.fi
124	UPM-Kymmene Oyj	upm.com

125	Uponor Oyj	uponor.com
126	Vaisala Oyj	vaisala.com
127	Valmet Oyj	valmet.com
128	Valoe Oyj	valoe.com
129	Verkkokauppa.com Oyj	verkkokauppa.com
130	Viking Line Abp	vikingline.fi
131	Wärtsilä Oyj Abp	wartsila.com
132	Wulff-Yhtiö Oyj	wulffyhtiot.fi
133	YIT Oyj	yitgroup.com